

Investigation of Linear Quadratic Control and Guidance System Design for a small UAV

Thesis submitted for the degree of

Doctor of Philosophy

at the University of Leicester

by

Teng David Chollom

Control Research Group

Department of Engineering



UNIVERSITY OF
LEICESTER

April 18, 2019

Abstract

Autonomous air vehicles that are efficient and robust require reliable control systems that enable them fly a mission. Modern flight controllers such as the linear quadratic regulators provide optimal solutions for obtaining control gains, however this approach requires the selection of weighting matrices which often entail time consuming trial and error process. A multi-objective particle swarm optimization scheme is developed to select weighted matrices based on the output performance specification. Simulation results of the application of this method for an UAV lateral-axis model provided an output response that was within the specification limits. Furthermore the rapid design and development of the autonomous air vehicle requires a fairly accurate simulation model which in this case for the first time was obtained for the GULMA air vehicle using the Athena Vortex Lattice method that gives aerodynamic coefficients and stability derivatives as well as a linear model. Model-based controller synthesization was conducted using this linear model to test for robust stability and performance within specified percentage of uncertainty. The results were within stable margins for μ high order controllers. Finally a guidance system based on the Lyapunov Vector Field was applied to the airborne test platform for tracking of straight line and circular paths in the presence of wind disturbance. The simulated results provided satisfactory results that would be considered for real-time flight tests.

Dedication

I dedicate this thesis to Almighty God and to my wife, Jenny Anya Chollom and parents, (Late Engineer) Davou Chibi Chollom and Mrs Paulina Chollom.

Acknowledgements

First of all, I want to thank Almighty God for His guidance to prepare, complete and submit this thesis. I would also want to thank my wife Jenny and my children, Kayla and Jotham; for all their prayers, support and understanding to achieve my goals in life; you guys have been really awesome throughout this tedious period. I also like to especially thank my mum, Ngo Paulina Chollom for her sweet words of encouragement and prayers. To all my family members, I say thank you for standing by me from start to end of this journey. God bless you all.

I sincerely want to thank my supervisor, Dr. Emmanuel Prempain, who went the extra mile to guarantee and contribute immensely to the success of my PhD. I want to appreciate his encouragement, support and advice given me within the period of my study. My appreciation extends to other supervisory academicians in the Control Group for their all their assistance and support.

Furthermore, I would like to thank my PhD colleagues for their dogged support day and night in the the Control laboratory; as well as all members of staff in the Department for creating a sublime environment and administrative support.

Finally, I also would like to acknowledge the Chief-of-the-Air Staff for the opportunity granted me to study for a PhD and to serve in the Nigerian Air Force.

Contents

1	Introduction	1
1.1	Background of the study	1
1.2	Nigerian Air Force Research and Development Centre	2
1.2.1	Objectives	5
1.2.2	Aim	5
1.3	Research Contributions	5
1.4	Organisation of Thesis	6
2	Literature overview	8
2.1	Overview	8
2.1.1	AVL Modelling	10
2.1.2	Optimization Algorithms	10
2.1.3	Particle Swarm Optimization	13
2.2	Flight Control Systems	15
2.2.1	Learning-based controllers	16
2.2.2	Linear Flight Controllers	16
2.2.3	Model-based Non-linear controllers	17
3	UAV Models	20
3.1	UAV Simulation Platform	20
3.1.1	Ultra stick UAV	20
3.1.2	GULMA UAV	21

3.2	Non-linear Model	22
3.2.1	Flight Dynamic Modelling	23
3.2.2	Calculation of Aerodynamic Coefficients	26
3.2.2.1	Performance Analysis	27
3.2.2.2	Weight Analysis	28
3.2.2.3	Trim Analysis	29
3.2.3	Inertia Model	31
3.2.4	Propulsion Model	31
3.2.4.1	Piston Engine	32
3.2.4.2	Calculations for Engine Thrust	34
3.2.5	Summary	34
4	Control and Optimisation	37
4.1	Linear Quadratic Control Design	37
4.1.1	Longitudinal Control Design	38
4.1.2	Simulation and Analysis	47
4.1.3	Lateral-directional Control Design	49
4.2	Multi-objective Optimization	56
4.2.1	Multi-objective PSO	56
4.2.2	Fitness Consideration	58
4.2.3	Simulation and Analysis	59
4.3	Robust Flight Controller Synthesis	62
4.3.1	Robust Design Objectives	64
4.3.2	Selection of Weighting Functions	65
4.3.3	Closed Loop System with 2-DOF H_∞ Synthesis	68
4.3.4	Closed Loop System with 2-DOF μ Synthesis	71
4.3.5	Simulation and Analysis of Controller Synthesis	72
4.3.5.1	Comparison of Controllers in the Frequency Do- main	73

4.3.5.2	Comparison of Controllers in the Time Domain	74
4.3.6	Robust Stability and Performance Analysis	76
4.4	Summary	77
5	Guidance System Design	80
5.1	UAV Guidance System	80
5.2	Guidance System	81
5.2.1	Kinematic Guidance Model	81
5.2.2	Dynamic Guidance Model	82
5.3	Objective	82
5.4	Lateral Guidance	83
5.4.1	Straight Path Following	83
5.4.2	Circular Path Following	86
5.5	Longitudinal guidance	88
5.6	Kalman Estimator Dynamics	90
5.7	Simulation Results	94
5.7.1	Summary	95
6	Conclusion and Further Work	105
6.1	Conclusions	105
6.2	Further Work	107
A	AVL Modelling	109
A	Performance	109
B	Aerodynamic	111
C	Weight analysis	112
D	Trim results from AVL	114
E	Eigenmode Analysis	114

B UAV Models	119
A Linearized models	119

List of Figures

1.1	Project Development V-model.	3
1.2	GULMA UAV on board system block diagram.	4
2.1	Framework diagram for implementation of AVL.	11
2.2	Satndard PSO Flow Chart algorithm.	14
3.1	Picture of the Ultra stick UAV adapted from [24].	21
3.2	Picture of the NAF GULMA UAV.	21
3.3	Diagram of the conventional fixed-wing non-linear model.	23
3.4	Diagram showing forces and moments acting on UAV.	24
3.5	Breakdown of weight of GULMA UAV sub-systems.	28
3.6	Breakdown of weight of GULMA UAV avionics.	29
3.7	Positions of critical components on the GULMA UAV.	29
3.8	GULMA UAV lift versus moment coefficients for varying AOA.	30
3.9	Power generation methods for 219 MALE/HALE UAVs.	32
3.10	Data for 23 UAVs similar to GULMA UAV.	33
3.11	Graphical representation of the engine power as a function MTOW and wingspan	34
3.12	Engine models considered for use on the GULMA UAV.	35
3.13	Trimmed level flight by velocity of the GULMA UAV.	36
4.1	Visualization of GULMA UAV singular value decomposition.	42

4.2	Long period longitudinal response of GULMA UAV to an initial pitch disturbance of 1° .	44
4.3	LQG controller block diagram for GULMA UAV.	44
4.4	LQ integral control and Kalman filter subsystem.	45
4.5	Closed-loop pitch response to 0.35 rad (20°) reference input.	48
4.6	Closed-loop pitch response with $\pm 10\%$ variability.	49
4.7	LQI block diagram.	56
4.8	Multi-objective PSO algorithm flow chart.	59
4.9	Doublet response based on multi-objective PSO results for Q_1R_1 .	60
4.10	Doublet response based on standard PSO results for Q_2R_2 .	61
4.11	Aileron response with state-feedback based on multi-objective PSO results for Q_1 and R_1 .	62
4.12	Aileron response with state-feedback based on standard PSO results Q_2 and R_2 .	62
4.13	Yaw angle response due to doublet command.	63
4.14	Rudder response due to doublet command.	63
4.15	Standard Lower LFT block diagram.	64
4.16	Roll command performance objective.	65
4.17	Yaw command performance objective.	66
4.18	Input multiplicative uncertainty model.	67
4.19	Error dynamics of nominal plant in terms of frequency.	68
4.20	2-DOF Closed loop for the lateral model.	69
4.21	μ -Synthesis block diagram.	71
4.22	Closed-loop worst-gain response of controllers.	74
4.23	Time domain responses H_∞ controller.	75
4.24	Time domain responses μ controller.	75
4.25	μ closed-loop time responses due to 50% increase in command signal.	77

4.26	μ closed-loop time responses due to 50% decrease in command signal.	78
5.1	Full UAV system architecture.	81
5.2	Geometry for lateral guidance of UAV to follow a straight-line path.	83
5.3	Crab angle due to wind disturbance.	85
5.4	Geometry for lateral guidance of UAV to follow a circular path.	86
5.5	Geometry for longitudinal guidance of UAV.	88
5.6	State estimation via EKF.	94
5.7	Estimation error for states via EKF.	95
5.8	Matlab/Simulink LGVF simulation diagram.	96
5.9	UAV straight line path error due to $wind_{n,e} = 1m/s$	97
5.10	Response to straight line path tracking due to $wind_{n,e} = 1m/s$	98
5.11	UAV straight line path error due to $wind_{n,e} = 3m/s$	99
5.12	Response to straight line path tracking with $wind_{n,e} = 3m/s$	100
5.13	UAV circular path error due to $wind_{n,e} = 1m/s$	101
5.14	Response to circular path tracking with $wind_{n,e} = 1m/s$	102
5.15	UAV circular path error due to $wind_{n,e} = 3m/s$	103
5.16	Response to circular path tracking with $wind_{n,e} = 3m/s$	104
A.1	List of performance values for analysis at sea level at 1000 metres.	110
A.2	List of performance values for analysis at 1000 metres.	111
A.3	Variation of coefficients for NACA 2415 air-foil.	112
A.4	Variation of coefficients for NACA 0012 air-foil.	112
A.5	Lift-Drag ratio versus airspeed for GULMA UAV.	113
A.6	Mass distribution sheet for GULMA UAV.	113
A.7	Trim for loading case 2.	115
A.8	Trim for loading case 4.	116
A.9	Eigenmode view of model in flight level condition.	117

A.10 Eigenmode side slip motion diagram. 117
A.11 AVL stability derivatives results for case 2. 118
A.12 AVL stability derivatives results for case 4. 118

B.1 UAV Non-linear Simulink Test Platform [56]. 121
B.2 UAV Non-linear Simulink Test Platform [56]. 122

List of Tables

3.1	Aerodynamic Coefficients.	27
3.2	GULMA and Ultrastick UAV Configuration.	27
4.1	Trim Conditions.	39
4.2	Characteristic values of longitudinal modes.	41
4.3	Characteristic values of lateral-directional modes.	51
4.4	GULMA UAV lateral plant eigenvectors.	52
4.5	Robust stability and performance results.	76
5.1	Parameters for straight line path tracking.	96
5.2	Parameters for circular path tracking.	97
A.1	Critical dimensional parameters.	109
A.2	Aerodynamic critical parameters.	110

List of Abbreviations

Acronyms

<i>AI</i>	Artificial Intelligence
<i>AOA</i>	Angle-Of-Attack
<i>AVL</i>	Athena Vortex Lattice
<i>CATIA</i>	Computer Aided Three-dimensional Interactive Application
<i>CS – VLA</i>	Certification Standards for Very Light Aircraft
<i>COTS</i>	Commercial-Of-The-Shelf
<i>DOF</i>	Degrees of Freedom
<i>FCS</i>	Flight Control System
<i>ISR</i>	Intelligence Surveillance Reconnaissance
<i>LGVF</i>	Lyapunov Guidance Vector Field
<i>LQR</i>	Linear Quadratic Regulator
<i>MOPSO</i>	Multi-Objective Particle Swarm Optimization
<i>NAF</i>	Nigerian Air Force
<i>NASA</i>	National Air Space Agency
<i>PID</i>	Proportional Integral Derivative
<i>PSO</i>	Particle Swarm Optimization
<i>UAV</i>	Unmanned Air Vehicle(s)

Symbols

c_l	aerodynamic coefficient of lift
c_d	aerodynamic coefficient of drag
C_x	aerodynamic coefficient of force along body $x - axis$
C_y	aerodynamic coefficient of force along body $y - axis$
C_z	aerodynamic coefficient of force along body $z - axis$
C_l	aerodynamic coefficient of roll moment
C_m	aerodynamic coefficient of pitching moment
C_n	aerodynamic coefficient of yaw moment
c_e	inertial east position of circular path
c_d	inertial down position of circular path
c_n	inertial north position of circular path
χ	UAV course angle
χ_{cmd}	commanded course angle for UAV
χ^0	tangent angle to circular path with respect to north east frame
χ_{path}	angle of straight line path for north, east down directions
χ^∞	initial course angle of UAV
d	distance between UAV and centre of a circular path
\dot{x}	derivative of x with respect to time, $\frac{dx}{dt}$
e_{pathy}	path error
k_{path}	positive constants transition of vector fields for a straight line path
k_{circ}	positive constants for transition of vector fields for a circular path
λ	direction of path, clockwise or anti-clockwise
p_e	position of inertial east
p_d	position of inertial down
p_n	position of inertial north
ρ	radius of circular path
\hat{x}	UAV state estimates

Chapter 1

Introduction

1.1 Background of the study

Unmanned Aerial Vehicles continue to increasingly attract interest all over the World. This is due to its relevance to military and civilian applications; such as reconnaissance, aerial intelligence gathering, agriculture as well as being referred to dull, dirty and dangerous concept of UAV operations. Hence the demand for these platforms to be operational in various environment off-set the need for improved robustness and performance qualities. Described in [32], an UAV flight control performance and robust qualities could be improved with minimal degradation to system stability. So, the general view of the inner-loop controller for an UAV provides the necessary stability analysis of the attitude rates and ensures improved outer-loop controller performance. To further enhance this performance, optimization algorithms have also been developed for tuning of the system control gains as detailed in [50] [10]. This then indicates the need for inclusion of artificial intelligence to control system design. Examples of some optimization algorithms used are particle-swarm, genetic, differential evolution, and multi-objective evolutionary [64] [33] [46] [81]. In this thesis therefore we shall develop a robust multi-objective PSO algorithm that is demonstrated

for optimized tuning and selection of weighting matrices used to obtain control gains of a lateral-directional inner-loop flight controller. The UAV (nick named 'GULMA'), is a H-tail fixed-wing type that was designed and developed by the Nigerian Air Force for ISR applications; currently suited for the north-eastern region of Nigeria. A simulation model (test platform) for the UAV is developed using the AVL software by [1], which enabled the prediction of the aerodynamic stability derivatives and the conduct of an eigenmode analysis to obtain a full linear model of the aircraft. A non-linear model, for the first time, for GULMA UAV was then further developed using the results from AVL by modifying an existing UAV flight test platform from the University of Minnesota [4]. This study is therefore limited to the application of a multi-objective PSO algorithm to a Matlab/Simulink linearized model of the UAV and demonstration of model's robust and performance stability using model-based controller synthesis. We shall further examine the development of a flight guidance system within the context of analysis of the existing LGVF method for guiding a non-linear airborne test platform to follow pre-set paths as described in [5]. Hence this would give a base-line approach for how we would conduct a flight guidance experiment test using an autopilot hardware such as the arduino autopilot for the GULMA UAV.

1.2 Nigerian Air Force Research and Development Centre

In line with best practices associated with aerospace industries, the GULMA UAV project was divided into structures, systems and avionics teams. Other research areas explored were in computational fluid dynamics and flight dynamics which were incorporated to provide appropriate design validation roles. Noteworthy, the majority of the GULMA UAV project team had previously been involved in the development of the AMEBO UAV, a joint project between the

NAF and Cranfield University.

Avionics are electronic systems used on aircraft, artificial satellites and spacecraft that essentially provide some form of awareness to effectively pilot the air vehicle [77]. Avionics systems also include communications, navigation, display and flight management of systems. The scope of the GULMA UAV avionics design included capabilities such as flight control, communication, autopilot, video and data telemetry and an integrated ground control system. The avionics system design was guided by requirements obtained from Certification Standards for Very Light Aircraft [17] and NAF requirements for autonomous flight and extended range for video and telemetry data acquisition. To achieve this a V-model development [54] diagram described in Figure 1.1 was implemented.

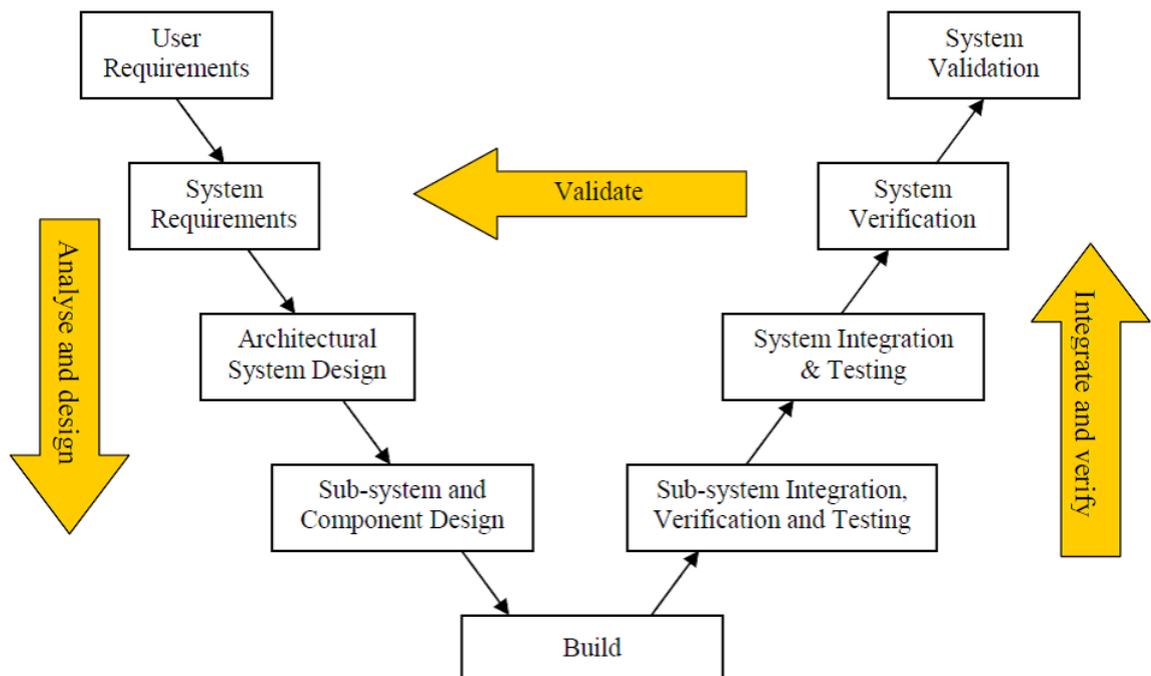


Figure 1.1: Project Development V-model.

The V-model consists of two main branches; the descending branch and the ascending branch. The descending branch is concerned with the collection of the requirements (from top to the single sub-system) while the ascending branch contains phases of integration in which the individual components are interface

together, and phases of validation to verify the overall system meets all the requirements defined in the descending branch. Implemented activities done included requirements capture, overall system and sub-systems block diagram design through to system integration and testing. The overall system block diagram is shown in Figure 1.2.

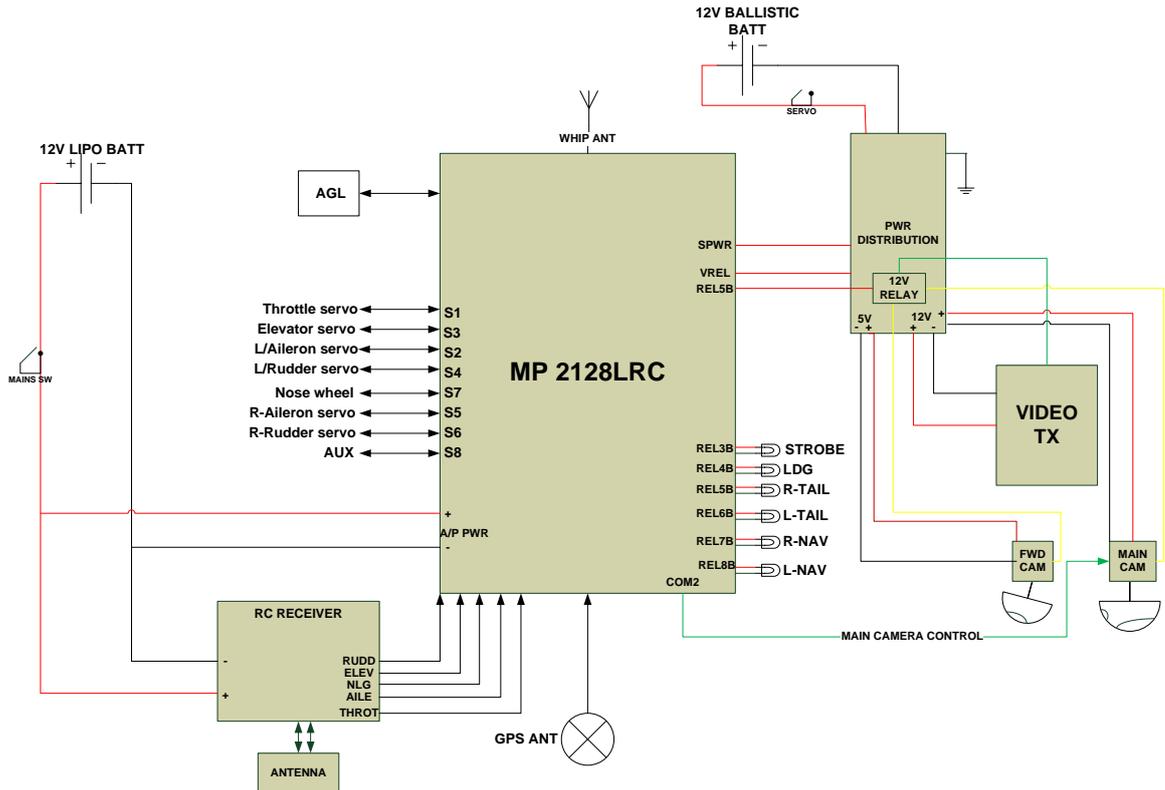


Figure 1.2: GULMA UAV on board system block diagram.

The conceptual diagram for the MP2128LRC micropilot autopilot provided the basis for the final integration design of the on-board avionics system as in [9]. Furthermore, additional COTS sub-components for this autopilot included: the lighting system, distribution box, video transmitter, forward looking camera and main camera(M1-D). Consequently, due to high costs of obtaining the autopilot system (usually, > 25,000.00 US Dollars), there was a need to rapidly design and develop a full or semi autonomous micro-controller which would effectively reduce overhead costs required to obtain an international licence as well as other

logistics items. This would entail first, developing a simulation model of the air vehicle to effect the design and analysis of a test-bench FCS for flight test and secondly, conduct the optimization of performance objectives.

1.2.1 Objectives

- Modelling and analysis for a fixed wing UAV simulation model using AVL and Matlab/Simulink software.
- Design and assessment of GULMA UAV lateral directional-axis LQR controller using a multi-objective particle swarm optimization algorithm as well as robust and performance analysis using H_∞ and μ controller synthesis.
- Design and application of a PID-based non-linear LGVF system to a fixed-wing UAV simulation model.

1.2.2 Aim

The aim of this thesis project is to assess the Nigerian Air Force UAV with a view to producing an airborne simulation test bench for designing and developing future UAV flight control and guidance systems.

1.3 Research Contributions

- Decentralised anti-windup design approaches with application to Quadrotor UAVs, Poster presentation for control and optimisation UK, St. John's College, Oxford, Ofodile, Nkemdilim A and Turner, Matthew C and Cholom, Teng D, September 2015.
- Application techniques of multi-objective particle swarm optimization: Aircraft flight control, UKACC 11th International Conference 1-6 pages IEEE,

Chollom, Teng D and Ofodile, Nkemdilim A and Ubadike, Osichinaka C, August 2016.

- LQR Controller Design with Antiwindup Compensation for Quadrotor Attitude Control, IET Control Theory and Applications ICA-219-2019, Matthew C. Turner, Paul Jemitola, Teng Chollom, Nkemdilim Ofodile, submitted 2019 awaiting acceptance.

1.4 Organisation of Thesis

The thesis is organised as follows:

Chapter 2 is comprised of a literature review on the AVL modelling framework that was developed and implemented to enhance modelling of the GULMA UAV. A step by step description of the framework ensured that a relatively accurate model of the GULMA UAV was achieved as well as obtaining the linear model which was compared to the model from Matlab environment. A review of existing literature on particle swarm optimization algorithm was also conducted to relate this technique with a developed MOPSO algorithm.

In Chapter 3, the non linear models used for simulation and subsequent control design and synthesis are described with respect to AVL and Matlab 6-DOF prediction and modelling techniques. A comparison of the two models - GULMA and Ultra stick UAV was also done to emphasize their difference in configuration geometry and weight. A description of the developed non linear model in terms of aerodynamic coefficients, inertia model and propulsion model is annexed in Appendix B.

Chapter 4 we analyse the longitudinal and lateral-axis models. Subsequently, a longitudinal controller using the LQG/LQR method was design and demonstrated through simulation and analysis. Furthermore a description of the application techniques of a multi-objective PSO algorithm for a fixed-wing UAV

lateral-axis model is examined via simulation and analysis as well.

Chapter 5 describes the design of a guidance system for a non-linear UAV model. We describe the lateral and longitudinal guidance strategy using the LGVF technique. This technique is applied and demonstrated using the Ultra stick airborne simulation UAV model in the presence wind disturbance. We also describe the Kalman state estimator used for estimating the UAV states.

Finally in Chapter 6, the thesis concludes with discussions on simulation results obtained and future work in the area of application of UAV guidance systems on the GULMA UAV.

Chapter 2

Literature overview

2.1 Overview

Research and Development of various UAV all over the World continues to gain momentum as applications of these vehicles vary rapidly over the years. These applications range from individual (social needs) to civilian (government) and military uses as described in [84] [2] [63]. Military applications vary from tactical to strategic missions which includes: combat, close air support, rescue, reconnaissance for intelligence gathering just to mention a few. All these missions require the vehicle to have robust performance throughout the its flight phases (take-off to landing). Hence the design and development of an efficient FCS cannot be over emphasized. Military literature and documents pertaining to state-of-the-art FCS are few due to security classification placed on such information. However, we can find well documented literature for some UAV presently being applied for military applications in [2]. Also in [84] [14], categorization of various UAV were described as; the Long-range, Medium-range and Close-range UAV. These various categories were generally based on their configurations such as maximum take-off weight, wing-span, payload and so on, as well as classification based on vehicle performance objectives which include:

optimization of fuel cost for long-range endurance, beyond-line-of-sight telemetry for intelligence gathering and optimized flight control during mission phases just to mention of few [84] [19]. In the design and development of these air vehicles, it is necessary to ensure that the FCS are reliable and possess efficient performance of their flight missions. The design and application of UAV flight controllers are heavily documented in [13] [58] [16] [5] [45]. This exhaustive literature for FCS design and analysis are mainly based on the use PIDs due to its great advantage in ease of implementation in most industries all over the World. Some of the other modern flight controllers are the adaptive controllers in [41], model predictive controllers [79], linear quadratic regulators, H_∞ and μ controllers, traditional gain-scheduling controllers [49]. The UAV designed and developed by the Nigerian Air Force requires a design, assessment and analysis of the on-board FCS using a simulation model (test platform) of the vehicle. This would enable reassessment and review for the handling quality of the aircraft with the aim of improving on the existing performance of the vehicle. Hence, this literature survey aims to cover the research made so far towards achieving the objectives of the project. The overview that covers the first objective is based on the work undertaken by the UAV research team in [1], main requirements for the developing the model will be revised. Then simulation of the model will be conducted and include discussion with specific remarks on limitation and model uncertainties.

Finally an in-depth look at the existing literature for the design and application of PID and LQR controllers for fixed wing UAVs with a view to proposing an application technique that improves flight performance during UAV flight mission.

2.1.1 Modelling using Athena Vortex Lattice

The Athena Vortex Lattice (AVL) method is used essentially for fast evaluation of lifting surfaces such as wings, control surfaces and so on, that are integrated together to form the aircraft configuration characteristics [26]. A comparison of the lift curves slopes using lifting line theory and AVL was done in line with [55], where it was concluded that the AVL was a significant tool for evaluation of an aircraft configuration scheme as the the program provided the required approximations of aircraft stability derivatives which could be further validated through wind tunnel testing or using computational fluid dynamics methods. Furthermore AVL was implemented by [47], to provide the necessary aerodynamic derivatives used to develop flight requirements for the blended wing body Hyperion 2.1 green aircraft designed and developed by NASA. In previous work undertaken by [59] [38] [75], we note that the AVL software can be interfaced with MATLAB to develop a 6-DOF aircraft model from the aerodynamic derivatives obtained from the program. To this end, the AVL software and MATLAB was used to develop the GULMA UAV model. The modelling and simulation of the Meridian UAV was successful with aid of advanced aircraft analysis and AVL software described by [69]. The model enabled the project team to analyse the sensitivity of the effect of varying some stability derivatives on aircraft dynamics and payload operated on the aircraft. The development of the GULMA UAV model was done using the the developed framework diagram described in Figure 2.1.

2.1.2 Optimization Algorithms

Optimization algorithms are highly used in control systems to improve performance of systems by computing the minimum values of specified mathematical functions [60]. Accordingly, optimization algorithms mostly evaluate trade-offs, analyse control systems, determine patterns in data, just to mention a few [35].

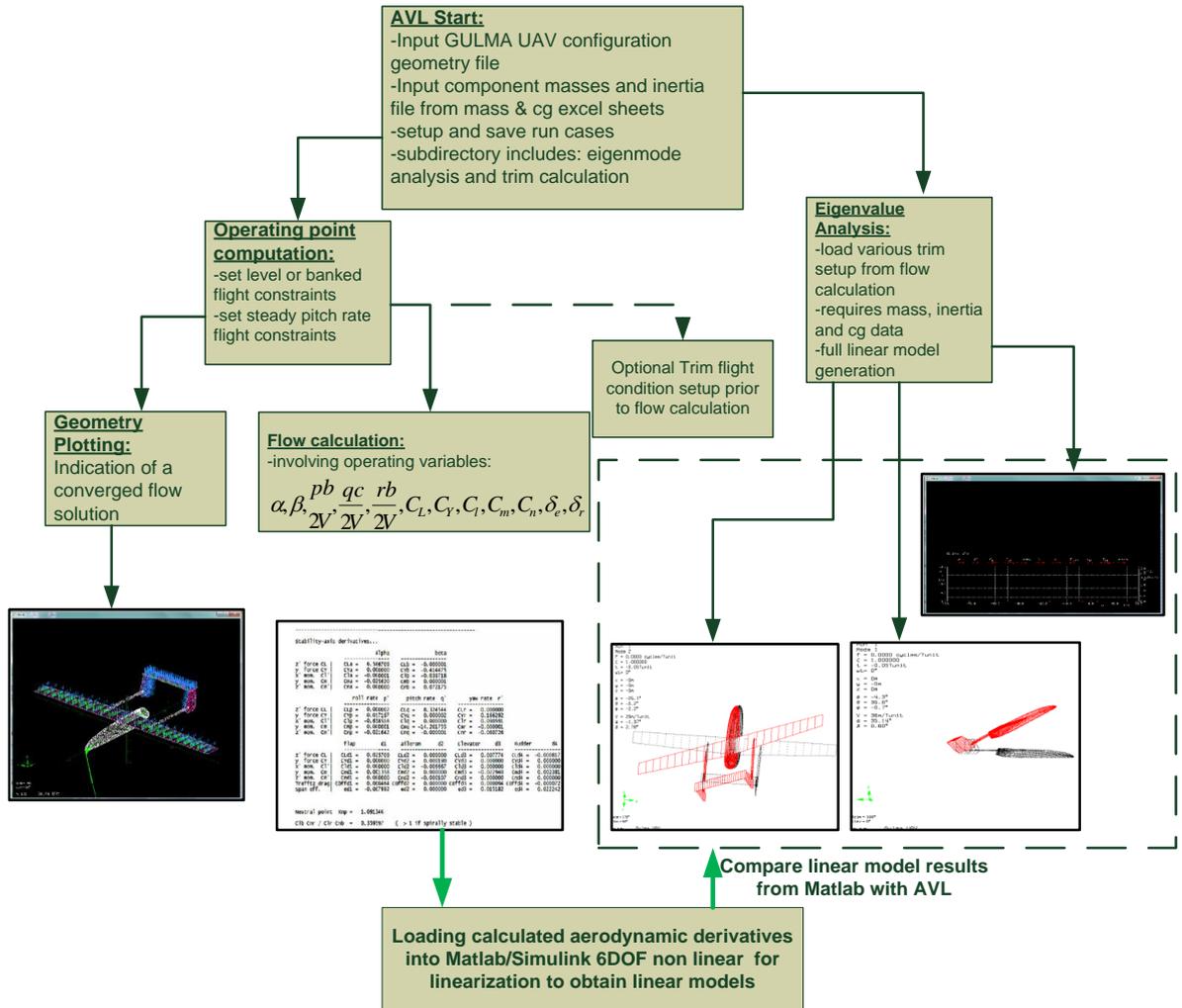


Figure 2.1: Framework diagram for implementation of AVL.

In literature there exist a vast of optimization algorithms applied for various systems as well as control systems [57]. In this thesis we shall consider the application of a SI optimization scheme, PSO, to control a small semi-autonomous unmanned air vehicles. The PSO would be a modified algorithm to solve a multi-task problem while maintaining a flight performance objective. Some other optimization algorithms available in literature include the evolutionary optimization algorithm, differential evolution, particle swarm optimization, dynamic programming, gradient methods just to mention a few [68] [70] [66]. According to [81], the similarities between these algorithms are not far fetched as the differences

are mainly the process of selection of dominated agents in the search space.

Directly related to this research as previously stated is the SI optimization algorithm that is seldom recognised as an AI technique which is based on the collective behaviour of self-organized systems in a somewhat distributed network [21] [44] [28] [31] [31]. Generally the AI technique is made up of agents or elements that interact with each other in a specified environment with a decentralized control pattern as well as having a group behaviour mostly found in nature. We mention here some recent SI techniques which include: Particle swarm, Ant colony, Bee colony, Wasp colony, e.t.c [20] [22].

Accordingly some on going research based on SI application includes:

- Design of cooperative control systems for small to micro UAV using swarm techniques in [39].
- Investigating the use of swarm technology for planning space trajectories [67].
- Swarm intelligence to control micro/nano robots for delivery, surgery, sensing and detoxification of in the human body system [52].
- Load balancing of call nodes in telecommunication networks [73].

Some identified merits of SI as described in [44] are as follows:

- **Scalability.** Scalable control architecture could be applied to a couple of agents or many more agents.
- **Flexibility.** The agents or elements can be easily added or removed without directly affecting the control structure.
- **Robustness.** The agents are fairly easy to design, the reliance on individual agents is small and failure of a single agent has little impact on the overall system's performance.
- **Adaptability.** The systems are able to adapt to new situations easily.

2.1.3 Particle Swarm Optimization

Renowned scientists, Dr. Kennedy and Dr. Eberhart in [83] [15], had the idea of simulating the social behaviour of flocking birds in search of food. Hence the PSO was developed by them to emulate this social behaviour of which each member (bird) in a swarm would adapt its search path by learning from its own characteristic and other members' characteristics. Their characteristic consisted of member position and velocity (direction). A velocity update of PSO is given in Equation 2.1 while for the particle position update is given in Equation 2.2. In an inertia weighted PSO, each particles learns from the its local best (p_{best}) and the global best (g_{best}) for all dimensions. The parameters C_1 and C_2 are the acceleration constants that reflect the weighting of the stochastic acceleration term that pull each particle toward the p_{best} and g_{best} respectively. The inertia weight (w_i) is used to facilitate both global and local search space. Large values of w_i facilitates global search while smaller values favour local search [36]. In this study, w_i was made to decrease exponentially as the generation progresses. This approach facilitated global search within the early stage and later was favourable to the local search as the generation comes to an end. In standard PSO, all particles learn from its own p_{best} and g_{best} at all times and for all dimensions [30]. Thus we proposed the use of a multi-objective PSO that would constrain the social learning process to multiple objectives [78]. The decision guiding the choice of particle's dimension to learn from the local best or other particles' local best depends on the function, P_c , called the learning probability [78]. For every dimension (N) of a particle (i) a random number of range $([0, 1])$ is generated so that if this random number is greater than P_c , the particular dimension will learn from its own local best. P_c is given in Equation 2.3 as described in [40] where (S) is the number of particles (swarm size).

$$V_i^d = w_i \cdot V_i^d + C_1 \cdot rand1_i^d \cdot (p_{best_i}^d - X_i^d) + C_2 \cdot rand2_i^d \cdot (g_{best_i}^d - X_i^d) \quad (2.1)$$

A flow chart as shown in Figure 2.2 for the standard PSO algorithm applied to the UAV linear model.

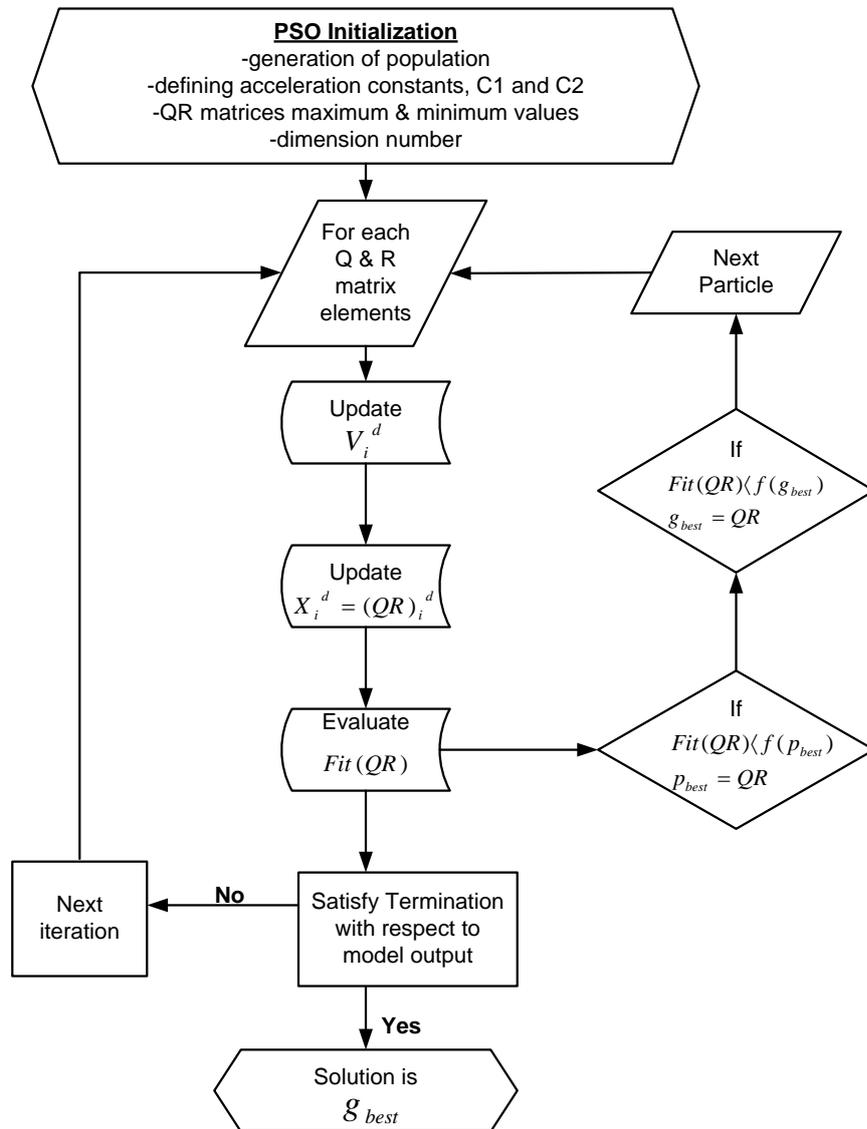


Figure 2.2: Standard PSO Flow Chart algorithm.

In Equation 2.1, $p_{best_i}^d$ is any particle's personal best position selected for initialization given as $f_i = [f_i(1), f_i(2) \dots f_i(N)]$ defined the path the particle's best should follow. Also, $rand_i^d$ is a normal distribution random number in the range $[0, 1]$, and each particle dimension (d) has its own $rand_i^d$. X is referred to particles position (potential solutions) while C is the acceleration pull from the particle centre of gravity.

$$X_i^d = X_i^d + V_i^d \quad (2.2)$$

$$P_c i = 0.005 + 0.45 \frac{e^{\frac{10(i-1)}{S-1}} - 1}{e^{10} - 1} \quad (2.3)$$

Two main points shall be considered when developing the multi-objective optimization which are described in [44]; how to select particles (to be used as leaders) in order to give preference to diverging solutions over those that are converging; how to retain the these solutions found during the search process in order to report solutions that are converging with respect to all initial populations and not only with respect to the current one, and how to maintain diversity in the swarm in order to avoid convergence to a single solution [39].

2.2 Flight Control Systems

The state of the art flight control system provides a basis for improving the autonomy level of UAV [18]. There are different control frameworks and algorithms which have been developed for pilot-commanded and pilot-less aircraft. Some of the traditional and more recent methods include: learning-based flight control, linear flight control and model-based non-linear control techniques. An overview of these various techniques are discussed.

2.2.1 Learning-based controllers

Some of the learning-based controllers consist of the following: fuzzy logic, human-based learning and neural networks. Fuzzy logic was developed by [86]. This technique involved the interpreting data and knowledge used by pilots into the rules used by a fuzzy control system. These control rules were generated automatically from training data using a model free methodology. Human-based learning involved extracting or modelling input sequences and feedback of the pilot during flight mission execution. This led to the development and implementation of an intuitive control approach for designing autonomous systems as described in [72]. Artificial Neural Networks is also a learning based control technique that is used to identify some model uncertainties and is combined with standard control techniques. [11] and Roberts described an ANN controller was developed for helicopter hovering mission. It uses direct mapping of inertial data to actuator control via a feed-forward network using the back-propagation training regime. Partial hovering for several seconds has been achieved with a small UAV helicopter.

2.2.2 Linear Flight Controllers

Linear flight controllers usually includes: PIDs, LQR, H_∞ , that have been used to achieve autonomous flight missions [71]. Controllers such as PID are constantly being used in the industry as its simple and cost effective to implement. Traditionally, PID are implemented as successive loop framework where inner loops provide the stabilization of aircraft attitude while outer loops guarantee aircraft motion by decoupling the PID controller into longitudinal and lateral axis frames [17]. The LQR scheme already discussed in this thesis is a popular optimal control method that has been successfully applied to control several fixed wing UAV configurations. The implementation of the LQR control method using the particle swarm optimization algorithm to implicitly select the stabilizing

matrices for computation of the control gains [13]. The H_∞ control approach is mainly implemented in cases that involve robust control design to guarantee aircraft stability for various parametric uncertainties and modelling errors. Other linear flight controllers also include the traditional gain scheduling and control selection techniques which is a collection of simplified linear models based on varying flight parameters.

2.2.3 Model-based Non-linear controllers

Linear controllers are limited to control aircraft with non-linear dynamics associated varying parameters, weather conditions, model uncertainties and so on. Hence the development of non-linear flight controllers to overcome such non-linearities. It involves non-linear state variable transformations to linear co-ordinate systems to enable the use of linear methods to control the system. Model-based non-linear methods include: Lyapunov based control, dynamic inversion (feedback linearisation), adaptive control, back-stepping, model predictive control, nested saturation, and so on. An overview of the Lyapunov based control method is described as follows:

The Lyapunov based control method is selected for this research as it gives the solution to guarantee the stability or convergence of the dynamic motion to its origin. This method is described according to the Lyapunov stability theorem. Firstly we consider the following definitions.

In a state-feedback framework it is already established that the control input (u) is often a function of the states (x). Hence, for stability analysis, it is sufficient to consider the following definitions and theorems for an autonomous system:

$$\dot{x}(t) = f(x(t)) \quad (2.4)$$

Definition according to Lyapunov, Equation 2.4 is said to be stable if for

every $\epsilon \geq 0$, there exist a $\sigma \geq 0$ such that $\|x(0)\| \leq \sigma$ then $\|x(t)\| \leq \epsilon$

Definition The origin of a dynamical system of 2.4 is said to be asymptotically stable if it is Lyapunov stable and $\|x(t)\| \leq \sigma$ then $\lim_{t \rightarrow \infty} x(t) = 0$

Definition The origin of a dynamical system of 2.4 is said to be stable if it there exist positive constants a_0 and b_0 such that the solution $x(t)$ satisfies $\|x(t)\| \leq a_0 \|x(0)\| e^{-b_0 t}$

Theorem 1 Consider the autonomous system of Equation 2.4, with a critical point at the origin. If there exists a function $\Phi(x)$ that is

- continuous and has continuous partial derivatives;
- $\Phi(x)$ is positive definite about the origin; Then if

$$\frac{\Phi}{\delta t}(x) := \frac{\Phi}{\delta x_1} \frac{x_1}{\delta t} + \frac{\Phi}{\delta x_2} \frac{x_2}{\delta t} + \dots + \frac{\Phi}{\delta x_n} \frac{x_n}{\delta t} = \nabla \Phi \cdot F(x) \quad (2.5)$$

is negative semi-definite about the origin, the origin is a stable critical point for (2.4). If, $\frac{\Phi}{\delta t}(x) < 0; \forall x$ within a small ball about the origin, then the origin is an asymptotically stable critical point for 2.4.

A consequence of Theorem 1 is

Corollary 1.1 [6] For a system having a state vector x , consider the function

$V(x) : \mathfrak{R}^n \rightarrow \mathfrak{R}$ such that

$$V(x) \geq 0 \quad (\text{positive definite})$$

$$V(x) = 0 \quad \text{for } x = 0$$

$$\dot{V}(x) \leq 0 \quad (\text{negative definite})$$

$$\dot{V}(x) = 0 \quad \text{for } x = 0$$

Hence defining this function, then x goes to zero asymptotically and the system is stable.

Therefore using this theorem we design the guidance model for the GULMA UAV for the first time and consequently adopt the design for subsequent series of the UAV. The existing algorithms are robust to initial and final conditions, computationally fast, no restrictions are imposed on the magnitude of UAV thrust and effective in a UAV dynamic models [37].

Chapter 3

UAV Models

3.1 UAV Simulation Platform

In this Chapter we shall introduce two UAV models that were used in this project - the Ultra stick and GULMA; as well as point out some of the major features of these air vehicles. A detailed documentation of the Ultra stick UAV model-based design and development can be found in the theses by Yew Chai Paw [65] and Andrei Dorobantu [23] while for the GULMA UAV can be found in NAF UAV Technical report [1] and [13] by GULMA UAV program team.

3.1.1 Ultra stick UAV

This is a micro UAV that was developed by the University of Minnesota UAV program team, for model-based flight research [62]. This airborne test platform is capable of supporting research in areas of advance guidance, navigation and control as well as fault tolerant identification and isolation [25]. It is important to note that the ultra stick UAV has been tunnel tested.

The vehicle comprises of a standard fixed-wing airframe with ailerons, rudder, elevator and flap control surfaces. These control surfaces are actuated using electric servos with a maximum deflection of 25° . An electric powered motor

is used for the propulsion system that drives a fixed-pitch propeller. Shown in Figure 3.1 is a pictorial view of the airframe by Andrei Dorobantu [24].



Figure 3.1: Picture of the Ultra stick UAV adapted from [24].

3.1.2 GULMA UAV

The GULMA UAV was designed and developed by the Nigerian Air Force UAV program for military applications such as intelligence gathering, surveillance and reconnaissance (ISR) [1].



Figure 3.2: Picture of the NAF GULMA UAV.

The NAF GULMA UAV shown in Figure 3.2 was manufactured based on advanced engineering design geared towards objectives such as weight management, autonomous flight for beyond line-of-sight, real time video and telemetry data streaming. Some of the specific design objectives of the design team includes [1]:

1. General structure and aerodynamics requirements:
 - Composite structure for airframe
 - Payload of 10 kg with gross take-off weight of 40 – 50 kg
 - Fixed tricycle landing gear configuration
 - Internal fuel tank and equipment
 - Conventional high wing configuration for stability
2. General flight requirements:
 - Take-off and landing on paved ground
 - Cruise speed of 40 – 50 m/s
3. General propulsion requirements:
 - High performance internal combustion engine specification
 - Propeller driven with minimum diameter and high efficiency during cruise
 - High engine efficiency
 - High power to weight ratio

3.2 Non-linear Model

We shall describe methods selected for modelling of the GULMA UAV simulation model. These methods include the use of the conventional Matlab/Simulink 6-DOF aerodynamic toolbox and AVL modelling and aerodynamic analysis software.

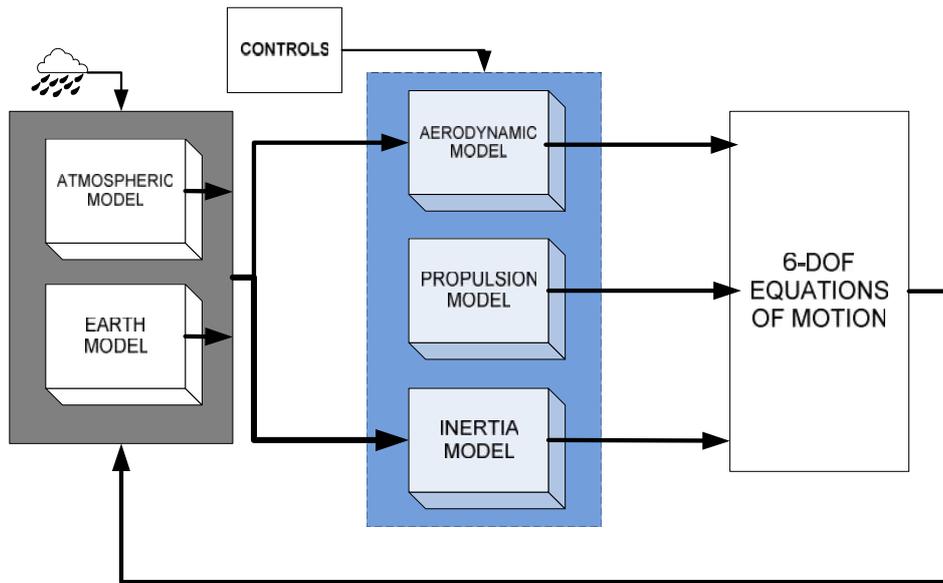


Figure 3.3: Diagram of the conventional fixed-wing non-linear model.

3.2.1 Flight Dynamic Modelling

The standard dynamic Equation of Motions (EOM) of the fixed-wing air vehicle is assumed for a rigid body of a given mass and moments of inertia. The forces and moments acting on the vehicle and causing translational and angular accelerations are of aerodynamic, propulsive and gravitational magnitudes. The aerodynamic forces mostly depend upon the external shape of the vehicle and different flight variables like dynamic pressure, angle of attack (α), angle of side-slip (β), vehicle's angular velocities and the control surface deflections, mass of aircraft (m) etc [16]. Hence, implementation was done using 6-degrees-of-freedom, fixed mass, flat non-rotating earth, rigid body EOM for simulation and modelling of the air vehicle dynamics. These EOM have been heavily documented in various modelling, simulation and experimental projects [16]. The derived equations for the UAV of mass(m) and thrust force (T) are described as follows:

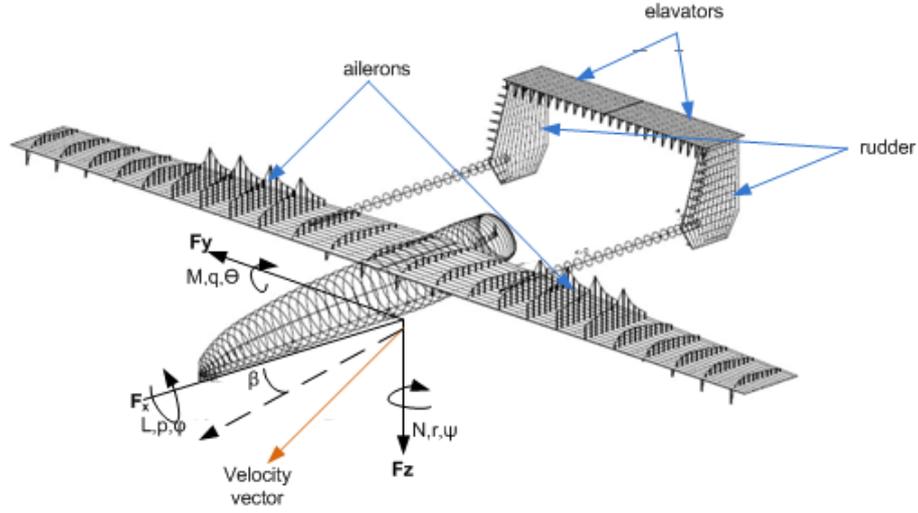


Figure 3.4: Diagram showing forces and moments acting on UAV.

Force equations:

$$\dot{u} = (rv - qw) + \frac{X}{m} - g \sin \theta + \frac{T}{m} \quad (3.1)$$

$$\dot{v} = (pw - ru) + \frac{Y}{m} + g \cos \theta \sin \phi \quad (3.2)$$

$$\dot{w} = (qu - pv) + \frac{Z}{m} + g \cos \theta \cos \phi \quad (3.3)$$

Moment Equations:

$$\dot{p} = \frac{I_{xz}}{I_x} r - qr \frac{(I_z - I_y)}{I_z} + qp \frac{I_{xz}}{I_x} + \frac{L}{I_x} \quad (3.4)$$

$$\dot{q} = pr \frac{I_{xz}}{I_x} - (p^2 - r^2) \frac{(I_{xz})}{I_y} + \frac{M}{I_y} \quad (3.5)$$

$$\dot{r} = \frac{I_{xz}}{I_z} \dot{p} - pq \frac{(I_y - I_x)}{I_z} - qr \frac{I_{xz}}{I_z} + \frac{N}{I_z} \quad (3.6)$$

Kinematic equations:

$$\dot{\phi} = p + \tan \theta (q \sin \phi + r \cos \phi) \quad (3.7)$$

$$\dot{\theta} = q \cos \phi - r \sin \phi \quad (3.8)$$

$$\dot{\psi} = \sec \theta (q \sin \phi + r \cos \phi) \quad (3.9)$$

These equations are a concise representation of a full aerodynamic model. The full non linear dynamics and identification were obtained from [61]. Standard nomenclature used for the states are: x-y-z body axis velocities in m/s (u, w, v), angular rates in rad/s (p, q, r), a standard 3 – 2 – 1 ordered rotation sequence of Euler angles in rad (ϕ, θ, ψ) as well as the aerodynamic forces and moments which are denoted as X, Y, Z and L, M, N respectively. The product of inertia are I_x, I_y, I_z in Kgm^2 and along the $x - z$ axis is I_{xz} assuming a symmetric model structure. The aerodynamic force have the following relationships in terms of air speed velocity, V_a , dynamic pressure \bar{d} and aerodynamic force coefficients:

$$X = \bar{d}SC_x \quad (3.10)$$

$$Y = \bar{d}SC_y \quad (3.11)$$

$$Z = \bar{d}SC_z \quad (3.12)$$

where the aerodynamic force coefficients, (C_x, C_y, C_z), and non-dimensional lift (c_l) and drag (c_d) coefficients are given as:

$$C_x = c_l \sin \alpha - c_d \cos \alpha \quad (3.13)$$

$$C_z = -c_d \sin \alpha - c_l \cos \alpha \quad (3.14)$$

$$C_y = c_{y\beta}\beta + c_{y\delta_r}\delta_r + b/2V_s(c_{yp}p + c_{yr}r) \quad (3.15)$$

$$c_l = c_{l_0} + c_{l_\alpha}\alpha + c_{l_{\delta_e}}\delta_e + c/2V_a(c_{l_\alpha}\dot{\alpha} + c_{l_q}q) \quad (3.16)$$

$$c_d = c_{d_0} + c_{d_{\delta_e}}\delta_e + c_{d_{\delta_r}}\delta_r + (c_l - c_{l_{min}})/\pi e \quad (3.17)$$

The non-dimensional coefficients (C_l, C_m, C_n) that form the moment equa-

tions are also given as:

$$L = \bar{d}SbC_l \quad (3.18)$$

$$M = \bar{d}SbC_m \quad (3.19)$$

$$N = \bar{d}SbC_n \quad (3.20)$$

$$C_l = C_{l_\beta}\beta + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r + b/2V_s(C_{l_p}p + C_{l_r}r) \quad (3.21)$$

$$C_m = C_{m_0} + C_{m_\alpha}\alpha + C_{m_{\delta_e}}\delta_e + c/2V_s(C_{m_\alpha}\dot{\alpha} + C_{m_q}q) \quad (3.22)$$

$$C_n = C_{n_\beta}\beta + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r + b/2V_s(C_{n_p}p + C_{n_r}r) \quad (3.23)$$

where c , b and S is mean chord length, air vehicle wing span and wing area respectively.

3.2.2 Calculation of Aerodynamic Coefficients

The wind tunnel testing method was used for estimating the aerodynamic coefficients of the Ultra stick UAV [25]; while we use the AVL method for computation and estimation of aerodynamic coefficients for GULMA UAV as well as its corresponding stability derivatives. This method requires specification of performance details of the aircraft, aerodynamic analysis, weight distribution matrix and trim analysis for successful computation. Furthermore, in this method the lifting surfaces are specified in terms of horse-shoe vortices in both span-wise and chord-wise directions. The aircraft geometrical configuration and mass are also required for numerical calculation of the stability derivatives. The software calculates induced drag prediction, hence drag computation was done numerically using first principles as in Equation 3.17 [80] in addition to induced drag obtained from AVL, however important to note that using XFOIL software it is

also possible to obtain initial estimate of parasitic drag. A detailed description is detailed in Appendix A. Results for the aerodynamic coefficients in radians from the AVL software are specified in Table 3.1.

Table 3.1: Aerodynamic Coefficients.

$cl_0 = 0.194$	$cl_\alpha = 6.2821$	$cl_{\delta_e} = 0.097$
$cl_\alpha = 0.021$	$cl_p = -14.2618$	$cl_{min} = 0.23$
$cd_0 = 0.0345$	$c_{\delta_e} = 0.017$	$c_{\delta_r} = 0.0153$
$cy_\beta = -0.4691$	$cy_{\delta_r} = 0.1913$	$cy_p = -0.1092$
$cy_r = 0.2646$	$cl_\beta = -0.0545$	$cl_{\delta_a} = -0.0067$
$cl_p = -0.6523$	$cl_r = 0.3213$	$cm_a = -1.6067$
$cm_{\delta_e} = -0.0232$	$cm_q = -15.3517$	$cn_\beta = 0.0510$
$cn_{\delta_r} = -0.0100$	$cn_p = -0.0003$	$cn_r = -0.0755$

3.2.2.1 Performance Analysis

In this subsection the critical performance specifications of both UAV-Ultra stick and GULMA are compared with a view to analyse their requirements for modelling. In Table 3.2 a comparative analysis was done for the UAV configurations.

Table 3.2: GULMA and Ultrastick UAV Configuration.

Serial	Characteristics	GULMA	Ultra stick	Units
1.	wing span	3.8	1.2	m
2.	Fuselage Length	1.4	1.05	m
3.	Mean aerodynamic chord	0.38	0.3	m
4.	Gross take-off weight	47.0	1.9	kg
5.	Max. payload weight	4	nil	kg
6.	wing reference area	1.3	0.32	m^2
7.	cruise speed	44.5 - 50	10 - 23	m/s
6.	stall speed	20.6	8	m/s

Table 3.2 clearly indicates that the GULMA UAV being more than 70% larger in size and weight would require increased control gains; also the GULMA UAV

has a symmetric structure while the Ultra stick UAV has a high angle-of-attack static structure.

3.2.2.2 Weight Analysis

The following information was obtained from the CAD model using CATIA program and reference documents [1]. The initial performance parameters the aircraft has the following weights:

- Overall take-off weight: 47.0kg
- Empty weight: 32.1kg
- Fuel weight: 14.9kg

The first weight breakdown can be seen in Figure 3.5a including estimated structural, systems, avionics and fuel weights.

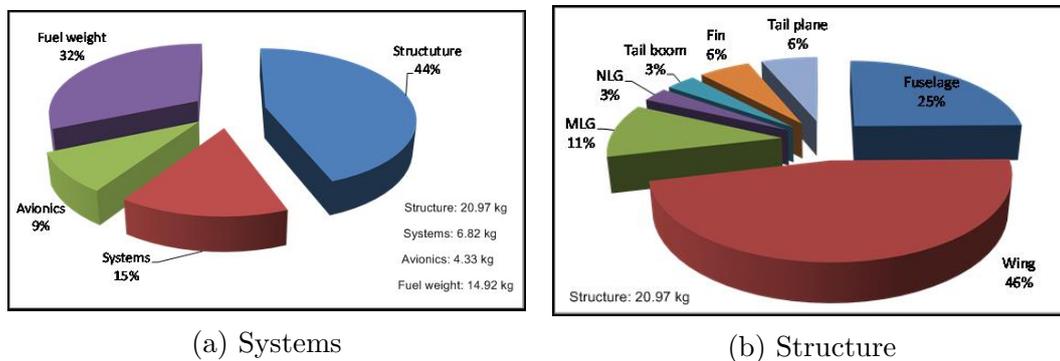


Figure 3.5: Breakdown of weight of GULMA UAV sub-systems.

The second breakdown can also be seen in Figure 3.5b for the sub-components of the structural section.

For the avionics section, the breakdown is shown in Figure 3.6.

The following graph shows the position of the critical components, as well as the location of the centre of gravity at gross weight take-off and aircraft neutral point was calculated using the AVL program.

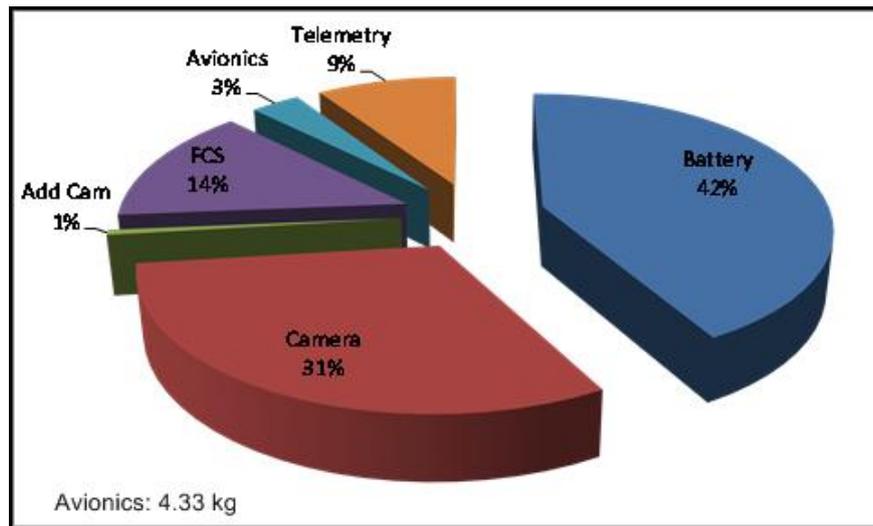


Figure 3.6: Breakdown of weight of GULMA UAV avionics.

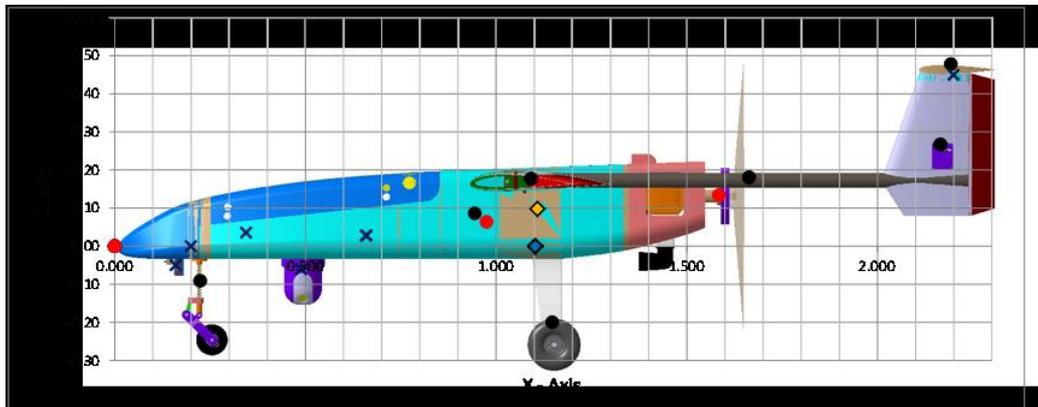


Figure 3.7: Positions of critical components on the GULMA UAV.

Structural components is distinguished by a ‘black dot’, systems by a ‘red dot’ and avionics by a ‘cross’. The location of the centre of gravity (cg) is shown in Figure 3.7 by the yellow square. The product of moment of inertia are subsequently computed and loaded in the AVL mass file.

3.2.2.3 Trim Analysis

As shown in Figure 3.8 the two values of the static margin show a tendency of making the aircraft stable from the operational take-off weight to the empty weight of the aircraft. Although the moment coefficient at zero lift and zero deflections is negative it is necessary to evaluate if the aircraft possess enough

elevator authority for the most critical conditions [80]. The trim analysis is performed for the most forward and aft cg positions.

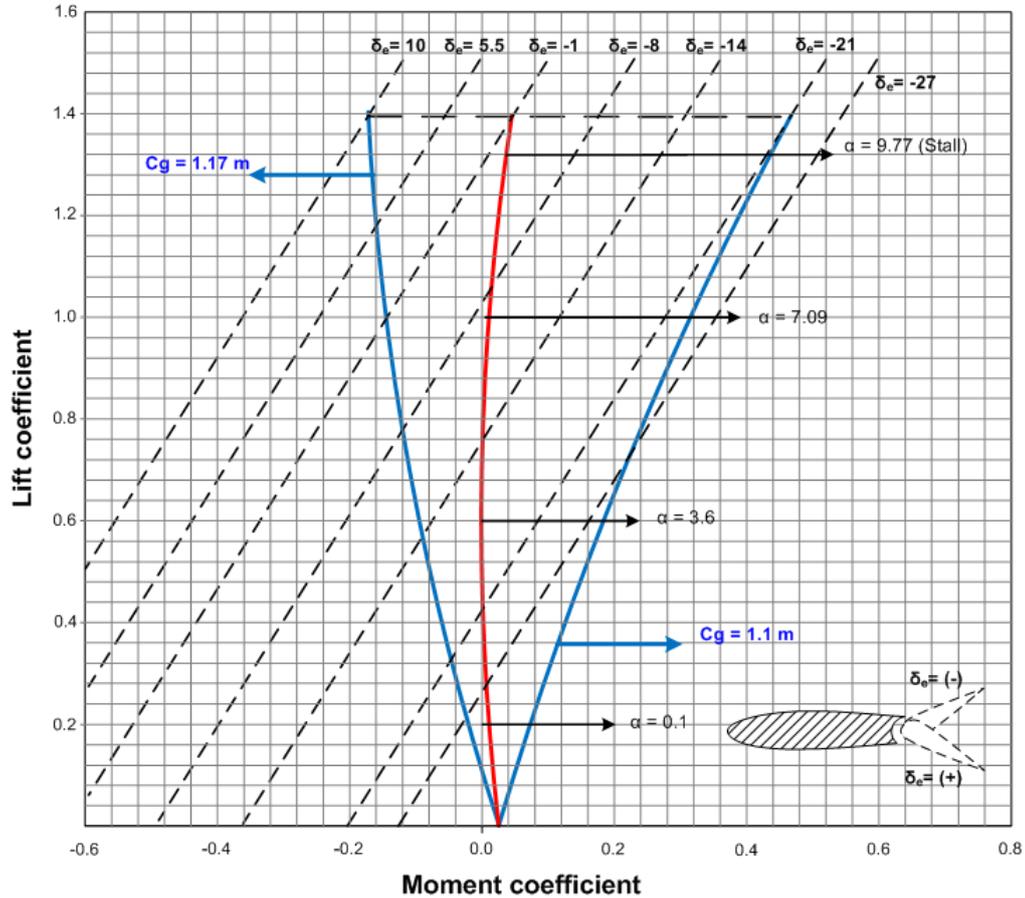


Figure 3.8: GULMA UAV lift versus moment coefficients for varying AOA.

The trim analysis indicates that it is necessary for the elevator to operate between 10° to -27° degrees in order to satisfy trim state during the complete flight phase regardless of the angle of attack or lift coefficient. The blue lines indicate the current boundaries of centre-of-gravity excursion on the aircraft. Furthermore, the red line corresponds to non-variation of moment coefficient which indicates that the centre-of-gravity and neutral point are located at the same distance. The slightly non linear variation of the lines corresponds to a small deviation of the neutral point at different lift coefficients. We note that at all points within the boundaries, the total moment coefficient is zero.

3.2.3 Inertia Model

The inertia model comprises of geometric aircraft mass, centre-of-gravity and moment of inertia coefficients. This is presented as a moment of inertia matrix equation [65]:

$$I = \begin{bmatrix} I_{xx} & 0 & -I_{xz} \\ 0 & I_{yy} & 0 \\ -I_{zx} & 0 & I_{zz} \end{bmatrix}$$

In this equation we assume the the UAV is symmetrical therefore, $I_{xy} = I_{yx} = I_{xz} = I_{zy} = 0$ and also assume propulsion system moment of inertia are minimal. The results for the moments of inertia were calculated using the GULMA UAV mass spreadsheets presented in Appendix A.

3.2.4 Propulsion Model

Preceding take-off from the ground, an aircraft requires thrust — either from an electric motor or a gasoline engine. The size and type of engine was determined by the size and type of aircraft. The objective of propulsion system design therefore was to ensure large power output per unit of engine size and weight. The engine thrust represents a concentration of weight, inertia, and gyroscopic forces that require additional structural weight in the aircraft. A medium altitude long-endurance (MALE) UAV [1] such as the GULMA UAV could be powered by piston or turbine engine, electric, glow or solar. Using the data obtained from 219 MALE UAV as shown in the Figure 3.9, piston power generation methods accounts for about 60% of these UAV. Hence the results of the survey lead to the adoption of piston power generation method for GULMA UAV. In line with

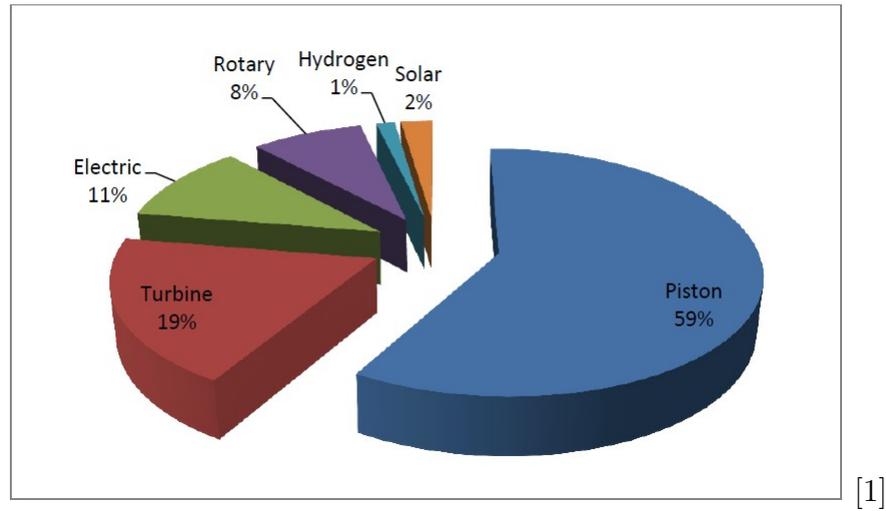


Figure 3.9: Power generation methods for 219 MALE/HALE UAVs. [1]

the conservation of angular momentum, the propulsion dynamics is given by:

$$(I_{piston} + I_{prop})\dot{\omega}_p = T_{piston} - T_{prop} \quad (3.24)$$

where:

- I_{piston} =moment of inertia of piston engine (kgm^2)
- I_{prop} =moment of inertia of propeller (kgm^2)
- T_{piston} =output torque of piston shaft (Nm)
- T_{prop} =output torque of propeller (Nm)
- $\dot{\omega}_p$ =speed of rotation of propeller ($\frac{m}{s}$)

3.2.4.1 Piston Engine

Further to selection of piston engine as the source of propulsion for the GULMA UAV, next task was to size the required engine for the vehicle. The particular requirement which determines the engine size was dictated by both engine and airframe characteristics mentioned in subsection 3.2.1. Two basic methods are

S/N	UAV model	Power (hp)	MTOW (kg)	Wing span (m)	Country of origin
(a)	(b)	(c)	(d)	(e)	(f)
1.	Backpack UAV	1.6	5.44	1.22	USA
2.	Javelin	2	9	2.44	USA
3.	X-Sight	3.5	16	2.6	Germany
4.	Half-Scale	4.5	18.1	2.5	USA
5.	ALO	6.5	20	3.03	Spain
6.	Heli 25	7.5	25	2.4	Israel
7.	K-100	6	28	2.6	France
8.	Luna	8	30	4.17	Germany
9.	Mercury	6	30	2.5	UK
10.	Yarará	8	30	3.98	Argentina
11.	Pelican Observer	17	36.3	7.05	Australia
12.	Aerolight	10	40	4	Israel
13.	Canard	10	40	2.4	Israel
14.	Sheddon MK3	10	40	4.2	Israel
15.	Mini-Vanguard	17	45	2.16	USA
16.	Lipan M3	17.8	60	4.38	Argentina
17.	Apid MK IV A	16	62	3	Sweden
18.	Aerosky	18	70	4.5	Israel
19.	Chacal 2	24	75	2.6	France
20.	Raven	21	84	3.6	UK
21.	Apid MK IV B	28	90	3	Sweden
22.	RPG Midget MK III	24	90	2.6	Sweden
23.	Aerohawk	40	100	3	Israel

Figure 3.10: Data for 23 UAVs similar to GULMA UAV.

involved in engine sizing namely: the empirical and the statistical methods [1] which are detailed in Appendix A. By using empirical method and applying the statistical solution of Figure 3.10 we get an estimated engine power rating of 13.5 hp and 10 hp respectively.

Considering that Aerolight, Canard and Sheddon MK3 UAV, all of which are in the same weight category as the GULMA UAV, have 10 hp each; the statistical solution indicates that it is closer to the realistic cases than the empirical method. Due to safety and reliability reasons, the engine and propeller efficiency have to be accounted for. Assuming a conservative propeller efficiency, of 80% and estimated engine efficiency of 94%, the required engine power for the GULMA UAV was estimated to be 13 hp. Given, maximum velocity of 43 m/s, gross take-off weight of 48 kg, the required thrust is 125N. Therefore, the thrust to weight ratio is given by:

$$\frac{T}{W} = \left(\frac{n_p}{V}\right) \left(\frac{P}{W}\right) = 0.32 \quad (3.25)$$

3.2.4.2 Calculations for Engine Thrust

In a traditional conceptual/preliminary design case, power is estimated as a function of variables - MTOW and wingspan of the aircraft [1]. These variables are used as standard because of their direct relationship to lift. Figure 3.11 is a graphical description for the formulation of the power function of the aircraft.

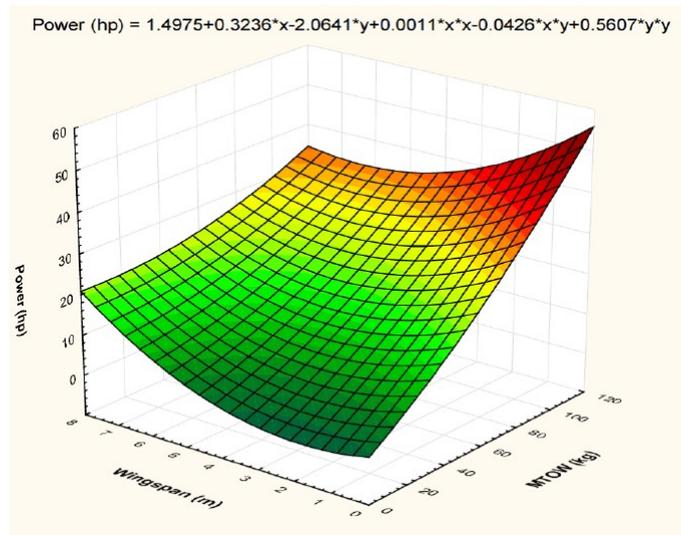


Figure 3.11: Graphical representation of the engine power as a function MTOW and wingspan

Based on the solutions to engine sizing methods previously described, in Figure 3.12 is a picture of various engine plants that were considered for the GULMA UAV.

3.2.5 Summary

In this chapter we present the Ultra stick UAV model that will be used as a test platform to develop the GULMA UAV model. Using the AVL program we developed the GULMA UAV linear model using the aircraft configuration parameters. The modelling procedure included the creation of the mass distribution sheets to calculate the moments of inertia matrix. Aerodynamic analysis is conducted by loading the model and calculating for aerodynamic derivatives at specified flight



Figure 3.12: Engine models considered for use on the GULMA UAV.

conditions for velocity, lift coefficient, and atmospheric conditions. The resulting trim results and stability derivatives are thus computed. Eigenmode analysis is conducted for this flight condition where the results for the linear models are analysed as described in Appendix A-E. The results are further verified using the Ultra stick UAV Matlab/Simulink model.

A graphical representation for the trimmed UAV over the velocity range of 34 – 43 m/s at flight level 100 m/s is shown in Figure 3.13. The results indicate some linearization points were achieved for velocity (35, 37, 39, 40) m/s. While at linearization points (40 – 43) m/s we achieve trimmed flights for zero elevator angle and angle-of-attack. Linearization for ranges outside these values was not achieved because the coefficient of thrust for engine model were obtained from first principles and linearization process.

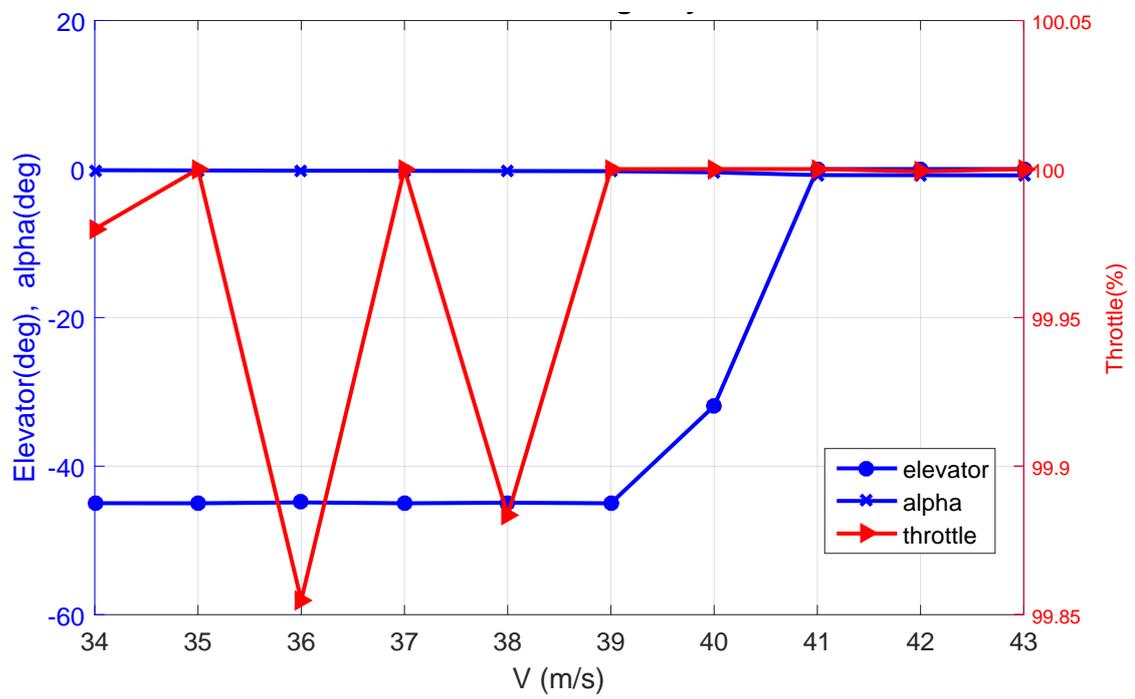


Figure 3.13: Trimmed level flight by velocity of the GULMA UAV.

Chapter 4

Control and Optimisation

4.1 Linear Quadratic Control Design

Designing a robust flight controller system is essential for any unmanned aerial vehicle which enables it to efficiently execute a flight mission especially in the presence of some magnitude of disturbance [27]. The use of advance modern control techniques guarantees efficient robust flight performance for most UAV platforms than would most classical control approaches. These modern flight control techniques such as dynamic inversion, model predictive control, H-infinity, to name a few have been widely used to design high performance controllers for UAV systems [8]. In this chapter our focus is on the design and application of an optimal linear controller using a multi-objective particle swarm optimization algorithm for tuning of the weighting matrices and to perform a model-based controller synthesis of the GULMA UAV linear model. The objective of the controller is to optimally track any lateral-directional trajectories of the UAV. The MOPSO algorithm optimizes the selection of tuning matrices necessary to obtain state-feedback optimal control gains for tracking of any input or reference sequence. We shall also briefly describe the design and analysis for an optimal controller for the UAV longitudinal plant with model uncertainty. The designed

controllers are simulated and demonstrated using Matlab/Simulink software [56].

The non-linear dynamic model described in Chapter 3 was linearized around the trimmed operating conditions. Each variable in the non-linear model described in EOM previously stated are now assumed to operate at a nominal value, u_0 , and added to a control perturbation or disturbance value, δu , on the trimmed flight condition such that we have $u = u_0 + \delta u$. The Full linearized state space system is shown in Appendix B and is represented as:

$$\begin{aligned}\dot{\tilde{x}} &= A\tilde{x} + B\tilde{u} \\ y &= C\tilde{x} + D\tilde{u}\end{aligned}$$

where

$$\begin{aligned}\tilde{x} &= \begin{bmatrix} u & v & w & \phi & \theta & \psi & p & q & r \end{bmatrix}^T \\ \tilde{u} &= \begin{bmatrix} \delta_e & \delta_a & \delta_r & \delta_\tau \end{bmatrix}^T\end{aligned}$$

Table 4.1 is a list of the trim condition values used to obtain the full linearized dynamics equation.

4.1.1 Longitudinal Control Design

The objective of this subsection is to design and analyze a LQR-based longitudinal-axis controller using the developed GULMA UAV model. This is necessary as controlled longitudinal axis guarantees that the aircraft maintains a flight condition in the presence of disturbances as well as high performance missions such as auto-landing [53]. We shall therefore consider using the LQG technique in order to test and predict for longitudinal robustness of the control system.

Consequently, with respect to forces X and Z and the moment equation M

Table 4.1: Trim Conditions.

Serial	Variable	Value	Units
1.	u	43	m/s
2.	v	-0.0859	m/s
3.	w	-0.56	m/s
4.	ϕ	0.0063	rad
5.	θ	-0.329	rad
6.	ψ	1.57	rad
7.	p	0	rad/s
8.	q	0	rad/s
9.	r	0	rad/s
10.	δ_e	-0.20	rad/s
11.	δ_e	-0.0001	rad/s
12.	δ_r	0.00062	rad/s
13.	δ_τ	0.45	rad/s

mentioned in Chapter 3; which are the aerodynamic forces and moment that depend upon different flight variables (aerodynamic coefficients) and the control actuation of the elevator (δ_e) and throttle (T) the longitudinal state space model after linearization at velocity = 43m/s was calculated and obtained as:

$$\dot{x}_{lon} = A_{lon}x_{lon} + B_{lon}u_{lon} \quad (4.1)$$

$$y_{lon} = Cx_{lon}$$

$$A_{lon} = \begin{bmatrix} X_u & X_w & X_q - W_e & -g\cos\theta_e \\ Z_u & Z_w & Z_q + U_e & -g\sin\theta_e \\ M_u & M_w & M_q & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad B_{lon} = \begin{bmatrix} X_{\delta_e} \\ Z_{\delta_e} \\ M_{\delta_e} \\ 0 \end{bmatrix} \quad (4.2)$$

where the control vector $u_{lon} \in \mathfrak{R}^{2 \times 1}$, state vector $x_{lon} \in \mathfrak{R}^{5 \times 1}$ and output vector $y_{lon} \in \mathfrak{R}^{2 \times 1}$ are given by:

$$u_{lon} = \begin{bmatrix} \delta_e & \text{-elevator deflection} & (\text{rad/s}) \\ \delta_t & \text{-thrust} & (\text{rpm}) \end{bmatrix}$$

$$x_{lon} = \begin{bmatrix} u & \text{-forward speed} & (\text{m/s}) \\ w & \text{-vertical speed} & (\text{m/s}) \\ q & \text{-pitch rate} & (\text{rad/s}) \\ \theta & \text{-pitch angle} & (\text{rad}) \\ h & \text{-Altitude} & (\text{m}) \end{bmatrix} \quad y_{lon} = \begin{bmatrix} q & \text{-pitch rate} & (\text{rad/s}) \\ \theta & \text{-pitch angle} & (\text{rad}) \end{bmatrix}$$

where, $h = -Z_e$ inertial down position of the vehicle. we compute the linearized longitudinal state space values as:

$$A_{lon}x = \begin{bmatrix} -0.3548 & 0.2129 & -0.5007 & -9.373 & -0.0003 \\ -0.4274 & -6.6470 & 41.64 & 2.8813 & -0.0005 \\ -0.0292 & -1.8480 & -2.4470 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0.2939 & 0.9557 & 0 & -41.24 & 0 \end{bmatrix} \begin{bmatrix} u \\ w \\ q \\ \theta \\ h \end{bmatrix} \quad (4.3)$$

$$B_{lon}u = \begin{bmatrix} 0.3626 & 0 \\ -0.345 & 0 \\ -0.8884 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_\tau \end{bmatrix} \quad (4.4)$$

$$y_{lon} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix} \theta$$

The poles of the system are the eigenvalues of the matrix A_{lon} [29]. The poles and transmission zeros of the UAV in the level flight condition mentioned are given in Table 4.2. Generally, the longitudinal dynamics of an aircraft possesses two natural modes-the phugoid (long-period) mode and the the short-period mode given as follows:

Table 4.2: Characteristic values of longitudinal modes.

Mode(1, 2)	$\lambda_{1,2} = -0.107 \pm 0.242i$	$\zeta_{pm} = 0.404$	phugoid mode
Mode(3, 4)	$\lambda_{3,4} = -4.52 \pm 8.51i$	$\zeta_{sp} = 0.469$	short-period
Transmission Zeros	$z_1 = -7.42$	$z_2 = -0.000127$	

The phugoid mode is conventionally a low-frequency, lightly damped mode which is associated with the gradual interchange of potential and kinetic energy about the equilibrium altitude and airspeed - at a nearly constant angle-of-attack [16]. The frequency of oscillation and the damping ratio are inversely proportional to the forward speed. When the UAV is being flown instrument flight rules (IFR) are used, therefore low phugoid damping becomes highly undesirable. This mode is generally slow enough so that an external pilot (remotely piloted) can easily negate the disturbance by small control movement. Even though the pilot can correct for the phugoid mode it would become extremely fatiguing if the damping were too low. An automatic stabilization system is used to improve the damping of the phugoid mode. For the GULMA UAV this mode is associated with complex conjugate poles at $\lambda_{1,2} = -0.107 \pm 0.242i$. The damping ratio ζ_{pm} is 0.404, while the frequency ω_n is $0.265rad/sec$ and the time constant is $9.34sec$. The short period mode is a high frequency, damped oscillation, which is associated with nearly constant speed. We see that the high frequency mode is highly damped, so the UAV would respond rapidly to the elevator input without any undesirable overshoot. This scenario is as examined in [16]. On the other hand, when this mode is lightly damped, it is very difficult

to control the aircraft. For the GULMA UAV aircraft, this mode is associated with complex conjugate poles at $\lambda_{3,4} = -4.52 \pm 8.51i$, damping ratio ζ_{sp} is 0.469, natural frequency w_n is 9.63 and time constant of 0.221sec.

Now considering the matrix of DC gains for this model we have,

$$\begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} = \begin{bmatrix} 32.6 & 37.8 \\ -3.95 & 1.76 \\ -18.7 & -22.3 \\ -10.4 & -5.61 \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_\tau \end{bmatrix} \quad (4.5)$$

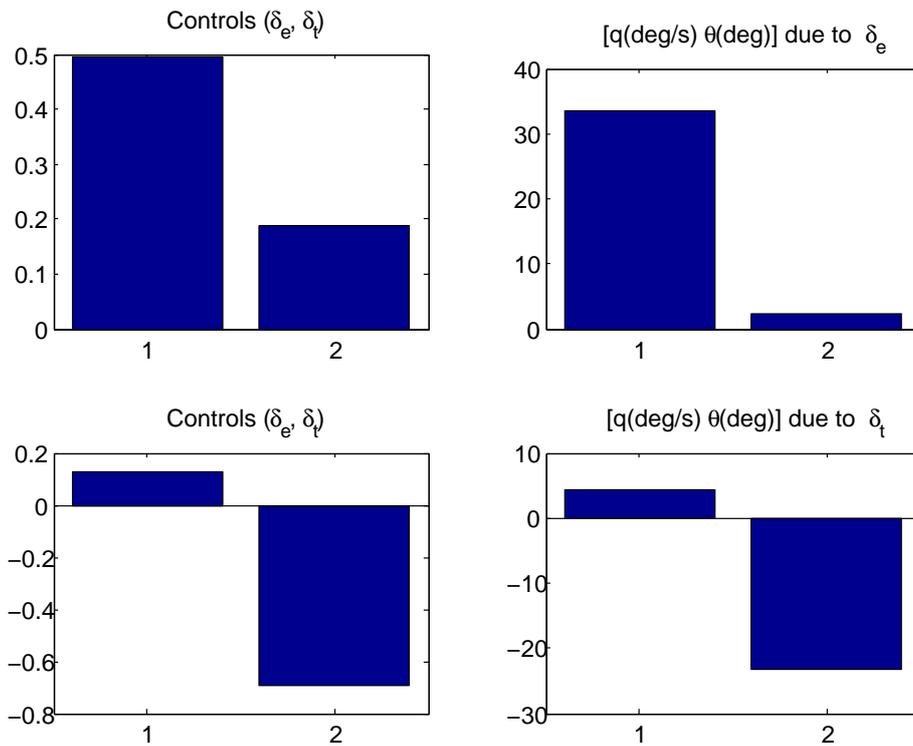


Figure 4.1: Visualization of GULMA UAV singular value decomposition.

This matrix shows that a negative elevator deflection ($\delta_e \lesssim 0$ -elevator deflects upward) causes the aircraft to develop a pitch upward motion ($\theta \gtrsim 0$) and slow speed of ($u \lesssim 0$), while maintaining steady cruise (flight path angle $\rightarrow 0$). This simple steady state analysis shown in Figure 4.1 suggests that in order to

maintain steady cruise conditions both elevator angle and engine constant *rpm* must contribute continuously for this phase of flight.

A singular value decomposition of the plant at DC yields the following results:

$$G(j0) = C(-A)^{-1}B \quad (4.6)$$

$$\nu\Sigma\nu^T = \begin{bmatrix} 0.49 & 0.13 \\ 0.19 & -0.69 \end{bmatrix} \begin{bmatrix} 58.84 & 0 \\ 0 & 5.88 \end{bmatrix} \begin{bmatrix} -0.66 & 0.75 \\ -0.75 & -0.66 \end{bmatrix} \quad (4.7)$$

We shall now examine the longitudinal response of the model to an initial pitch angle disturbance of 1° . This now results in a phugoid mode that comprises of a slow oscillation forward speed, u , at a nearly constant pitch angle ($\theta = 0$). The small damping in this mode is due to a small magnitude of the stability derivative for change in drag due to change in u . A plot for the longitudinal response to an initial pitch angle disturbance is as shown in Figure 4.2.

This gives the indication that an increase damping is required in the long-period to remove the slow airspeed oscillation and improve overall control of the aircraft longitudinal performance [29]. We shall now therefore design a LQR-based longitudinal controller that guarantees improved and robust performance for the GULMA UAV longitudinal axis.

The LQG control is a modern state-space technique for designing optimal dynamic regulators and servo controllers with integral action (also known as set point trackers) [51]. This technique allows for a trade-off in reference tracking and/or regulator performance and control effort, as well as to take into account process disturbances and measurement noise [85]. Hence, a combination of the LQ regulator and the Kalman filter enables robustness of the controlled system. The LQ optimal gain is selected using Equation 4.26 and the a block diagram of the control system is described in Figure 4.3 and the integral action subsystem

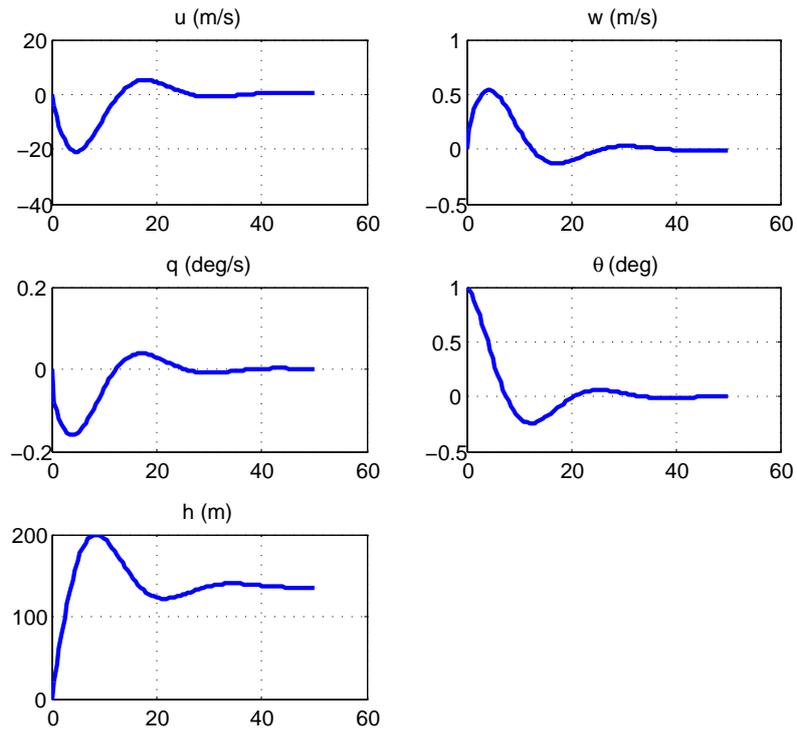


Figure 4.2: Long period longitudinal response of GULMA UAV to an initial pitch disturbance of 1° .

in Figure 4.4.

We shall now consider the linear model described in equation with elevator deflection as input variable. It is normal to control the longitudinal autopilot with either the elevator and/or throttle at a time but not together as described in [82]. Hence adopting the method described in [56] we have:

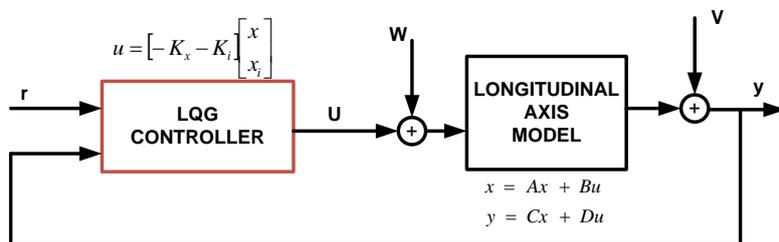


Figure 4.3: LQG controller block diagram for GULMA UAV.

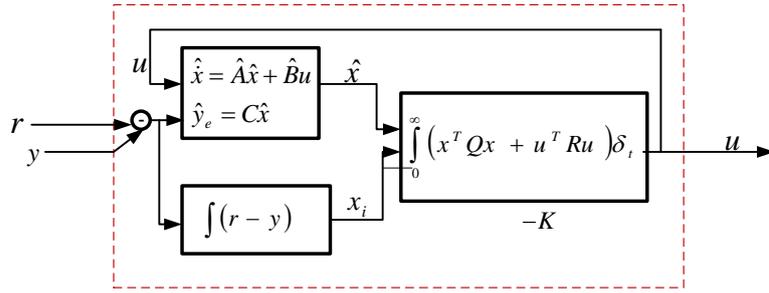


Figure 4.4: LQ integral control and Kalman filter subsystem.

$$\dot{x} = Ax + Bu + Gw \quad (4.8)$$

$$y = Cx + Du + Hw + k \quad (4.9)$$

where G is the plant noise covariance matrix, w is the plant process noise, H is measurement noise matrix of the output and k is the measurement noise.

Now to design the LQG controller, the first step is calculate the LQ optimal gains for the longitudinal plant to formulate a tracking problem and to add the integral state, x_i to Equation 4.1 so that we have [76]:

$$\begin{bmatrix} \dot{x} \\ \dot{x}_i \end{bmatrix} = \begin{bmatrix} A_{lon} & 0 \\ -C_y & 0 \end{bmatrix} \begin{bmatrix} x \\ x_i \end{bmatrix} + \begin{bmatrix} B_u \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ I \end{bmatrix} r \quad (4.10)$$

$$x_i = r - y$$

also $\bar{x} = \begin{bmatrix} x & x_i \end{bmatrix}^T$,

where the optimal feedback for the cost and corresponding gain are given as:

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

$$u = - \begin{bmatrix} K & K_i \end{bmatrix} \begin{bmatrix} x \\ x_i \end{bmatrix} = -\bar{K}\bar{x} \quad (4.12)$$

This gives us the closed loop dynamics for the LQ optimal solution for set-point tracking as,

$$\begin{bmatrix} \dot{x} \\ \dot{x}_i \end{bmatrix} = \begin{bmatrix} A_{lon} - B_{lon}K & -B_{lon}K_i \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ x_i \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} r \quad (4.13)$$

Now similar to the plant optimal gain we add the integral control for the Kalman estimator to obtain robust tracking performance. The conventional Kalman filter representation is given as [56]:

$$\dot{\hat{x}} = A_{lon}\hat{x} + B_{lon}u + L(y - \hat{y}) \quad (4.14)$$

$$= (A_{lon} - LC)\hat{x} + B_{lon}u + Ly \quad (4.15)$$

$$\dot{x}_i = r - y$$

$$u = - \begin{bmatrix} K & K_i \end{bmatrix} \begin{bmatrix} \hat{x} \\ x_i \end{bmatrix}$$

$$x_f = \begin{bmatrix} x^T & x_i^T \end{bmatrix}^T$$

Hence we now have the closed loop system for the Kalman estimator

$$\dot{x}_f = \begin{bmatrix} A_{lon} - LC - B_{lon}K & -B_{lon}K_i \\ \hline 0 & 0 \end{bmatrix} x_f + \begin{bmatrix} L & I \end{bmatrix} y + \begin{bmatrix} 0 \\ I \end{bmatrix} r \quad (4.16)$$

$$u = -\bar{K}x_f \quad (4.17)$$

We now combine the resulting closed loop system from equation

$$\begin{bmatrix} \dot{x} \\ \dot{x}_f \end{bmatrix} = \begin{bmatrix} A_{lon} & & & \\ & -B_{lon}\bar{K} & & \\ \begin{bmatrix} LC \\ -C \end{bmatrix} & & \begin{bmatrix} A_{lon} - LC - B_{lon}K & -B_{lon}K_i \\ \hline 0 & 0 \end{bmatrix} & \end{bmatrix} \begin{bmatrix} x \\ x_f \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ I \end{bmatrix} r \quad (4.18)$$

$$y = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x \\ x_f \end{bmatrix}$$

4.1.2 Simulation and Analysis

Considering the longitudinal model, we have the following trim conditions used for linearizing the non-linear model listed in Table 4.1 we get the results for a pitch attitude autopilot for the GULMA UAV. The Q and R matrices were tuned according to Bryson's rule [80] so that we have:

$$Q = \begin{bmatrix} \frac{1}{\delta_{umax}^2} & 0 & 0 & 0 \\ 0 & \frac{1}{\delta_{wmax}^2} & 0 & 0 \\ 0 & 0 & \frac{1}{\delta_{qmax}^2} & 0 \\ 0 & 0 & 0 & \frac{1}{\delta_{\theta max}^2} \end{bmatrix}$$

$$R = \begin{bmatrix} 1 \\ \delta_e^2 \end{bmatrix}$$

where, $\delta_{umax} = 45$ m/s, $\delta_{wmax} = 16$ m/s, $\delta_{qmax} = 45$ deg/s, $\delta_{\theta max} = 45$ deg and $\delta_{emax} = 40$ deg.

We also create an uncertain longitudinal plant to design the controller for the variability of $\pm 10\%$, furthermore we assume that the process noise of 2 deg/s in angular rate channels and 0.8 m/s in the acceleration channels [51]. Hence for pitch attitude autopilot we have results shown in Figure 4.5 for the closed loop plant with LQG controller and in Figure 4.6 for the uncertain plant.

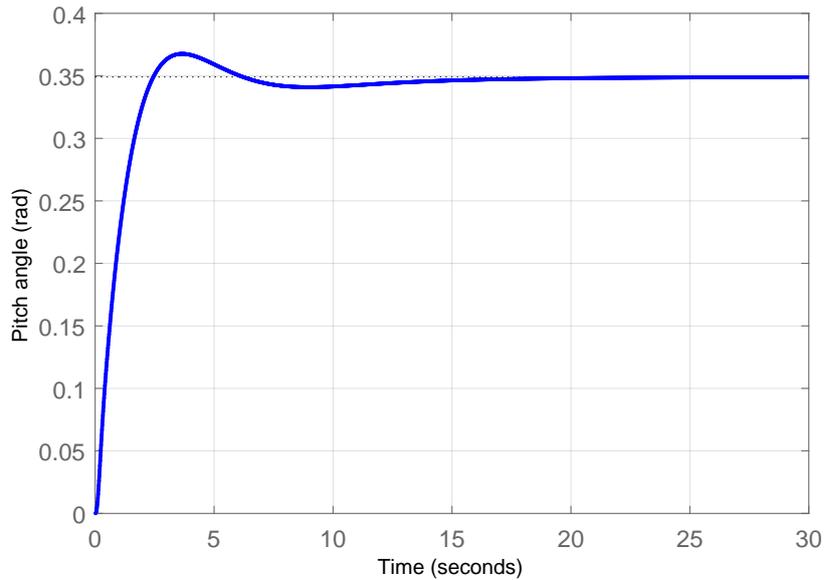


Figure 4.5: Closed-loop pitch response to 0.35 rad (20°) reference input.

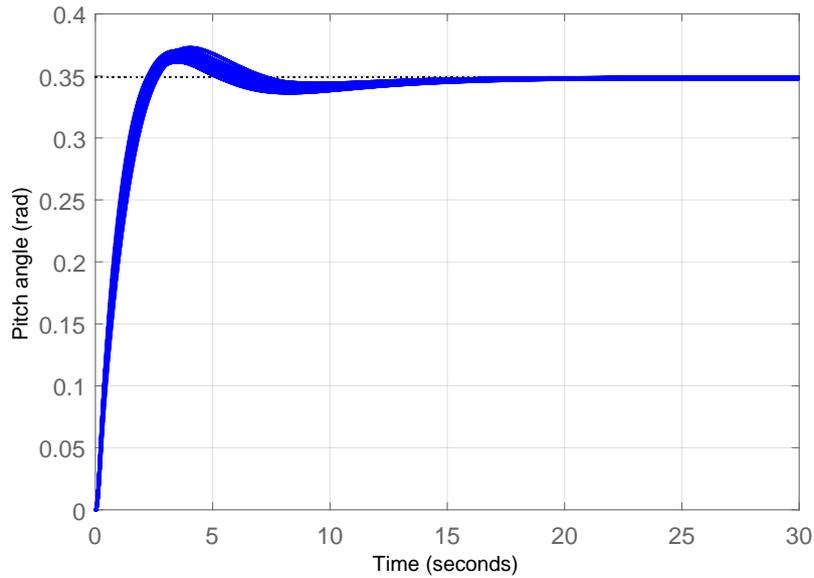


Figure 4.6: Closed-loop pitch response with $\pm 10\%$ variability.

The nominal results indicates a rise time of 1.63 seconds, steady state error of 0.063 and settling time of 10.4 seconds. The results were obtained from tuning of the Q and R matrices to guarantee optimized performance which was time consuming since it involved trial and error. Therefore for faster results for the lateral model we shall use a multi objective optimization algorithm to reduce computation time and optimally select the matrices that guarantees optimized solutions for tracking the reference input.

4.1.3 Lateral-directional Control Design

The control of the lateral dynamics of the UAV is considered in this subsection due to existing roll-yaw coupling dynamics in the model [82]. In this design, the roll angle is controlled to allow the system to follow a command signal as described in Figure 4.7. The first objective here is to control the roll-angle while aiming for well-damped responses and no steady state error. To achieve well damped responses, a multi-objective fitness function based PSO algorithm is used for tuning of the Q and R matrices for the LQR controller. Model reduction to obtain the dutch-roll mode parameters from the linear model is

required, hence we begin with the selection of the states we want to feedback; roll angle, yaw angle, as well as roll and yaw rates which allow for good damping and its control surfaces, aileron (δ_a) and rudder (δ_r) respectively. The linearized lateral state space dynamics is given as:

$$\dot{x} = A_{lat}x + B_{lat}u_{lat} \quad (4.19)$$

$$u_{lat} = \begin{bmatrix} \delta_a & \text{-aileron deflection} & \text{(rad/s)} \\ \delta_r & \text{-rudder deflection} & \text{(rad/s)} \end{bmatrix}$$

$$x_{lat} = \begin{bmatrix} v & \text{-sideslip speed} & \text{(m/s)} \\ p & \text{-roll rate} & \text{(rad/s)} \\ r & \text{-yaw rate} & \text{(rad/s)} \\ \phi & \text{-roll angle} & \text{(rad)} \\ \psi & \text{yaw angle} & \text{(rad)} \end{bmatrix} \quad y_{lon} = \begin{bmatrix} q & \text{-pitch rate} & \text{(rad/s)} \\ \theta & \text{-pitch angle} & \text{(rad)} \end{bmatrix}$$

where we have the stability derivatives (described in Appendix B.4) values as:

$$A_{lat}x = \begin{bmatrix} -0.54 & -0.78 & -42.46 & 9.29 & 0 \\ -4.28 & -26.51 & 12.81 & 0 & 0 \\ 0.4645 & -2.08 & -1.036 & 0 & 0 \\ 0 & 1 & -0.34 & 0.036 & 0 \\ 0 & 0 & 1.06 & -0.1115 & 0 \end{bmatrix} \begin{bmatrix} V \\ p \\ r \\ \phi \\ \psi \end{bmatrix} \quad (4.20)$$

$$B_{lat}u = \begin{bmatrix} -0.10 & 0 \\ -5.98 & 0 \\ -0.27 & 0.14 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \quad (4.21)$$

The characteristics of the GULMA UAV lateral-directional axis is as shown in Table 4.3.

Table 4.3: Characteristic values of lateral-directional modes.

Mode	Poles	Damping	Characteristics
Spiral mode	$\lambda_1 = -0.06$	1	Slow
Roll damping mode	$\lambda_2 = -26.20$	1	Fast
Dutch-roll mode	$\lambda_{3,4} = -0.89 \pm 6.01$	0.15	Oscillatory

The roll and dutch-roll modes are of natural frequencies, 26.2 and 26.07 Hz while the spiral mode pole with frequency 0.016 Hz is generally close to the imaginary axis. Now a closer look at the eigenvectors of the lateral-axis plant shown in Table 4.4, we see that the spiral mode eigenvector indicates a negligible side-slip and roll rate, but significant yawing motion with zero roll-angle. Thus a stable spiral mode is observed that is virtually a coordinated turn ($\beta = 0$) with a slowly varying roll angle that translates into a spiral path. The eigenvectors of the dutch-roll mode indicate oscillations in roll and yaw. A

Table 4.4: GULMA UAV lateral plant eigenvectors.

States	Spiral	Roll	Dutch-roll
β	-0.0786	-0.1695	0.9766
p	-0.0023	-0.9818	$-0.1633 - 0.0276i$
r	-0.0310	-0.0778	$0.0126 - 0.1318i$
ϕ	-0.1452	0.0365	$0.0063 + 0.0268i$
ψ	0.9858	0.0033	$-0.0232 + 0.0014i$

further inspection reveals that that the combined magnitudes of the roll and yaw rate are much larger than that of the side-slip, which indicates a dominant roll and yaw oscillation motion compared to the side-slip motion. Therefore for the GULMA UAV we expect to experience a snaking dutch-roll mode due to this roll-yaw oscillation.

We have shown that the poles of dutch-roll mode for the model is lightly damped, thus to guarantee improved flight performance we move from the lateral-directional state-space representation to a reduced model where we apply state LQR feedback method assuming all states are readily available. Now considering our objective which is to find an optimal control input $u(t), t \in [0, \infty]$ to track roll commands of the lateral model,

$$\begin{aligned}\dot{x} &= A_{lat}x + B_{lat}u \\ x &\in \mathfrak{R}^5, u \in \mathfrak{R}^2\end{aligned}$$

that minimizes the quadratic criterion J ,

$$J = \int_0^{\infty} (x(t)' Q x(t) + u(t)' R u(t) + 2x(t)' dt) \quad (4.22)$$

applying square completion to Equation 4.22, the criterion is reduced to

$$J = J_0 + \int_0^{\infty} (u(t) - u_0(t))' R (u(t) - u_0(t)) dt \quad (4.23)$$

Hence finding the feedback invariant, J_0 , is relatively simple as follows:

Lemma 1 (Feedback Invariant). Let P be a symmetric matrix. \forall control input $u(t)$, $t \in [0, \infty]$ we have that

$$\begin{aligned} & \int_0^{\infty} (x'(A'P + PA)x + 2x'PBu(t)) dt \\ &= \int_0^{\infty} (x'(A' + u'B)Px + x'(Ax + Bu)) dt \\ &= \int_0^{\infty} (\dot{x}'Px + x'P\dot{x}) dt \\ &= \frac{\int_0^{\infty} (d(x'Px))}{dt} = \lim_{x \rightarrow \infty} x(t)'Px(t) - x(0)'Px(0) \end{aligned}$$

Hence,

$$\int_0^{\infty} (x'(A'P + PA)x + 2x'PBu(t)) dt = x(0)'Px(0) \quad (4.24)$$

Now assume there exist a symmetric solution to P , i.e. $P^T = P$, i.e. P must be a square matrix to the algebraic equation

$$A'P + PA + Q - PBR^{-1} + BP = 0 \quad (4.25)$$

such that $A - BR^{-1}(BP)$ is Hurwitz then the feedback law becomes

$$K = -R^{-1}(BP) \quad (4.26)$$

So now the reduced model is defined by,

$$\begin{cases} \tilde{x}^T = \begin{bmatrix} p & r & \phi \end{bmatrix}; \tilde{u}^T = \begin{bmatrix} \delta_a & \delta_r \end{bmatrix} \\ \dot{\tilde{x}} = A_R x + B_R u \\ y = C_R x \end{cases} \quad (4.27)$$

The property of *Lemma1* below is used as:

- a A stabilizing solution to the LQR design problem where a finite performance index (PI) exists and is unique;
- b The reduced lateral model pair (A_R, B_R) is stabilizable and pair (A_R, C_R) is detectable, where $Q = C_R^T C_R$

Given the system in Equation 4.27, the optimal LQR controller is obtained by choosing the control input (\tilde{u}) that minimizes the quadratic cost or PI is

$$\int_0^\infty (\tilde{x}^T Q \tilde{x} + \tilde{u}^T R \tilde{u}) dt \quad (4.28)$$

where $Q > 0, R \geq 0$ (R must be invertible).

We see here the aim of the LQR is to steer the error values of the initial states to zero; in order to maintain a stable system [76]. The control input (\tilde{u}) uses the feedback gain K_{43} such that, in closed loop, the performance index is minimized for the feedback control law:

$$u_{optimal} = -K_{43} \tilde{x}$$

where

$$K_{43} = R^{-1} B^T P$$

and P is the solution of the algebraic quadratic Riccati equation described as:

$$A^T P + PA + Q - PBR^{-1}B^T P = 0$$

with the optimal cost $J(u_{optimal}) = \tilde{x}^T(0)P\tilde{x}(0)$.

$$\int_0^\infty (x^T Q x + u^T R u) dt \quad (4.29)$$

where, $Q > 0, R \geq 0$ The Q and R are weighing matrices for controls or states energy. The PI is minimized for the feedback control law:

$$u_{optimal} = -K_{43}x$$

Where, K_{43} , is the optimal gain at $V = 43\text{m/s}$.

In order to have optimal regulation and tracking with zero steady-state error, the approach in [80], reveals that an integral action should be added which creates an extended system of the model. In accordance to this technique, the error state is added between the ϕ and its reference, ϕ_{ref} , and then integrated to steer the error to zero. The resulting new state, \tilde{x}_2 , is now made up of the former state, \tilde{x}_1 , and the integral of the roll angle error such that the extended lateral state space model is given as:

$$\begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} = \begin{bmatrix} A & 0 \\ -E & 0 \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} \tilde{u} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \dot{\phi}_{ref}$$

where

$$\tilde{x} = \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix}$$

with $\tilde{x}_1 \in \mathfrak{R}^3$ and $\tilde{x}_2 \in \mathfrak{R}^2$

A suitable tuning of the Q & R weighting matrices with consideration for fast

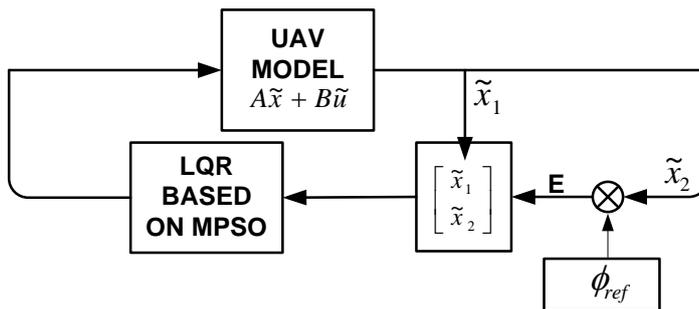


Figure 4.7: LQI block diagram.

response to ϕ and good damping of roll-yaw coupling will therefore guarantee that ϕ will track ϕ_{ref} . To achieve this, the best approach to selection of matrices is needed. Applying Bryson's method as previously used for the longitudinal plant would result in tedious and time consuming work to effectively optimize the PI [50]. The application of this PSO algorithm provides faster tuning time to obtain the best fit matrices with respect to output performance objectives. The block diagram of the linear plant with LQR feedback and integral action is shown in Figure 4.7.

4.2 Multi-objective Optimization

4.2.1 Multi-objective PSO

Generally as earlier discussed, PSO methods present a stochastic calculation based on the movement of swarms of particles tending towards the most weighted particle in the swarms. According to [33], all solutions in PSO can be represented as particles in a swarm where each particle has a position and velocity vector and the position co-ordinate represents a parameter value. Similar to most optimization techniques, PSO requires a fitness evaluation function relevant to the particle's position [87]. Computational search for the personal best and global best positions of the i_{th} particle are given as X_{pb} and X_{gb} respectively. Each particle is initialized with a random position and velocity. The velocity

of each particle is accelerated toward the X_{gb} and its own X_{pb} based on the following equation [36].

$$V_{update} = w \times V_i + C_1 \times Rand(1) \times (X_{pb} - X_i) + C_2 \times Rand(1) \times (X_{gb} - X_i) \quad (4.30)$$

where

$$w = w_{max} - (w_{max} - w_{min}) \times 0.33$$

$$w_{max} = Range(Q_{max}R_{max}) \times 0.5$$

$$w_{min} = Range(Q_{min}R_{min}) \times 0.5$$

$$Rand = random[0, 1]$$

C_1 and C_2 are acceleration constants and w is a weighting factor that guarantees convergence to X_{gb} . According to [34], it is possible to select suitable w to balance between the X_{pb} and X_{gb} search space.

$$X_{pb} = \begin{cases} X_{pb} & \text{if Fit(QR)} \geq X_{pb} \\ QR & \text{if Fit(QR)} \leq X_{pb} \end{cases} \quad (4.31)$$

$$X_{gb} = \min\{X_{pb}\}$$

LQR synthesis requires iterating on Q and R which can sometimes be a time consuming task as previously discussed for the longitudinal control design. The choice of these weighting matrices is described in the Bryson's rule [36]. This trial and error method describes how the weighting matrices may be selected using the maximum desired deviations of the states x and the control inputs u . Here it described an algorithm to obtain the best solution to an initial popula-

tion of Q and R matrices. The initial population is constrained and compared with the with a population size over a few iterations. For the tracking problem the performance output is weighed with the performance objectives. The performance objectives is for ϕ closed loop output response characteristics: percentage overshoot $< 10\%$, rise and settling times of < 2 and 5 seconds respectively. The flow chart for the multi-objective PSO process is described in Figure 4.8.

4.2.2 Fitness Consideration

A multi-objective based fitness function for the selection of weighting matrices Q and R are needed to perform the PSO process. A similar multi-objective optimization model that described this process is in [78]. This process used the error of amplitude from a stable state, magnitude of a control value and pole position of the closed-loop system to formulate the multi-objective function which gave pareto-optimal solutions without shape of the pareto-front end needed to obtain the best optimal solution. However, here we use a fitness function based on the closed-loop feedback characteristics of overshoot (OS), rise time (tr) and settling time (ts) of the output response and a performance function based on Equation 4.28. The proposed multi-objective function is formulated as:

$$Fit(QR) = tr * tr_{scale} + OS * OS_{scale} + ts * ts_{scale} \quad (4.32)$$

$$MultiPSO = \begin{cases} Fit(QR) \\ J(QR) \end{cases} \quad (4.33)$$

where OS_{scale} is weighting factor for OS, tr_{scale} is tr weighting factor and ts_{scale} is ts weighting factor.

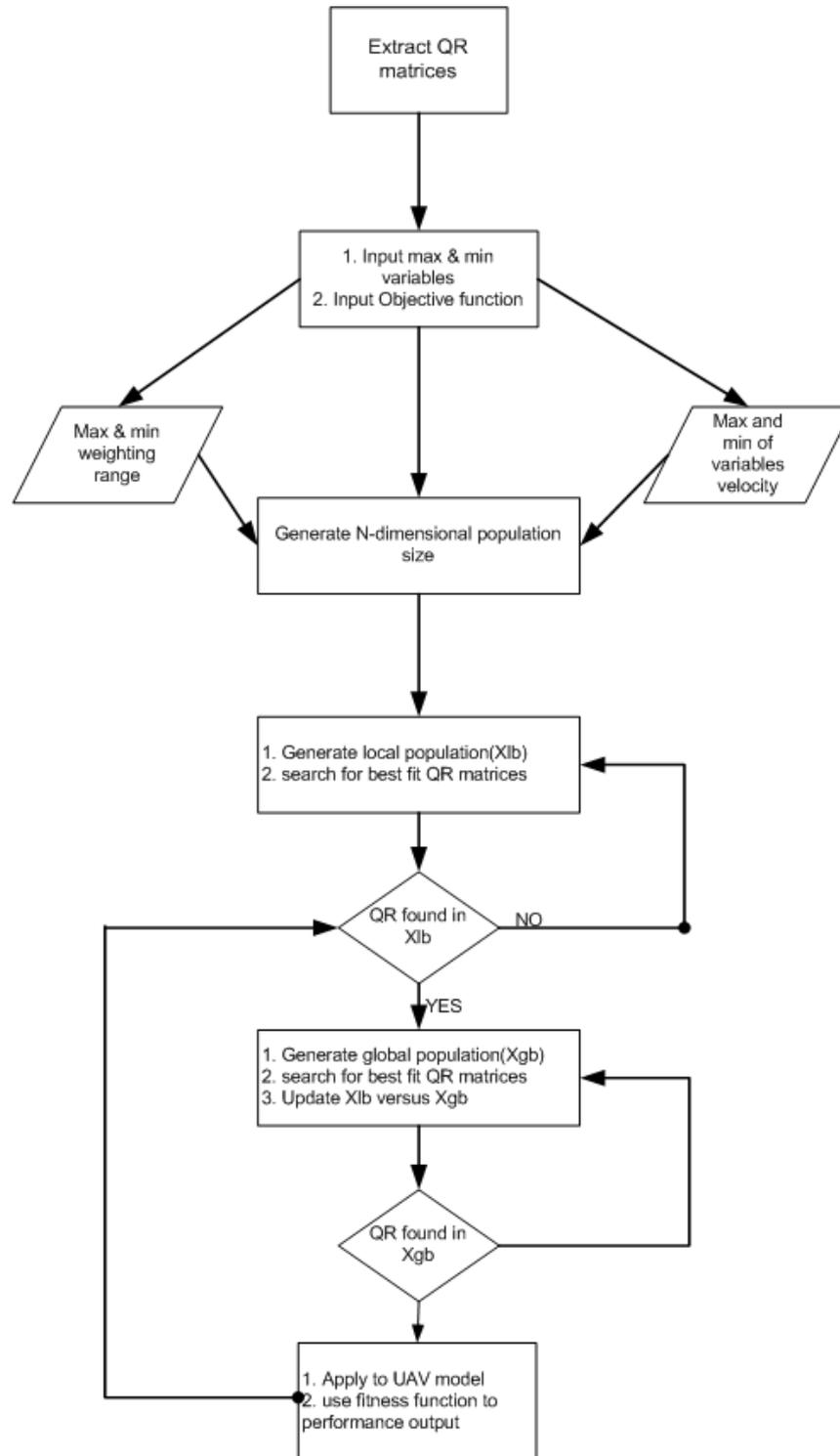


Figure 4.8: Multi-objective PSO algorithm flow chart.

4.2.3 Simulation and Analysis

The aircraft was considered to be at steady flight and trimmed lateral-directional dynamic motion. The design method was first to select the QR matrices then

apply PSO algorithm. The next step was to implement the results of the control gains on the model which are based on the robust PSO algorithm and multi-objective function.

Using the GULMA UAV linear lateral model it was possible to implement the multi objective PSO algorithm in optimal tuning of QR matrices to track the aircraft roll command ϕ_{ref} . The maximum and minimum actuator (aileron and rudder) saturation points are 25.58° to -24.42° and 24.81° to -25.18° respectively and were used to obtain the initial iteration point for the input matrix R . Subsequently for the Q matrix the maximum values for the states using Bryson's rule, an initial population matrix was developed using in the algorithm. Figure 4.9 shows the response to a 2-step doublet input for $< 5secs$. The response based on the objective parameters described in the previous section indicates a well-damped response with a $tr < 2secs$ and $OS < 10$. The actuator response in Figure 4.10 is well within bounds.

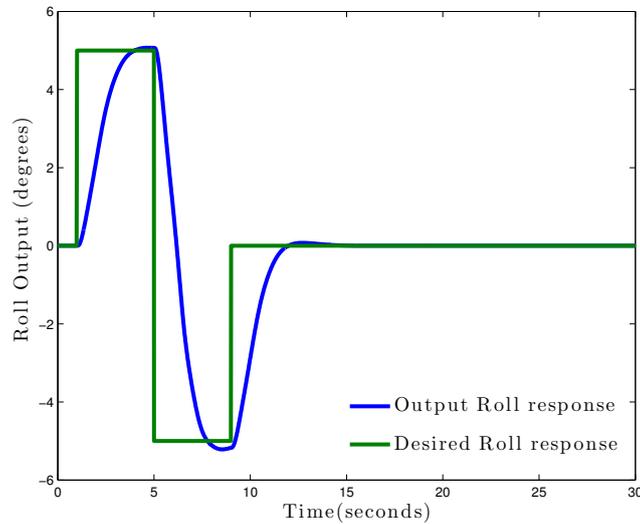


Figure 4.9: Doublet response based on multi-objective PSO results for Q_1R_1 .

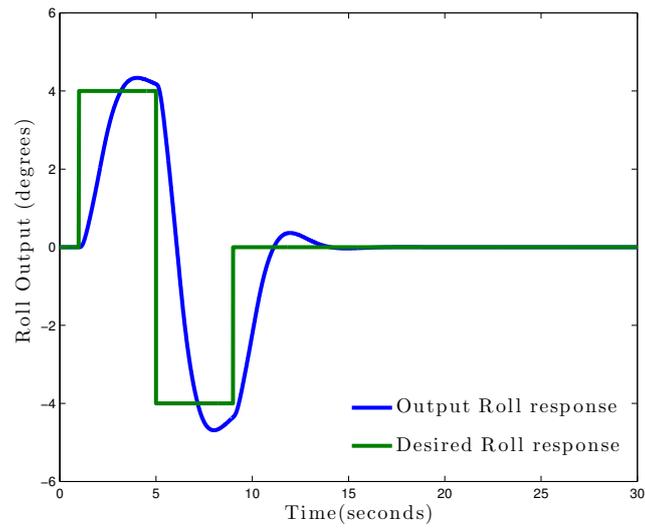


Figure 4.10: Doublet response based on standard PSO results for Q_2R_2 .

$$Q_1 = \begin{pmatrix} 0.0017 & 0 & 0 & 0 \\ 0 & 0.3014 & 0 & 0 \\ 0 & 0 & 0.0810 & 0 \\ 0 & 0 & 0 & 0.2515 \end{pmatrix}$$

$$R_1 = \begin{pmatrix} 0.0022 & 0 \\ 0 & 0.0003 \end{pmatrix}$$

$$Q_2 = \begin{pmatrix} 0.1707 & 0 & 0 & 0 \\ 0 & 0.2771 & 0 & 0 \\ 0 & 0 & 0.3067 & 0 \\ 0 & 0 & 0 & 0.2989 \end{pmatrix}$$

$$R_2 = \begin{pmatrix} 0.0017 & 0 \\ 0 & 0.0003 \end{pmatrix}$$

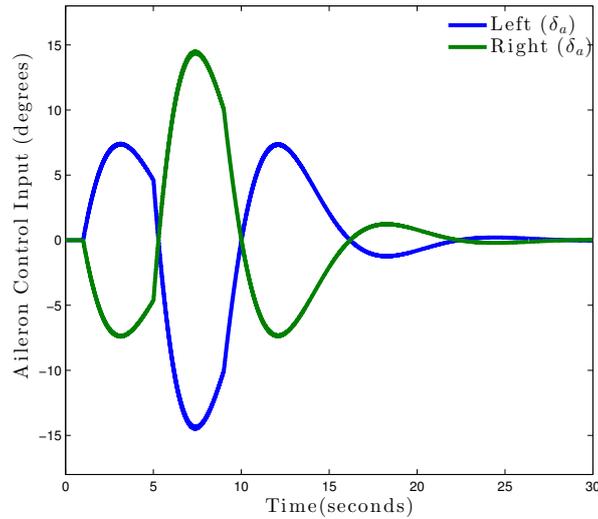


Figure 4.11: Aileron response with state-feedback based on multi-objective PSO results for Q_1 and R_1 .

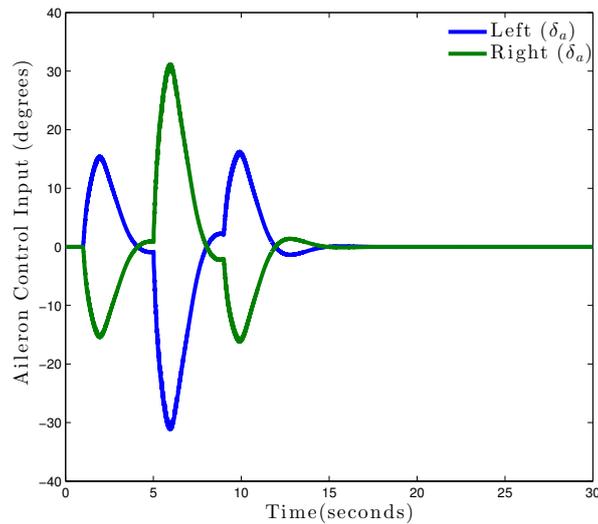


Figure 4.12: Aileron response with state-feedback based on standard PSO results Q_2 and R_2 .

4.3 Robust Flight Controller Synthesis

We shall now re-design the lateral axis controller using the robust controller synthesis techniques. This enables us to assess the lateral model of the GULMA UAV with regards to handling quality specification for the roll-yaw axis. The model-based methods we shall consider for robust controller design and synthesis are the H_∞ and μ approach as described in [32]. These methods allows for

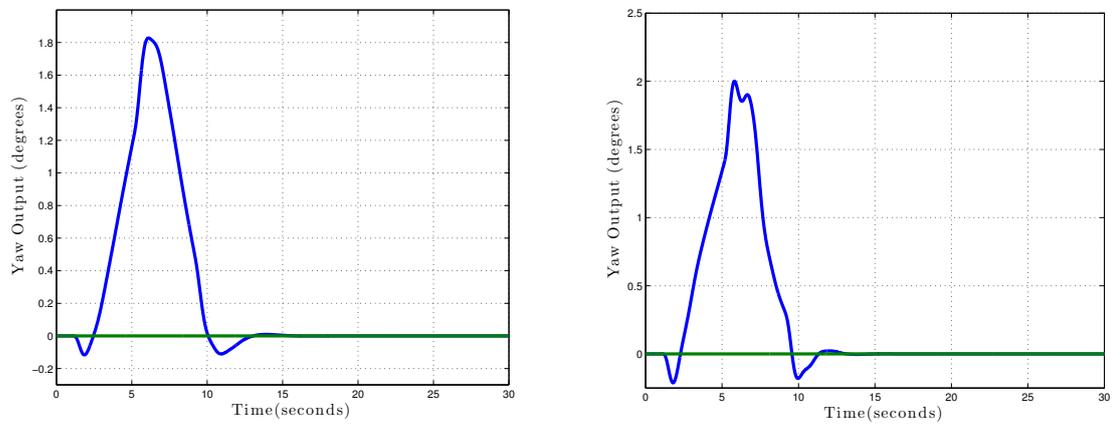


Figure 4.13: Yaw angle response due to doublet command.

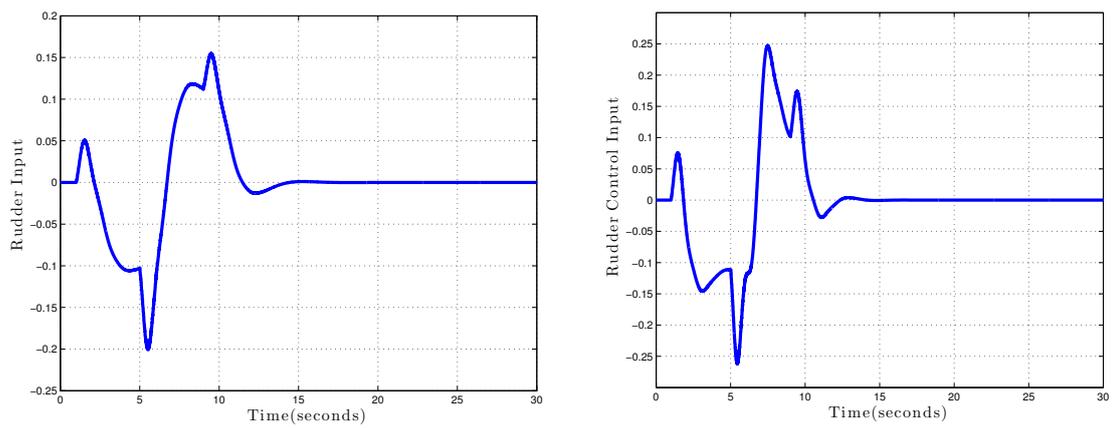


Figure 4.14: Rudder response due to doublet command.

design objectives to be optimally guaranteed for different performance objectives through the usage of weighting functions and if necessary filters.

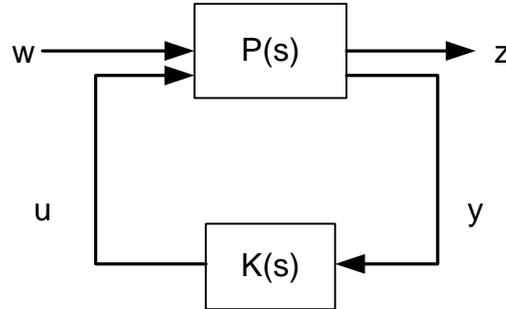


Figure 4.15: Standard Lower LFT block diagram.

4.3.1 Robust Design Objectives

The performance objectives is to track reference command signals generated by the autopilot to the lateral-directional axis. The goal is to design the decoupled transient responses of the inputs from the aileron and rudder to the side slip angle. This enables us to test for better handling quality of the UAV. Therefore we shall use the 2-degrees-of-freedom (2DOF) control scheme described in [32] [48] to design and synthesize the H_∞ and μ stabilizing controllers. We adopt standard configuration for robust closed-loop system as shown in Figure 4.15. We shall consider the following robust and performance objectives for control design and synthesis:

- Tracking roll command signals with $< 2^\circ$ tracking error, a bandwidth of 2 rad/s, settling time of < 2 seconds and rise time of 1.1 seconds.
- Tracking side slip command signals with $< 1^\circ$ tracking error, a bandwidth of 2 rad/s, settling time of < 5 seconds and rise time of < 3 seconds.
- Control effort is within aileron actuator angle and rate limits of 30° and

50° per second respectively. The rudder actuator angle and rate limits are 30° and 60° per second respectively.

- At least one of the controllers should meet the robust and performance objectives for all values of the plant model uncertainties.

The desired response (command signal) for the lateral input to roll rate is to match the first-order response shown in Figure 4.16 while that for the yaw command signal is as shown in Figure 4.17. These models represent our tracking performance specifications.

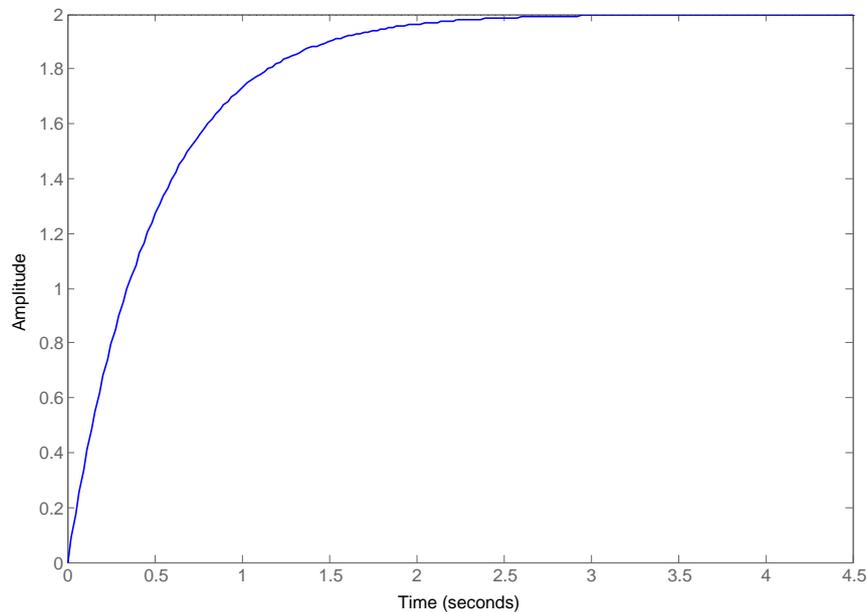


Figure 4.16: Roll command performance objective.

Now based on the performance objectives we shall select the weighting functions to shape the loop for the input and output error signals.

4.3.2 Selection of Weighting Functions

The performance weighting functions, W_{perf1} and W_{perf2} , are used to shape the response from the ailerons and rudder to side-slip angle to match the desired models. We aim here to minimize the difference in peak gains between the command and actual signals in order to achieve the desired closed-loop performance

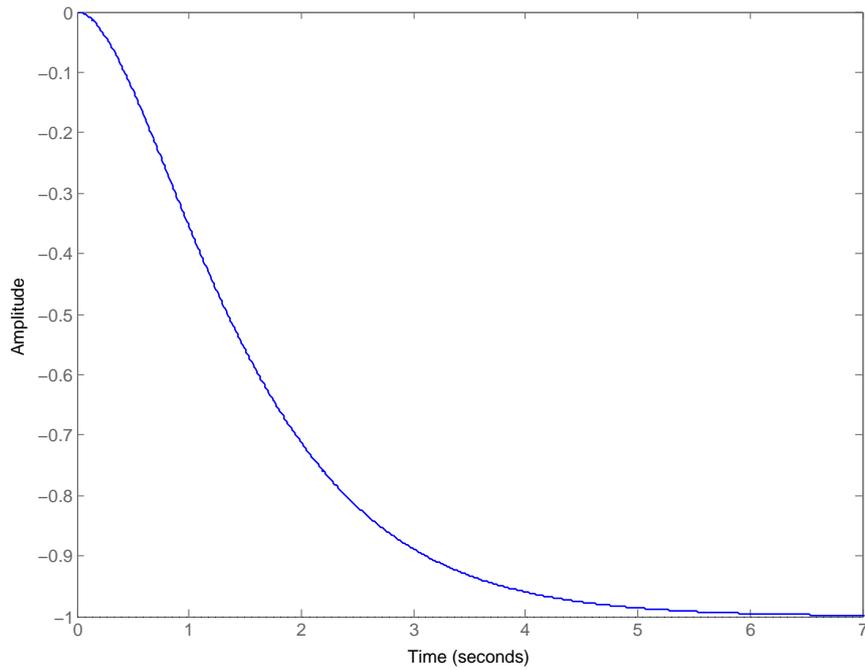


Figure 4.17: Yaw command performance objective.

specifications. The minimum frequency of the model is 0.07 rad/s for the zero in the right-half plane. Hence for frequencies below this value are not useful for tracking purposes, so we model the weight of the roll angle using a passband filter for frequency range of 0.07 to 30 rad/s. The limits on the actuator deflection magnitude and rate are weighted as W_{ac} , that corresponds to the aileron and rudder deflection angle and rate limits as indicated in performance our objectives. The following weights specifications were obtained through tuning and iteration processes of the full closed loop system:

$$W_{p1} = \frac{0.05s^4 + 2.9s^3 + 105.93s^2 + 6.17s + 0.16}{s^4 + 9.19s^3 + 30.80s^2 + 18.83s + 3.95}$$

$$W_{p2} = \frac{0.075s^4 + 4.35s^3 + 158.9s^2 + 13.75s + 0.24}{s^4 + 9.19s^3 + 30.80s^2 + 18.83s + 3.95}$$

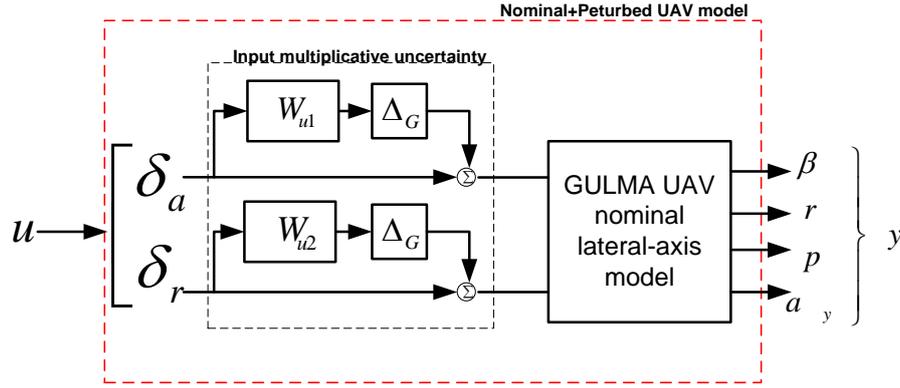


Figure 4.18: Input multiplicative uncertainty model.

$$W_{ac} = \begin{pmatrix} 1/50 & 0 & 0 & 0 \\ 0 & 1/30 & 0 & 0 \\ 0 & 0 & 1/60 & 0 \\ 0 & 0 & 0 & 1/30 \end{pmatrix}$$

We consider the nominal model of the lateral plant that only approximates the true behaviour of the UAV. Therefore, to account for model uncertainties we introduce the multiplicative uncertainty, $(W_u \Delta_G)$, at the plant input where error dynamics Δ_G have gain < 1 across all frequencies and W_u is a high pass filter to capture the degree of accuracy for the two models shown in Figure 4.18. Hence we have as follows:

$$W_u = \begin{pmatrix} W_{u1} & 0 \\ 0 & W_{u2} \end{pmatrix}$$

where, W_{u1} and W_{u2} are the error dynamics for the aileron to side slip and rudder to side slip respectively which reflects the frequency ranges in which the model is more or less accurate. There are typically more modelling errors at high frequencies given as:

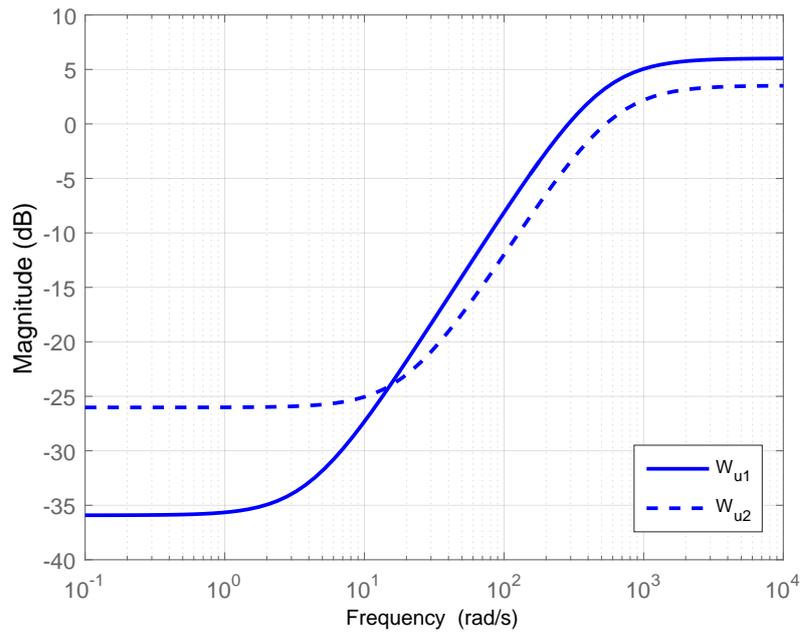


Figure 4.19: Error dynamics of nominal plant in terms of frequency.

$$W_{u1} = \frac{2s + 4}{s + 90}$$

$$W_{u2} = \frac{0.4s + 2}{s + 600}$$

4.3.3 Closed Loop System with 2-DOF H_∞ Synthesis

In this 2-DOF scheme we implement a feedback controller to achieve our performance objective for reference tracking and maintaining internal and robust stability of the closed loop system as well as to minimize the output of the overall system and that of the reference model described in subsection 4.3.1. Figure 4.20 shows the structure of the 2-DOF approach. With regards to the standard configuration, we rearrange the exogenous inputs and outputs so that,

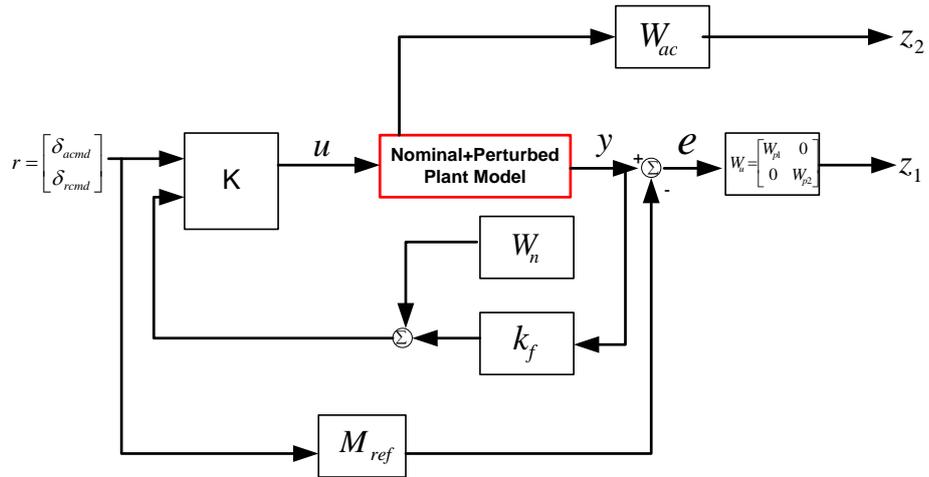


Figure 4.20: 2-DOF Closed loop for the lateral model.

$$w = r = \begin{pmatrix} \delta_{acmd} \\ \delta_{rcmd} \end{pmatrix}$$

$$z = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} W_p \\ W_{ac} \end{pmatrix}$$

$$y = \begin{pmatrix} r \\ y \end{pmatrix}$$

Hence the interconnected system is

$$P = \begin{pmatrix} -W_p M_{ref} & W_p G \\ 0 & W_{ac} \\ I & 0 \\ 0 & G \end{pmatrix}$$

where P is set as

$$P_{11} = \begin{pmatrix} -W_p M_{ref} \\ 0 \end{pmatrix}$$

$$P_{12} = \begin{pmatrix} W_p G \\ W_{ac} \end{pmatrix}$$

$$P_{21} = \begin{pmatrix} I \\ 0 \end{pmatrix}$$

$$P_{22} = \begin{pmatrix} 0 \\ G \end{pmatrix}$$

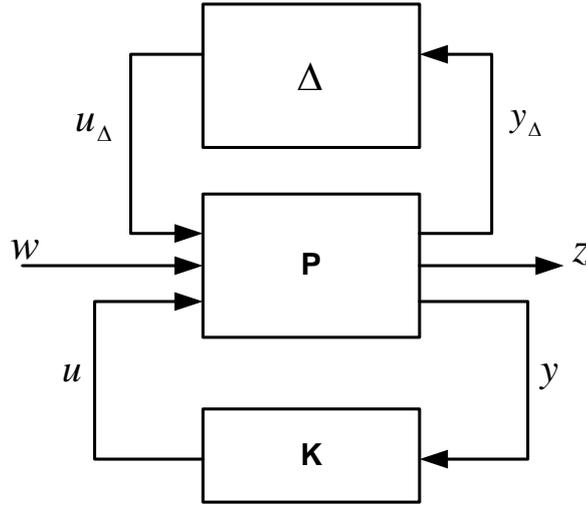
The state space description for the plant nominal model, G of the interconnected system is given as

$$\dot{x} = Ax + B_1 w + B_2 u$$

$$z = C_1 x + D_{11} w + D_{12} u$$

$$y = C_2 x + D_{21} w + D_{22} u$$

where $x \in \mathfrak{R}^4$ is state vector, $w \in \mathfrak{R}^2$ exogenous input vector, $u \in \mathfrak{R}^2$ is control input vector, $z \in \mathfrak{R}^2$ is the error dynamics of the output vector and $y \in \mathfrak{R}^2$ is the measurement vector.

Figure 4.21: μ -Synthesis block diagram.

4.3.4 Closed Loop System with 2-DOF μ Synthesis

Here we aim to perform a μ synthesis that determines the stabilizing controller, K , for also for a stable closed loop of the nominal system model $\forall \Delta$, $\max_{\omega} \bar{\sigma}[\Delta(j\omega)] \leq 1$ and satisfies

$$\|F_U[F_L(P, K), \Delta]\|_{\infty} < 1$$

where $F_L(P, K)$ is the Linear Fractional Transformation of the interconnected plant P . The performance is tested with respect to the the addition of an input multiplicative uncertain structure.

$$\Delta P = \left\{ \begin{pmatrix} \Delta & 0 \\ 0 & \Delta_s \end{pmatrix} : \Delta \in C^{8 \times 2}, \Delta_s \in C^{2 \times 2} \right\}$$

Δ_s is an unstructured uncertainty. For robust performance with controller K we have

$$\mu_{\Delta p}(F_L(P, K)(j\omega)) < 1$$

Therefore we aim to minimize the peak value of the structured singular value, $\mu_{\Delta p}(\cdot)$ of the closed loop system matrix $F_L(P, K)$ over the set of all stabilizing controllers K .

4.3.5 Simulation and Analysis of Controller Synthesis

We shall simulate the design for the robust controllers for the lateral-axis of GULMA UAV for a climbing turn flight condition. The linearized plant model was obtained at a velocity of 35 m/s and height of 100 m. The linearized state space model is given as

$$A_{lat_{35}}x = \begin{bmatrix} -0.2985 & -34.72 & -0.1649 & 8.04 \\ 0.4222 & -0.8421 & -1.675 & 0 \\ -0.8302 & 10.45 & -21.6 & 0 \\ 0 & -0.1286 & 1 & 0.02596 \end{bmatrix} \begin{bmatrix} v \\ r \\ p \\ \phi \end{bmatrix} \quad (4.34)$$

$$B_{lat_{35}}u = \begin{bmatrix} 0.0011 & 3.907 \\ 10.33 & -0.3194 \\ -100.2 & -0.5659 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \quad (4.35)$$

$$y_{lat35}x = \begin{bmatrix} 0.02856 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -0.2965 & 0.293 & -0.0499 & 0 \end{bmatrix} \begin{bmatrix} \beta \\ r \\ p \\ ay \end{bmatrix}$$

$$D_{lat35}u = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0.0181 & 3.9 \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix}$$

where ay is the lateral acceleration of the UAV in m/s^2 . We introduce a multiplicative uncertainty model, $W_u\Delta_G$ at the input of the plant model to account for model uncertainties. The frequency description of the relative error dynamics is shown in Figure 4.19. The controllers are designed using the Matlab subroutines **hinfsyn** and **dknsyn** to perform H_∞ and $\mu/D - K$ iterations synthesis respectively as described in [56].

4.3.5.1 Comparison of Controllers in the Frequency Domain

Comparing the frequency response results obtained for both controllers as shown in Figure 4.22, we note that the H_∞ controller initially works within performance limits (max peak gain of $1dB$) for the nominal model while at a peak gain close to $6dB$ and frequency of 0.6 rad, the closed loop performance becomes unstable for some perturbed model within pre-set uncertainty modelling error bounds. We also observe that for the nominal model the μ controller has a better performance when compared to that of the H_∞ controller, as observed the controller minimizes the peak gain between to 0.79 and 0.83 at 0.6 rad for the overall system performance including errors associated with the uncertain model variation.

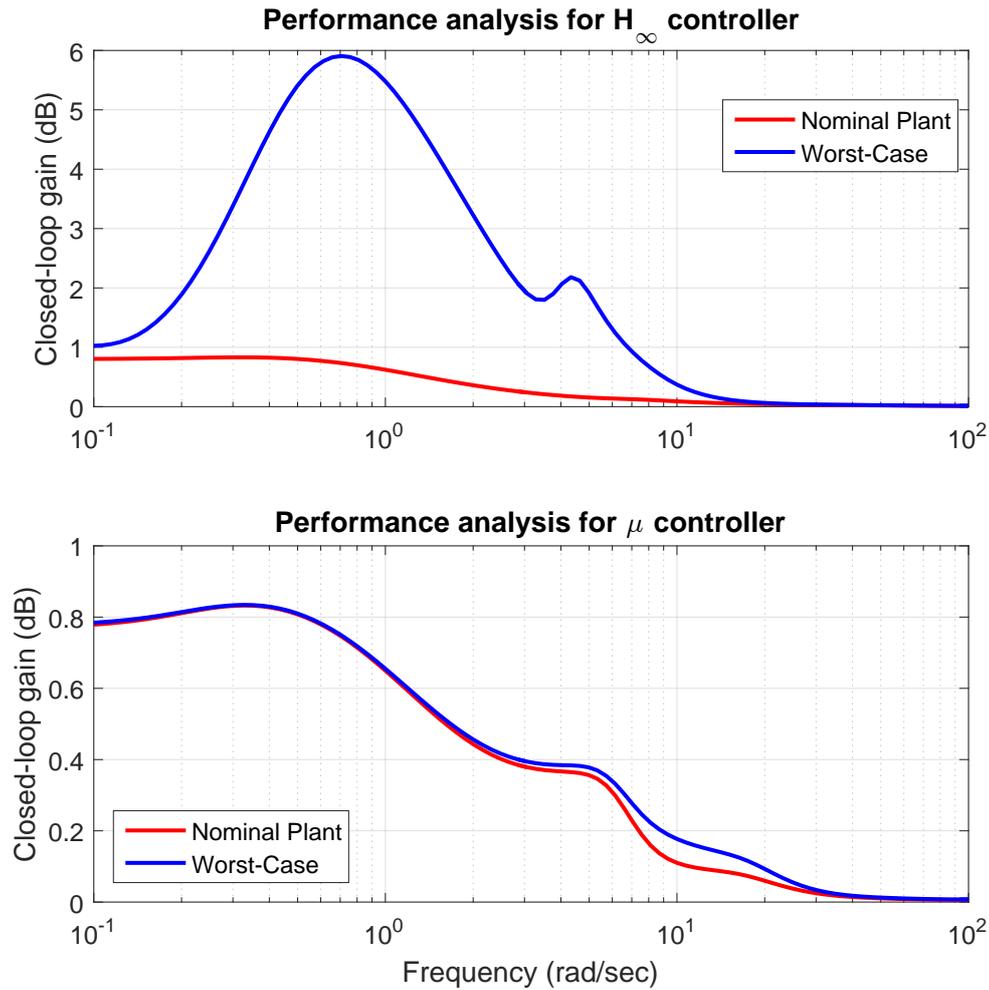
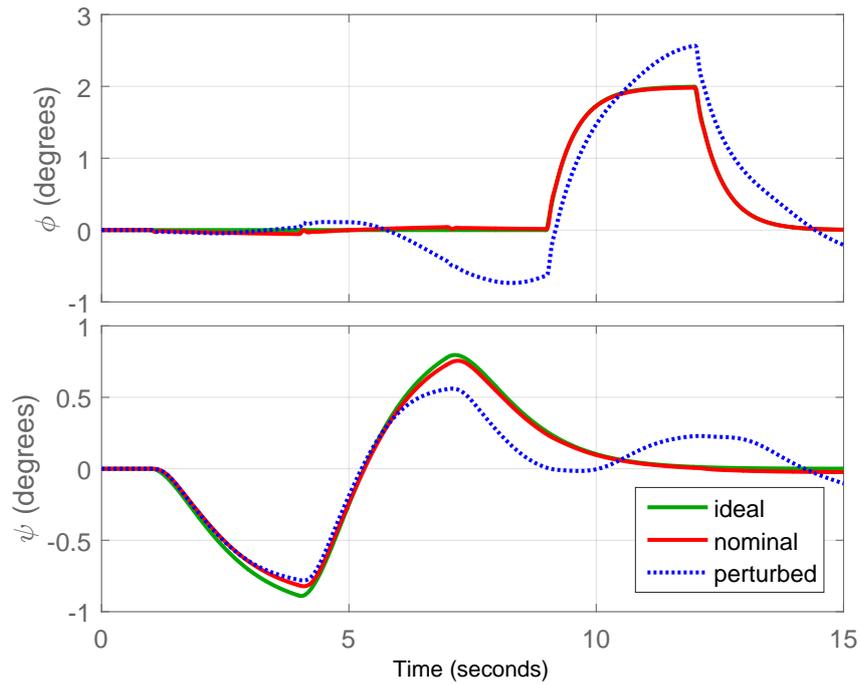
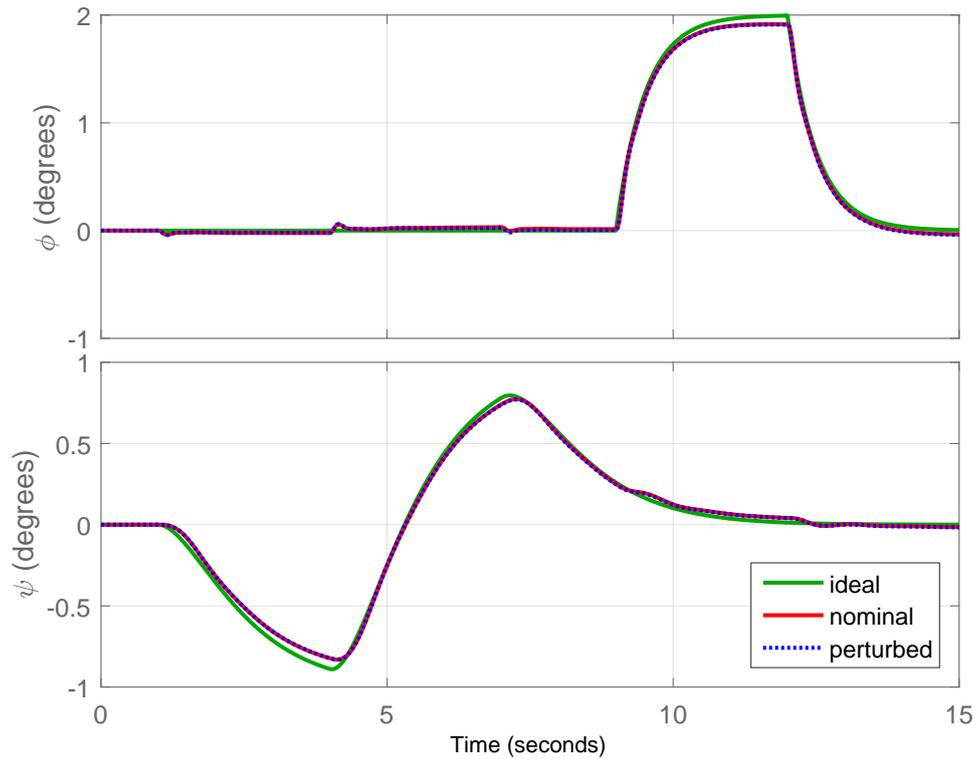


Figure 4.22: Closed-loop worst-gain response of controllers.

4.3.5.2 Comparison of Controllers in the Time Domain

We use the time domain to test and compare the robustness of the designed controllers as described in [32]. The time domain responses of the nominal and perturbed models to the reference tracking commands is shown in Figure 4.24. We observe as expected the μ controller has better tracking performance in comparison to that of the H_∞ controller.

The closed-loop response for the μ controller indicates a nearly identical nominal and perturbed model response. We see that the roll response of the air vehicle closely tracks the ideal roll-rate command input initially and then slightly departs from this command. This is due to a right-half plane zeros of

Figure 4.23: Time domain responses H_∞ controller.Figure 4.24: Time domain responses μ controller.

model at 0.07 rad/sec. Comparing this result with the H_∞ controller we see that the although it meets performance specifications for the nominal plant model, performance and robustness deteriorates for part of the perturbed model.

4.3.6 Robust Stability and Performance Analysis

Robust stability (RS) and performance (RP) analysis is done in order to assess margin for which the controller can work effectively over the specified flight condition. We do this by implementing the functions in the robust control toolbox for MATLAB. The use of the the functions, **robuststab** and **robustperf** enables the analysis of the robust stability and performance of the model. The results in Table 4.5 are from repetitive tuning of the weighting specifications using the these functions.

Table 4.5: Robust stability and performance results.

Margin	μ upper	μ lower	Remark
μ controller 46 states	4.64 1.19	4.64 1.19	RS achieved by 464% RP achieved by 120%
H_∞ controller 26 states	54.84 0.40	54.84 0.40	RS achieved by 5480% RP not achieved by 40.3%

We see that with H_∞ controller the results indicate the closed-loop system is robustly stable and performance robustness is not achieved as the lower bound is < 1 as described in [7]. However for the μ controller we have achieved robust stability that satisfies the lower and upper bounds > 1 that is stable to tolerate up to 464% of plant uncertainty at 21.5 rad/s which at this frequency the destabilizing poles are at $\pm 0.129i$. The μ controller also achieved robust performance To further illustrate the robustness of the closed-loop system with μ controller we shall test the system by increasing and decreasing the input disturbance of 50% for a time domain response shown in Figure 4.25 and 4.26 respectively.

We observe in these responses that the performance specification for the control system is maintained due to these changes, hence the controller is robustly stable.

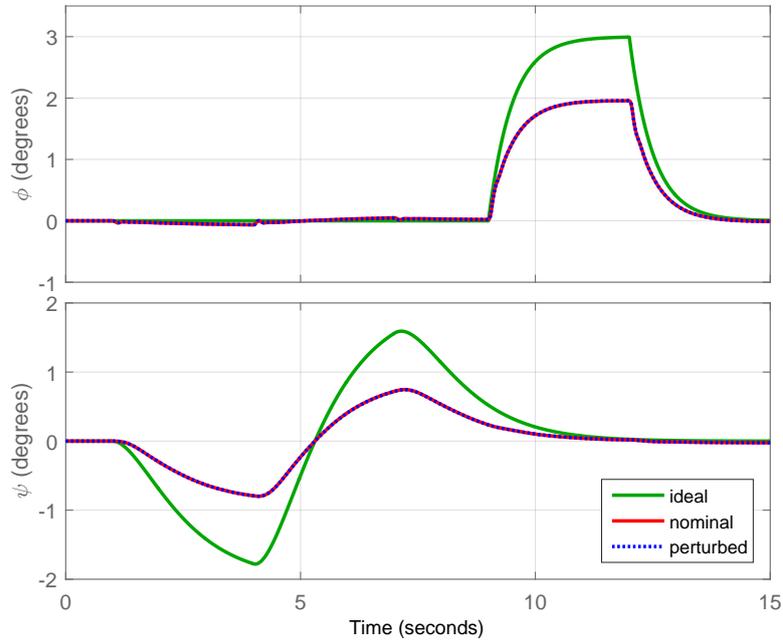


Figure 4.25: μ closed-loop time responses due to 50% increase in command signal.

4.4 Summary

The aircraft model used was considered at steady flight condition with lateral-directional dynamic values computed. The design method was first to select the Q and R matrices. The next step was to implement the results of the control gains on the model which are based on the multi-objective PSO algorithm.

Accordingly we implemented a state feedback controller based on the optimal weighting matrices obtained explicitly from the multi-objective PSO algorithm and compared to a conventional PSO algorithm. An initial population for Q and R was computed using the Bryson's rule. Subsequently for the Q matrix the maximum values for the states using Bryson's rule, an initial population matrix was developed using the algorithm. Figure 4.9 shows the response to a

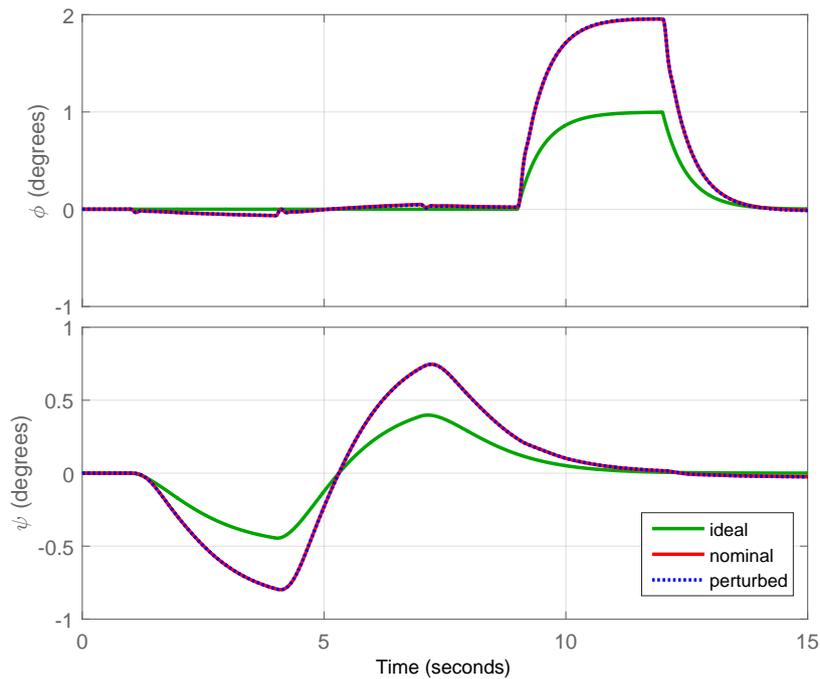


Figure 4.26: μ closed-loop time responses due to 50% decrease in command signal.

doublet input to roll angle of $< 5secs$. The response is well damped with a rise time $< 2secs$ and an overshoot $< 10\%$. The actuator response corresponding to the doublet is given in Figure 4.10 and it can be seen that it stays within the aileron permissible limits.

The corresponding results for the yaw output response and rudder actuator is also visualized in the Figures 4.13 and 4.14 respectively. These results indicate that while the cost function minimizes the control action, a selection of the Q and R matrices using the multi-objective PSO algorithm described guarantees that the yaw output response in Figure 4.13(a) is $< 1.8^\circ$ due to the control input of $< 0.2^\circ$ in Figure 4.14(a). This is an improvement when compared to output response using the conventional PSO algorithm in Figure 4.13(a) and its corresponding control input $> 0.2^\circ$ in Figure 4.14(b).

We also demonstrate the robust stability and performance of the UAV lateral model by designing a model-based controller synthesis. A test for the robustness of the designed controllers (μ and H_∞), where the closed-loop response for the

μ controller proved to be more robust (stability and performance) to modelling errors than with the H_∞ controller. These results suggest that during a climbing turn flight condition we achieved robust stability for the roll-yaw motion of the air vehicle with stability margin value of 4.64 satisfying the lower and upper bounds requirement of > 1 and a tolerance value, 464%, of plant uncertainty. The robust performance was achieved with margin value of 1.19 which also satisfies the bound requirement.

Chapter 5

Guidance System Design

5.1 UAV Guidance System

Autonomous unmanned aerial systems (UAS) share the basic need for sensors or sensor-systems which provide an estimate of the vehicle's full state vector. The state vector normally consists of three position co-ordinates, components of the velocity vector and anywhere between three and nine parameters which describe the vehicle's attitude. UAV operating autonomously need some form of control and guidance, in addition to these sensor systems that allow it to manoeuvre in a way consistent with specific flight missions [42] [43]. There are numerous varieties of guidance, navigation and control (GNC) frameworks already published in the academic and industrial environment. In this Chapter we therefore implement a Lyapunov-based guidance vector field to provide instruction for operating the aircraft controls (aileron, elevators, thrust, etc) to follow a pre-determined path set by a user. This method was selected due to the ease of implementation in real-time. The test platform used for this chapter is the Ultrastick UAV simulation model and our focus here is limited to simulation and analysis purposes using the Matlab/Simulink codes developed in [5]. The software architecture adopted is shown in Figure 5.1.

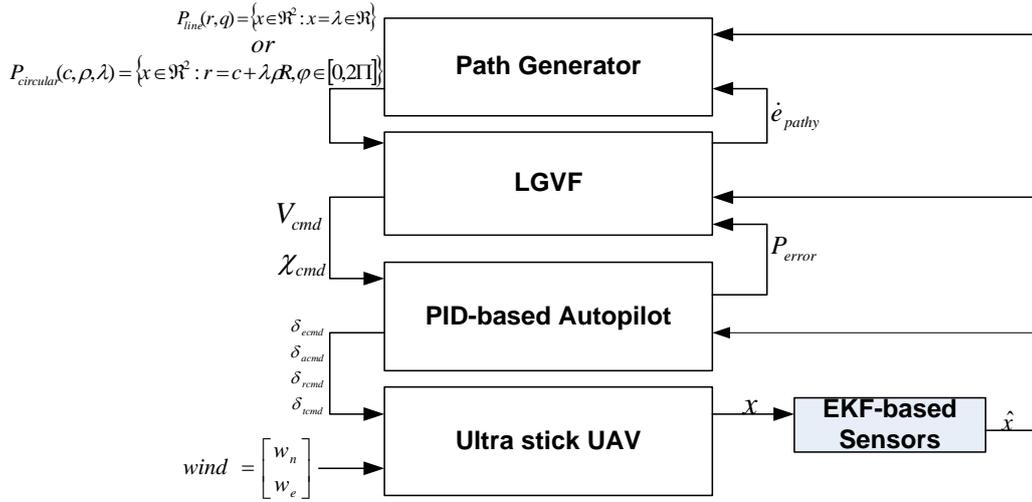


Figure 5.1: Full UAV system architecture.

5.2 Guidance System

Guidance as defined by [74] is the process for guiding the path of an object towards a given point, which in general may be moving. Furthermore, according to [6] various guidance models exist and they can be derived with regards to the level of fidelity or existing uncertainty of the aircraft model that are highly dependent on the choice of its application. An overview of these models is described and grouped in two categories: the kinematic and dynamic guidance models.

5.2.1 Kinematic Guidance Model

The kinematic guidance model was used to develop both longitudinal and lateral-directional guidance laws for generation of the UAV flight-path and course commands. The simplest laws for longitudinal and lateral-direction motions was adapted as described in [6]:

$$\dot{p}_n = V_a \cos \psi \cos \gamma_a + \omega_n \quad (5.1)$$

$$\dot{p}_e = V_a \sin \psi \cos \gamma_a + \omega_e \quad (5.2)$$

$$\dot{\psi} = \frac{g}{V_a} \tan \phi \quad (5.3)$$

$$\dot{V} = K_a(V_a^c - V_a) \quad (5.4)$$

$$\dot{\gamma} = K_\gamma(\gamma^c - \gamma) \quad (5.5)$$

$$\dot{\phi} = K_\phi(\phi^c - \phi) \quad (5.6)$$

$$\dot{h} = V_a \sin \gamma_a - \omega_d \quad (5.7)$$

5.2.2 Dynamic Guidance Model

The dynamic model essentially provides guidance laws in respect of the thrust and drag forces which generates trajectories for determining the positional vectors of the aircraft [6]. These laws were given as:

$$\dot{p}_n = V_g \cos \chi \cos \gamma \quad (5.8)$$

$$\dot{p}_e = V_g \sin \chi \cos \gamma \quad (5.9)$$

$$\dot{\chi} = \frac{F_{lift} \sin \phi \cos(\chi - \psi)}{m V_g \cos \gamma} \quad (5.10)$$

$$\dot{V}_g = \frac{F_{thrust}}{m} - \frac{F_{drag}}{m} - g \sin \gamma \quad (5.11)$$

$$\dot{\gamma} = \frac{F_{lift}}{m V_g} \cos \phi - \frac{g}{V_g} - \cos \gamma \quad (5.12)$$

$$\dot{h} = V_g \sin \gamma \quad (5.13)$$

5.3 Objective

Objective is to assess and demonstrate the use of LGVF method using the Ultra stick UAV model in the presence of wind disturbance magnitude with a view to conduct an experimental flight test for the GULMA UAV. This guidance system is applied to the non-linear UAV simulation model. The full system architecture

as described in [5] is in Figure 5.1. The autopilot block is implemented using low-level PID based control algorithms for the lateral and longitudinal motions of the UAV [23].

5.4 Lateral Guidance

5.4.1 Straight Path Following

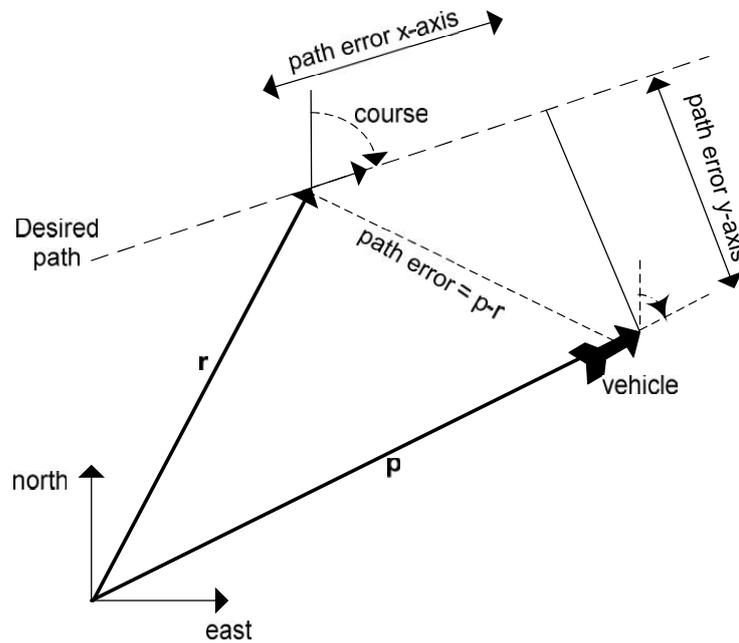


Figure 5.2: Geometry for lateral guidance of UAV to follow a straight-line path.

The lateral guidance strategy for path-following in [12] [3] is adopted from the frame relative to the path being followed. The path for a straight line is described in Figure 5.2 as

$$path(r, q) = \{x \in \mathbb{R}^3 : x = r + \lambda q, \lambda \in \mathbb{R}\} \quad (5.14)$$

where r is the initiation of the path, q is the vector for indicating the direction of the path, $(q_{north}, q_{east}, q_{down})$. The course angle, χ_{path} is given as

$$\chi_{path} \doteq \text{atan2} \frac{q_{east}}{q_{north}} \quad (5.15)$$

Now projecting, r , as the center of the path frame, $path$, with the x-axis (i.e north east) and z-axis as the inertial down axis we get the conventional coordinate system that translate the direction from the inertial frame to the path frame given as:

$$R \doteq \begin{bmatrix} \cos \chi_{path} & \sin \chi_{path} & 0 \\ -\sin \chi_{path} & \cos \chi_{path} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.16)$$

$$\begin{bmatrix} \dot{e}_{pathx} \\ \dot{e}_{pathy} \end{bmatrix} = \begin{bmatrix} \cos \chi_{path} & \sin \chi_{path} \\ -\sin \chi_{path} & \cos \chi_{path} \end{bmatrix} \begin{bmatrix} V_a \cos \chi \\ V_a \sin \chi \end{bmatrix} = \begin{bmatrix} V_a \cos(\chi - \chi_{path}) \\ V_a \sin(\chi - \chi_{path}) \end{bmatrix} \quad (5.17)$$

The relative path error from the geometry of Figure 5.2 is seen as:

$$e_{path} = \begin{bmatrix} e_{pathx} \\ e_{pathy} \\ e_{pathz} \end{bmatrix} \doteq R(path_i - r_i) \quad (5.18)$$

Hence for path follower the path error y-axis, e_{pathy} , must tend to zero steering a commanded course angle, χ_{cmd} .

$$e_{pathy} = V_a \sin(\chi - \chi_{path}) + \omega_{wind} \quad (5.19)$$

So the straight-line following guidance problem for lateral-axis is to formulate χ_{cmd} so that e_{pathy} tends to zero for the specified course angle. ω_{wind} is the wind vector in the lateral plane.

So therefore using the Lyapunov second method to define the function, $V(e_{pathy}) = 1/2 e_{pathy}^2$ and $\chi = \chi_q + \chi^d(e_{pathy})$

Now,

$$\dot{W} = e_{pathy} \dot{e}_{pathy}$$

substituting for e_{pathy} from Equation 5.19

$$\dot{W} = -V_a e_{pathy} \sin(\chi^\infty \tan^{-1}(k_{path} e_{pathy})) < 0$$

$\forall e_{pathy} \neq 0, \therefore e_{pathy} \rightarrow 0$ asymptotically. Subsequently, the command for the lateral path is

$$\chi_{cmd} = \chi_{path} - \frac{2}{\pi} \chi^\infty \tan^{-1}(k_{path} e_{pathy}) \quad (5.20)$$

In the case for a wind disturbance magnitude the air vehicle would induce an opposition to a drift (crab angle) to maintain the desired course as seen in Figure 5.3. We hence add an integral controller with the lateral command path in Equation 5.26 [5].

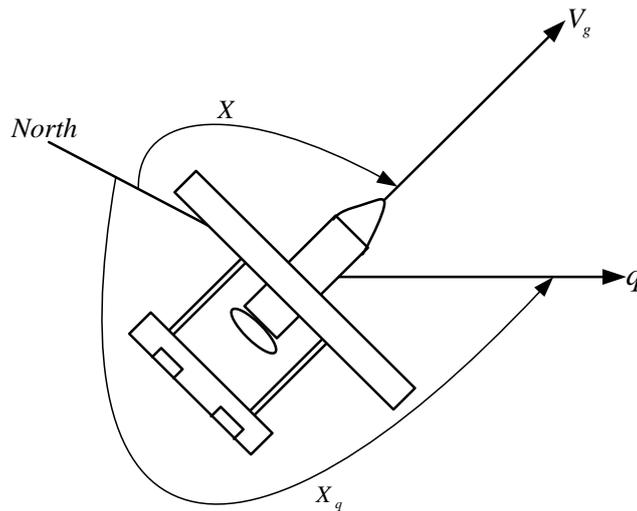


Figure 5.3: Crab angle due to wind disturbance.

$$\chi_{cmd} = \chi_{path} - \frac{2}{\pi} \chi^\infty \tan^{-1}(k_{path} e_{pathy}) + K_I \int_{-\infty}^t e_{pathy}(\tau) \delta\tau \quad (5.21)$$

where,

- K_I is the Integral control for crab angle.
- k_{path} vector fields fast/slow transition positive constants usually between 0 – 1.
- χ^∞ initial UAV course angle constrained to $(0, \frac{\pi}{2})$.

5.4.2 Circular Path Following

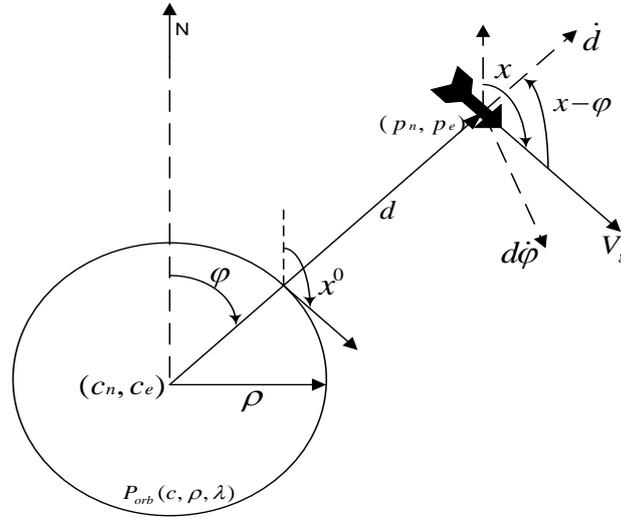


Figure 5.4: Geometry for lateral guidance of UAV to follow a circular path.

The circular path geometry we consider as described in [6] is shown in Figure 5.4. The UAV is to follow the circular path smoothly and converging with improved stability performance. Here we also adopt the LGVF method.

$$\begin{bmatrix} \dot{p}_n \\ \dot{p}_e \end{bmatrix} = \begin{bmatrix} V_a \cos \chi \\ V_a \sin \chi \end{bmatrix} \quad (5.22)$$

converting to polar coordinates as done in the straight line strategy we have

$$\begin{aligned} \begin{bmatrix} \dot{d} \\ d\dot{\varphi} \end{bmatrix} &= \begin{bmatrix} \cos \varphi & \sin \varphi \\ -\sin \varphi & \cos \varphi \end{bmatrix} \begin{bmatrix} \dot{p}_n \\ \dot{p}_e \end{bmatrix} \\ &= \begin{bmatrix} \cos \varphi & \sin \varphi \\ -\sin \varphi & \cos \varphi \end{bmatrix} \begin{bmatrix} V_a \cos \chi \\ V_a \sin \chi \end{bmatrix} \\ &= \begin{bmatrix} V_a \cos(\chi - \varphi) \\ V_a \sin(\chi - \varphi) \end{bmatrix} \end{aligned}$$

and

$$\chi^0 = \varphi + \lambda \frac{\pi}{2}$$

When $d \gg \rho \rightarrow \chi \approx \chi^0 + \lambda \frac{\pi}{2}$

When $d = \rho \rightarrow \chi = \chi^0$

Therefore, let the desired course angle be

$$\chi(d - \rho, \lambda) = \chi^0 + \lambda \tan^{-1} \left(k_{\text{circ}} \left(\frac{d - \rho}{\rho} \right) \right) \quad (5.23)$$

The commanded course is

$$\chi_{\text{cmd}} = \varphi + \lambda \left[\frac{\pi}{2} + \tan^{-1} \left(k_{\text{circ}} \left(\frac{d - \rho}{\rho} \right) \right) \right] + K_c \int_{\infty}^t e_{\text{pathy}}(\tau) \delta \tau \quad (5.24)$$

- K_c is the Integral control for crab angle during circular motion.
- k_{circ} vector fields for fast/slow transition positive constants during circular manoeuvre usually between 0 – 1.

- φ phase angle relative to UAV position $(0, 2\pi)$.
- λ shows the direction of UAV either clockwise or anti-clockwise 1 or -1 respectively.

The circular angle must be wrapped:

$$\varphi = \text{atan2}(p_e - c_e, p_n - c_n) + 2\pi m$$

The $\int_{\infty}^t e_{\text{pathy}}(\tau) \delta\tau$ term in the heading command, χ_{cmd} , is the integral controller for the error due to wind vector.

5.5 Longitudinal guidance

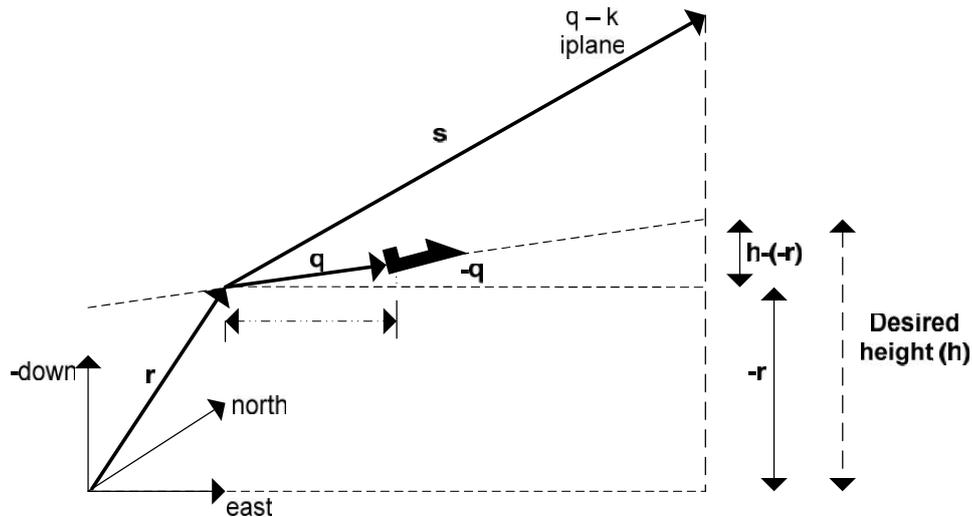


Figure 5.5: Geometry for longitudinal guidance of UAV.

The LQR longitudinal control approach described in Chapter 4 is adopted here for the guidance system. We shall augment the measured altitude state with an integral control in order to track the reference altitude signal. So we have,

$$\dot{x} = A_{lon}x + B_{lon}u \quad (5.25)$$

$$z = Hx$$

where z is the measured altitude output. Objective is to drive z to a reference signal h_r . The augmented state with an integrator is

$$x_i = \int_0^t (z(\tau) - h_r) \delta_\tau \quad (5.26)$$

let's define the augmented state as

$$\xi = (x^T, x_i^T)^T$$

hence

$$\dot{\xi} = A_{aug}\xi + B_{aug}u \quad (5.27)$$

$$A_{aug} = \begin{bmatrix} A & 0 \\ H & 0 \end{bmatrix} \quad B_{aug} = \begin{bmatrix} B \\ 0 \end{bmatrix} \quad (5.28)$$

$$(5.29)$$

So now considering the longitudinal state space equations described in Chapter 4

$$\dot{x} = A_{lon}x_{lon} + B_{lon}u_{lon} \quad (5.30)$$

where $x_{lon} = (u, w, q, \theta, h)^T$ and $u_{lon} = (\delta_a, \delta_t)^T$. The objective is to track altitude, h , to a reference altitude signal, h_r and UAV airspeed V_a to commanded airspeed, V_c . Hence we proceed to augment as in Equation 5.26.

$$\begin{aligned}
x_i &= \begin{bmatrix} \int (h - h_r) \delta_t \\ \int (V_a - V_c) \delta_t \end{bmatrix} \\
&= \int (H_{lon} - \begin{bmatrix} h_r \\ V_c \end{bmatrix}) \delta_t
\end{aligned} \tag{5.31}$$

where

$$H_{lon} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

5.6 Kalman Estimator Dynamics

In reality, aircraft state estimates (sensors) are required to effectively control the air vehicles as there are no direct measurements of these states. Hence using the dynamic observer theory as described by Luenberger in [6] the air vehicle's rate of change of velocity, angular rates, altitude and airspeed in terms of change of static and dynamic pressure respectively, e.t.c. are measured [80]. In the simulation model direct feedback of the true states of the UAV is also assumed which is compared with the estimated states. The UAV state system model in terms of sensor and measurement noises is described as [6]:

$$\dot{x} = f(x, u) + \xi \tag{5.32}$$

$$y[n] = h(x[n], u[n]) + \eta[n], \tag{5.33}$$

where $\xi \sim \mathcal{N}(0, Q)$ is the process noise, $\eta_i \sim \mathcal{N}(0, R)$ is the measurement noise.

The continuous-time observer for this system is given by

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x}) \quad (5.34)$$

where \hat{x} is the estimated value of x and the observation error defined as $\tilde{x} = x - \hat{x}$

$$\dot{\tilde{x}} = (A - LC)\tilde{x}.$$

Therefore, observation error $\rightarrow 0$ if $\text{eig}(A - LC)$ is stable.

For a non linear propagation model with respect to non linear model described in Chapter 3 we have:

$$\dot{\phi} = p + q \sin \phi \tan \theta + r \cos \phi \tan \theta + \xi_{\phi} \quad (5.35)$$

$$\dot{\theta} = q \cos \phi - r \sin \phi + \xi_{\theta} \quad (5.36)$$

where

$$\xi_{\phi} \sim \mathcal{N}(0, Q_{\phi}) \quad \text{and} \quad \xi_{\theta} \sim \mathcal{N}(0, Q_{\theta})$$

Accelerometers for measured outputs are described as:

$$y_{\text{accel}} = \begin{pmatrix} \dot{u} + qw - rv + g \sin \theta \\ \dot{v} + ru - pw - g \cos \theta \sin \phi \\ \dot{w} + pv - qu - g \cos \theta \cos \phi \end{pmatrix} + \eta_{\text{accel}}. \quad (5.37)$$

if $\dot{u} = \dot{v} = \dot{w} \approx 0$

and

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} \approx V_a \begin{bmatrix} \cos \alpha \cos \beta \\ \sin \beta \\ \sin \alpha \cos \beta \end{bmatrix}$$

now assuming that $\alpha \approx \theta$ and $\beta \approx 0$, gives

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} \approx V_a \begin{bmatrix} \cos \theta \\ 0 \\ \sin \theta \end{bmatrix}$$

we see that

$$y_{\text{accel}} = \begin{bmatrix} qV_a \sin \theta + g \sin \theta \\ rV_a \cos \theta - pV_a \sin \theta - g \cos \theta \sin \phi \\ -qV_a \cos \theta - g \cos \theta \cos \phi \end{bmatrix} + \eta_{\text{accel}} \quad (5.38)$$

Now $x = (\phi, \theta)^\top$, $u = (p, q, r, V_a)^\top$, $\xi = (\xi_\phi, \xi_\theta)^\top$, and $\eta = (\eta_\phi, \eta_\theta)^\top$, gives

$$\dot{x} = f(x, u) + \xi$$

$$y = h(x, u) + \eta,$$

where

$$f(x, u) = \begin{bmatrix} p + q \sin \phi \tan \theta + r \cos \phi \tan \theta \\ q \cos \phi - r \sin \phi \end{bmatrix} \quad (5.39)$$

$$h(x, u) = \begin{bmatrix} qV_a \sin \theta + g \sin \theta \\ rV_a \cos \theta - pV_a \sin \theta - g \cos \theta \sin \phi \\ -qV_a \cos \theta - g \cos \theta \cos \phi \end{bmatrix} \quad (5.40)$$

According to [6], implementation of Kalman filter requires Jacobians: $\frac{\delta f}{\delta x}$ and $\frac{\delta h}{\delta x}$.

$$\frac{\delta f}{\delta x} = \begin{bmatrix} q \cos \phi \tan \theta + r \sin \phi \tan \theta & \frac{q \sin \phi - r \cos \phi}{\cos^2 \theta} \\ -q \sin \phi - r \cos \phi & 0 \end{bmatrix} \quad (5.41)$$

$$\frac{\delta h}{\delta x} = \begin{bmatrix} 0 & qV_a \cos \theta + g \cos \theta \\ -g \cos \phi \cos \theta & -rV_a \sin \theta - pV_a \cos \theta + g \sin \phi \sin \theta \\ g \sin \phi \cos \theta & (qV_a + g \cos \phi) \sin \theta \end{bmatrix}. \quad (5.42)$$

Other states were measured using a low pass filter (LPF) method. We shall use a simple unity DC gain LPF with a cut-off frequency of $3Hz$. The filter is represented as

$$y_{n+1} = e^{-3T_{st}}y(n) + (1 - e^{-3T_{st}})u(n) \quad (5.43)$$

we assume $\rho_{LPF} = e^{-3T_{st}}$, then

$$y_{n+1} = \rho_{LPF}y(n) + (1 - \rho_{LPF})u(n) \quad (5.44)$$

where y_{n+1} is update of the measured output and input, u , for sampling time, T_{st} . Also $\rho_{LPF} \in [0, 1]$. The results for using the LPF method is given as follows:

$$\hat{p} = y_{n+1}(y_{\text{gyro},x}) \quad (5.45)$$

$$\hat{q} = y_{n+1}(y_{\text{gyro},y}) \quad (5.46)$$

$$\hat{r} = y_{n+1}(y_{\text{gyro},z}) \quad (5.47)$$

$$\hat{h} = \frac{y_{n+1}(y_{\text{static pres}})}{\rho g} \quad (5.48)$$

$$\hat{V}_a = \sqrt{\frac{2}{\rho} y_{n+1}(y_{\text{diff pres}})} \quad (5.49)$$

While ground speed, position (north and east) and course is obtained using the

GPS sensor model as,

$$\hat{p}_n = y_{n+1}(y_{\text{GPS},n}) \quad (5.50)$$

$$\hat{p}_e = y_{n+1}(y_{\text{GPS},e}) \quad (5.51)$$

$$\hat{\chi} = y_{n+1}(y_{\text{GPS},\chi}) \quad (5.52)$$

$$\hat{V}_g = y_{n+1}(y_{\text{GPS},V_g}) \quad (5.53)$$

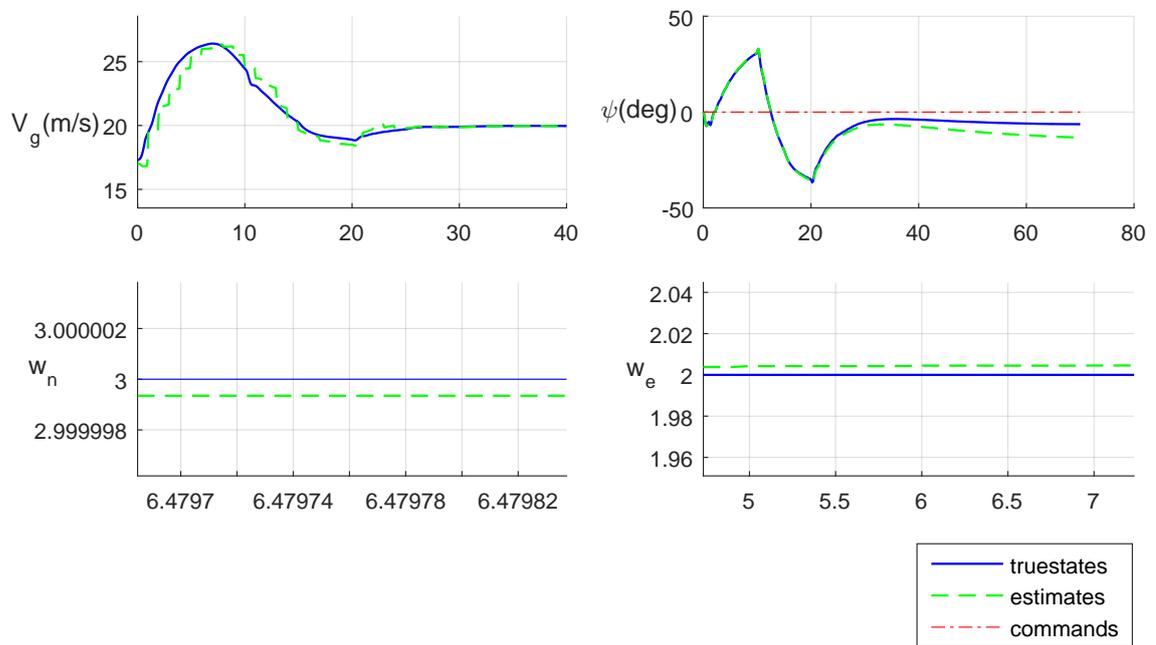


Figure 5.6: State estimation via EKF.

We observe in Figures 5.6 and 5.7 that the estimated states closely follow the true states which indicates that the implemented EKF sensor model gives an acceptable result.

5.7 Simulation Results

The results of the simulation was obtained via tuning of the inner PID loops and outer guidance loop. The parameters used is listed in Table 5.1 are for straight

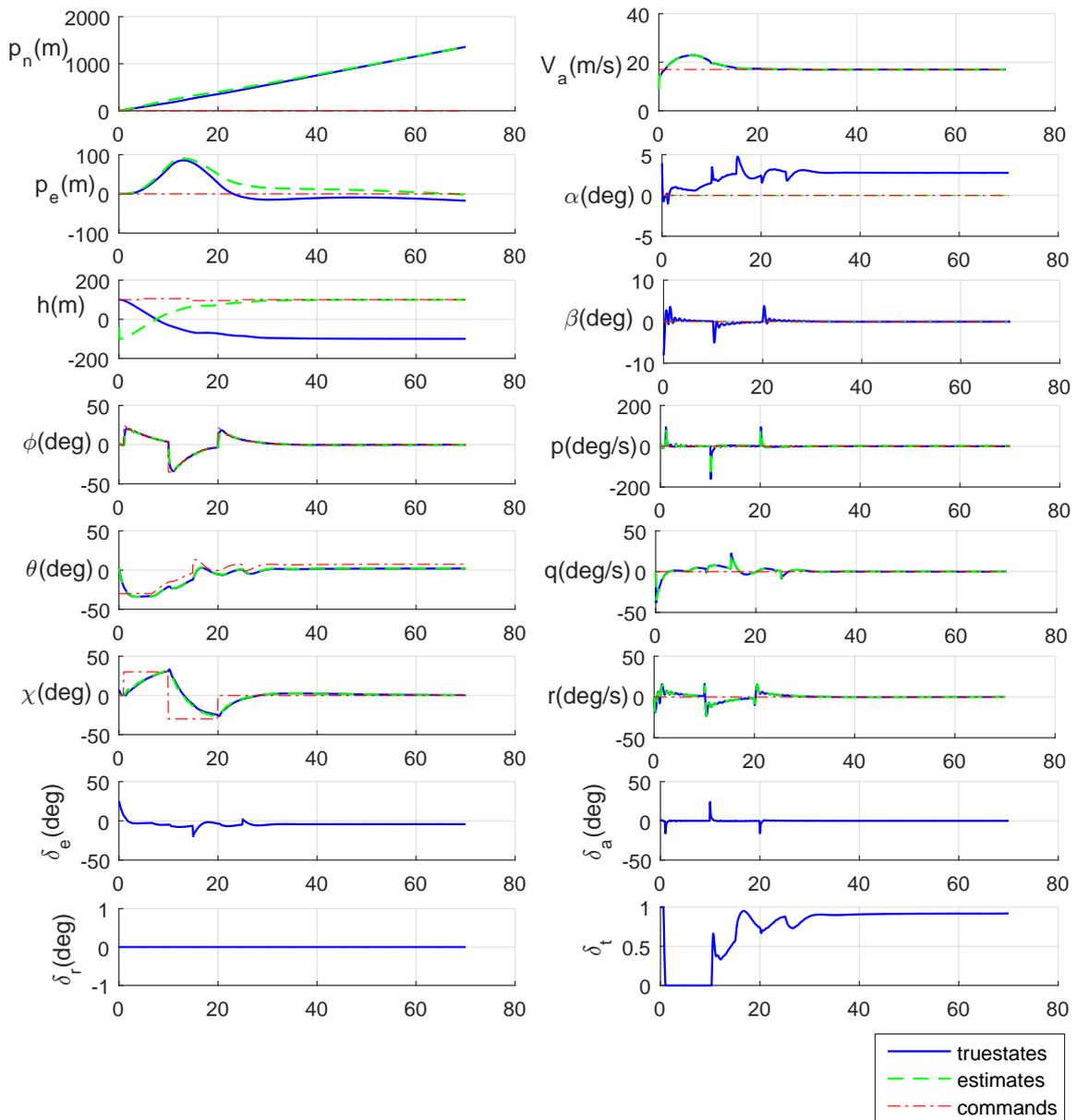


Figure 5.7: Estimation error for states via EKF.

line path tracking while for circular path tracking is listed in Table 5.2. The LGVF method is implemented as described in Figure 5.8.

5.7.1 Summary

Aircraft modelling using commercial tools such as AVL which model the aircraft in terms of aerodynamic and flight dynamics analysis used for the calculation of

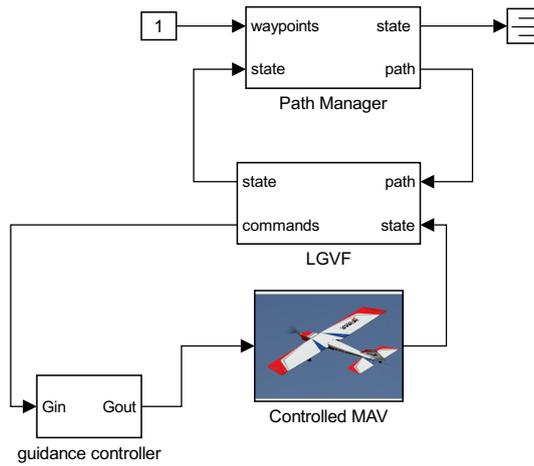


Figure 5.8: Matlab/Simulink LGVF simulation diagram.

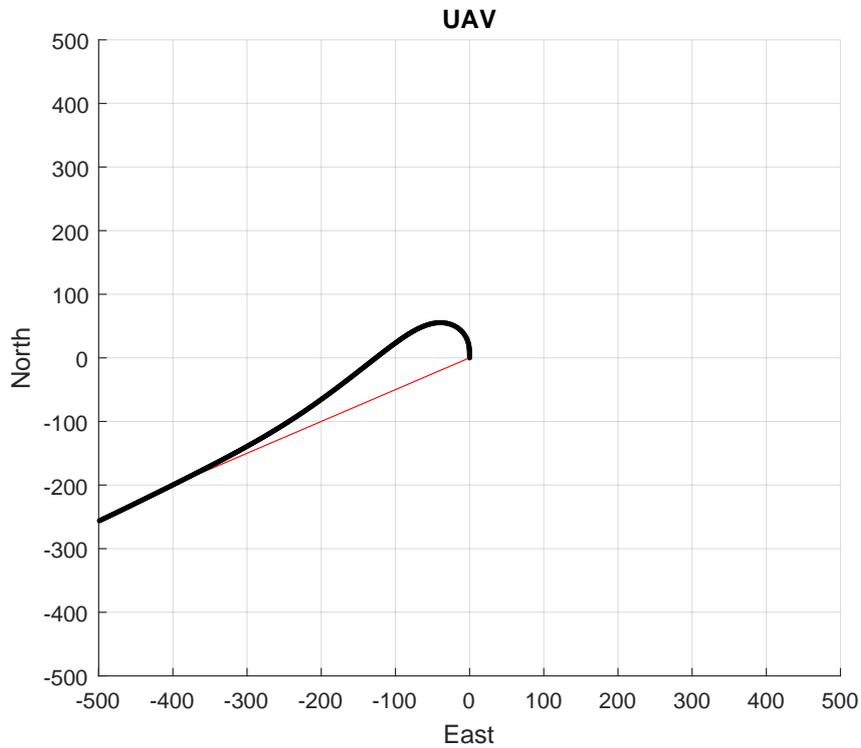
Table 5.1: Parameters for straight line path tracking.

Parameters	Values	Range
Direction of path (q)	$[-0.5, -1, -0.05]^T m$	-
Initial position of path (r)	$[0, 0, -100]^T m$	-
UAV velocity	$17m/s$	$10m/s - 23m/s$
UAV initial position	$[0, 0, -100]^T m$	-
k_{path}	0.03	0.02 - 0.03
ϕ_{max}	35°	-
K_I	0.00075	-
$wind_n$	$1m/s$	$1 - 3m/s$
$wind_e$	$-1m/s$	$-3m/s$ to $-1m/s$

stability derivatives. Guidance algorithms on their side are typically based on sensor fusion using extended Kalman filtering. The LGVF algorithm provided a low cost and high performance solution and was adapted to the Ultra stick UAV model. The guidance framework used was developed by Beard and Randal for small autonomous fixed wing UAV. The LGVF technique guaranteed convergence for straight line and circular path tracking of the vehicle. Tuning of PID controllers for stabilization of the inner loop was conducted and to guarantee outer loop stability in the presence of wind disturbances in the range of $(-1 \leq wind_{n,e} \pm 3)m/s$ the integral controller was added to the loop.

Table 5.2: Parameters for circular path tracking.

Parameters	Values	Range
Initial position of path (r)	$[0, 0, -100]^T m$	-
Initial direction of circular path (λ)	1	± 1
UAV velocity	$13m/s$	$10m/s - 23m/s$
UAV initial position	$[0, 0, -100]^T m$	nil
k_{path}	0.05	$0.05 - 0.7$
ϕ_{max}	35°	nil
K_c	0.02	$0.02 - 0.0356$
$wind_n$	$1m/s$	$1 - 3m/s$
$wind_e$	$-1m/s$	$-3m/s$ to $-1m/s$

Figure 5.9: UAV straight line path error due to $wind_{n,e} = 1m/s$.

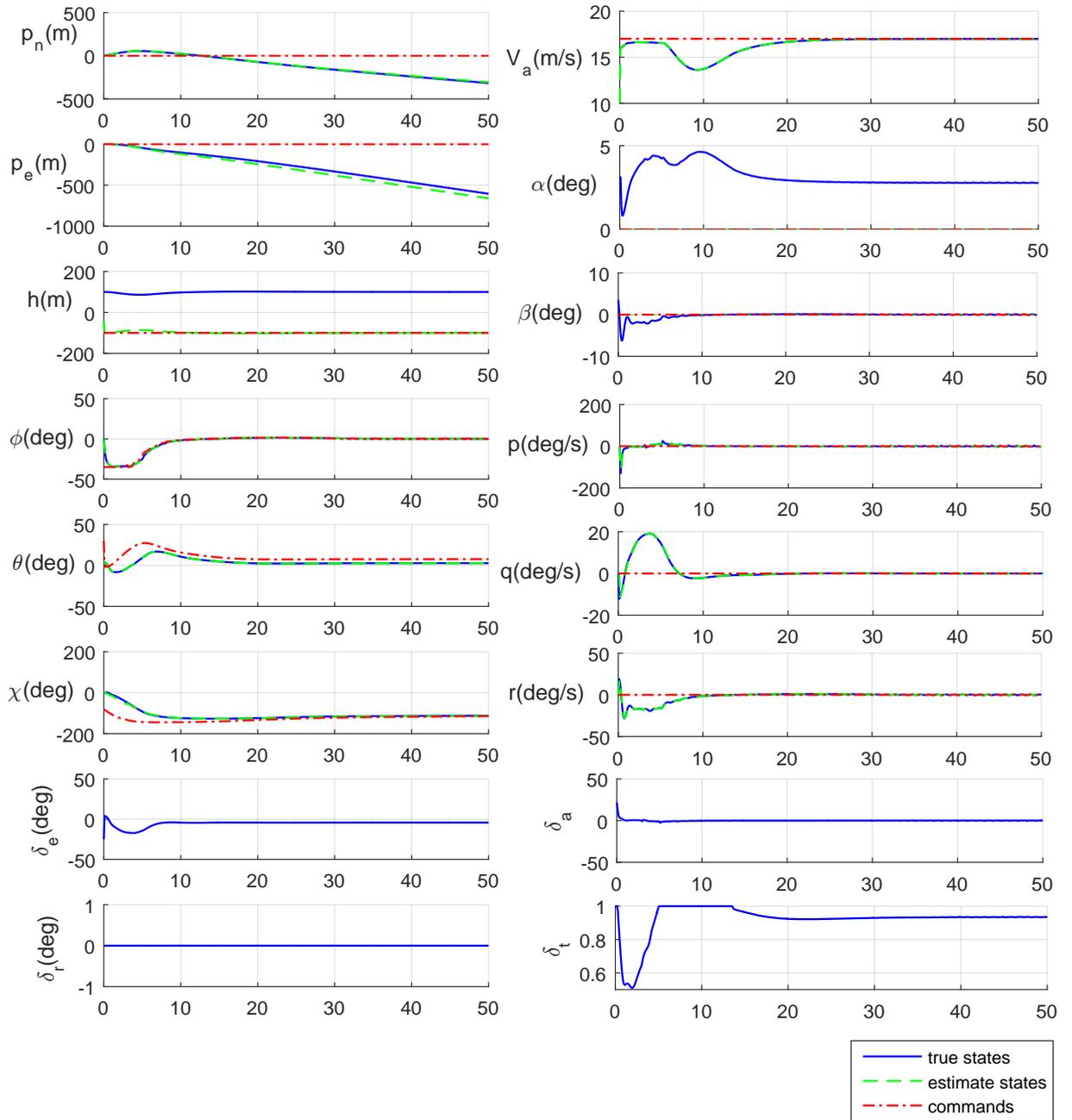


Figure 5.10: Response to straight line path tracking due to $wind_{n,e} = 1m/s$.

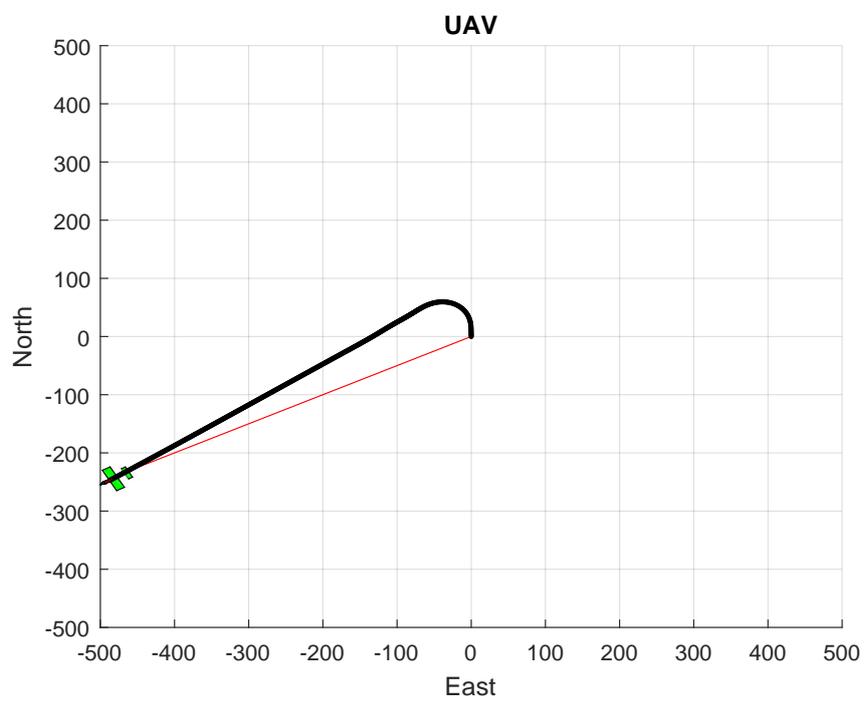


Figure 5.11: UAV straight line path error due to $wind_{n,e} = 3m/s$.

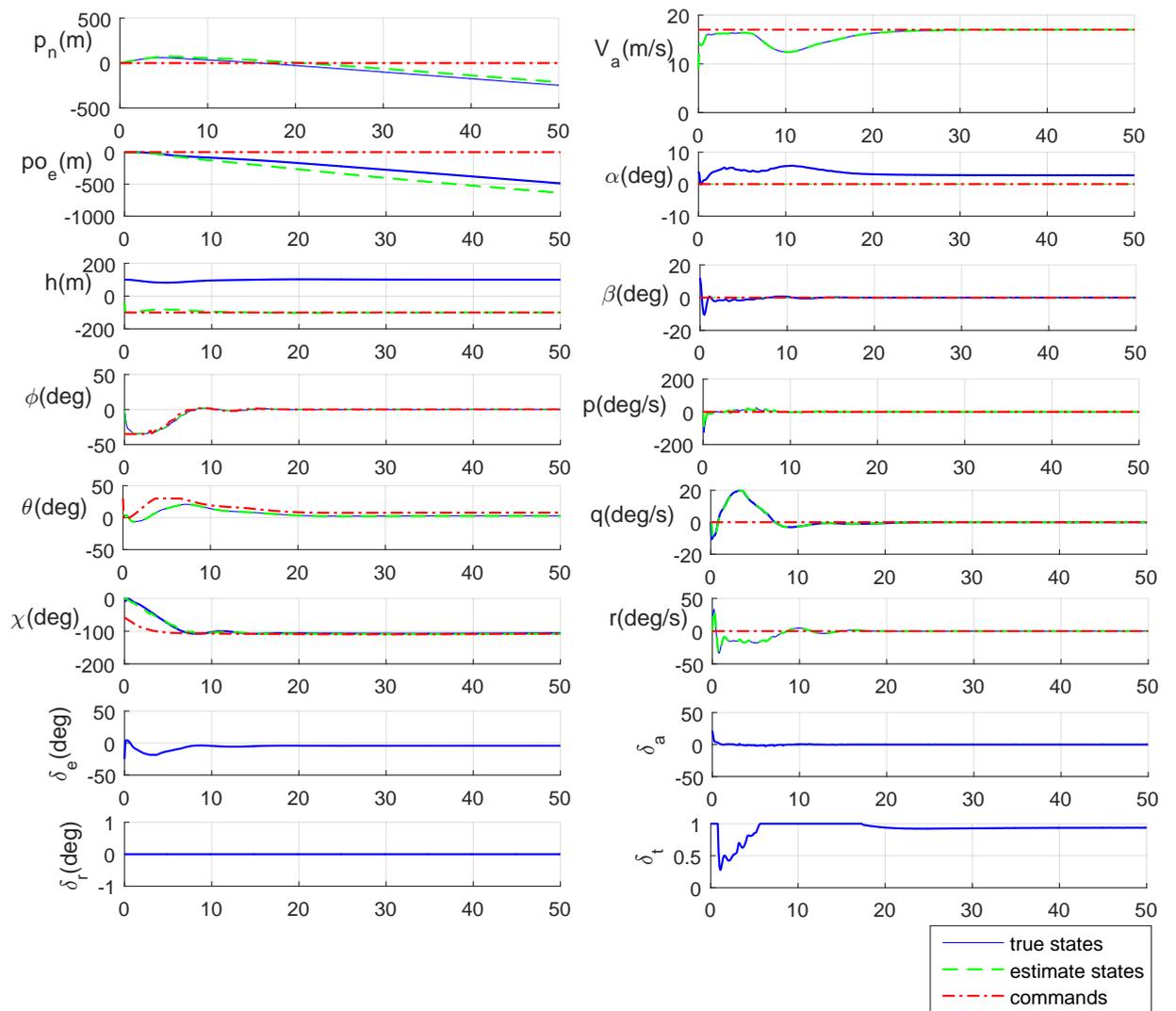


Figure 5.12: Response to straight line path tracking with $wind_{n,e} = 3m/s$.

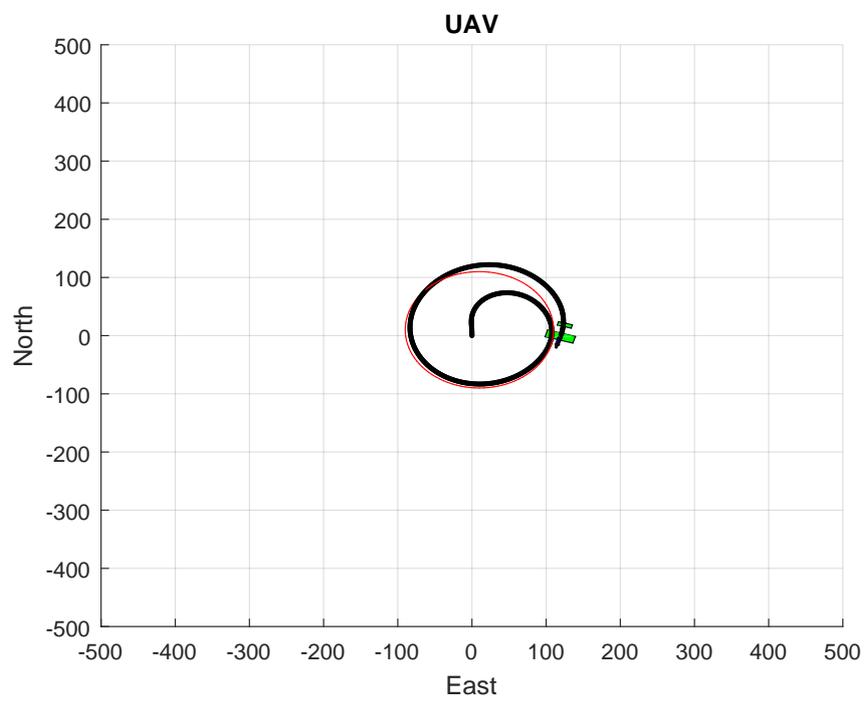
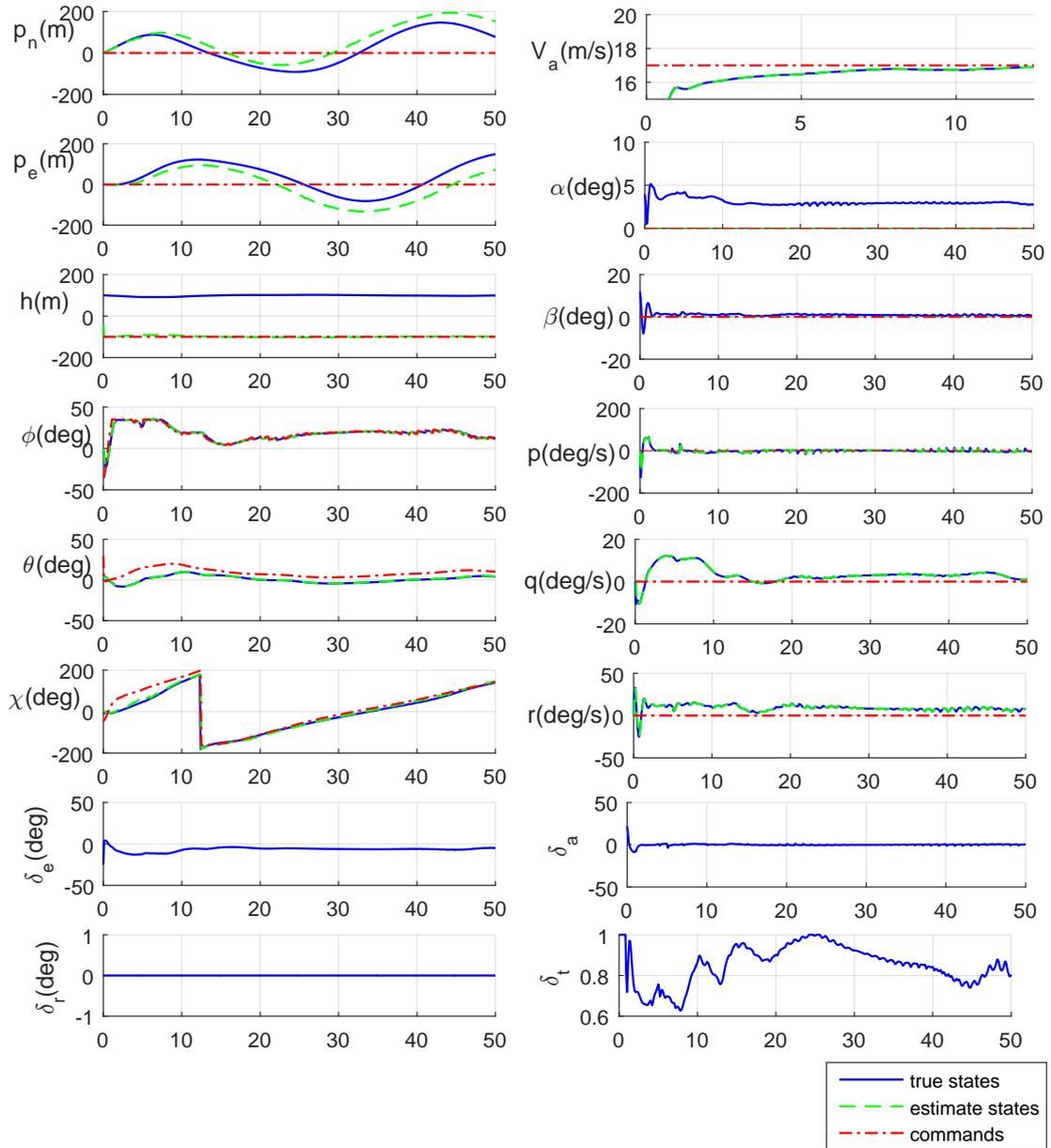


Figure 5.13: UAV circular path error due to $wind_{n,e} = 1m/s$.

Figure 5.14: Response to circular path tracking with $wind_{n,e} = 1m/s$.

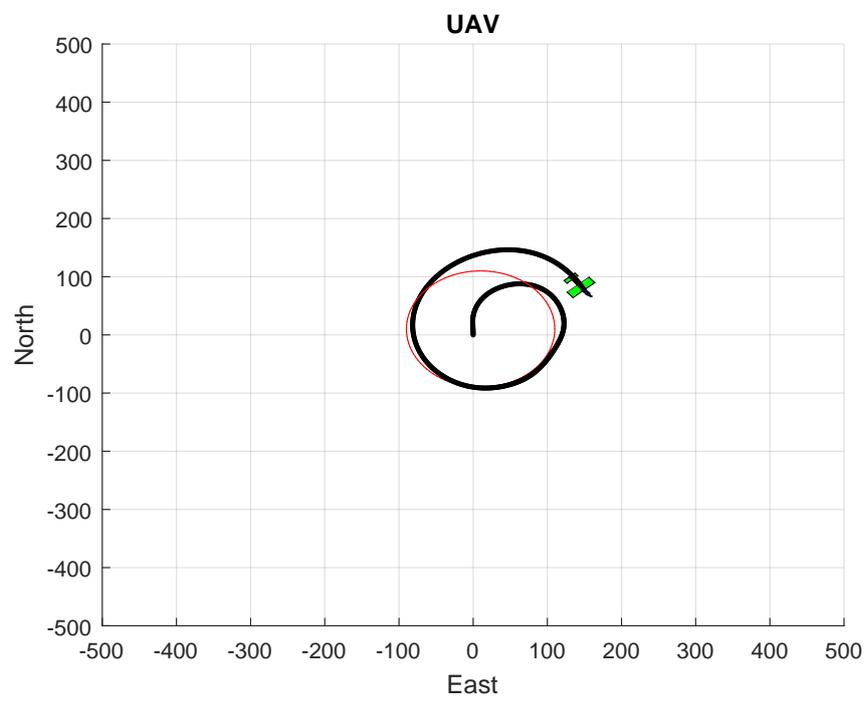


Figure 5.15: UAV circular path error due to $wind_{n,e} = 3m/s$.

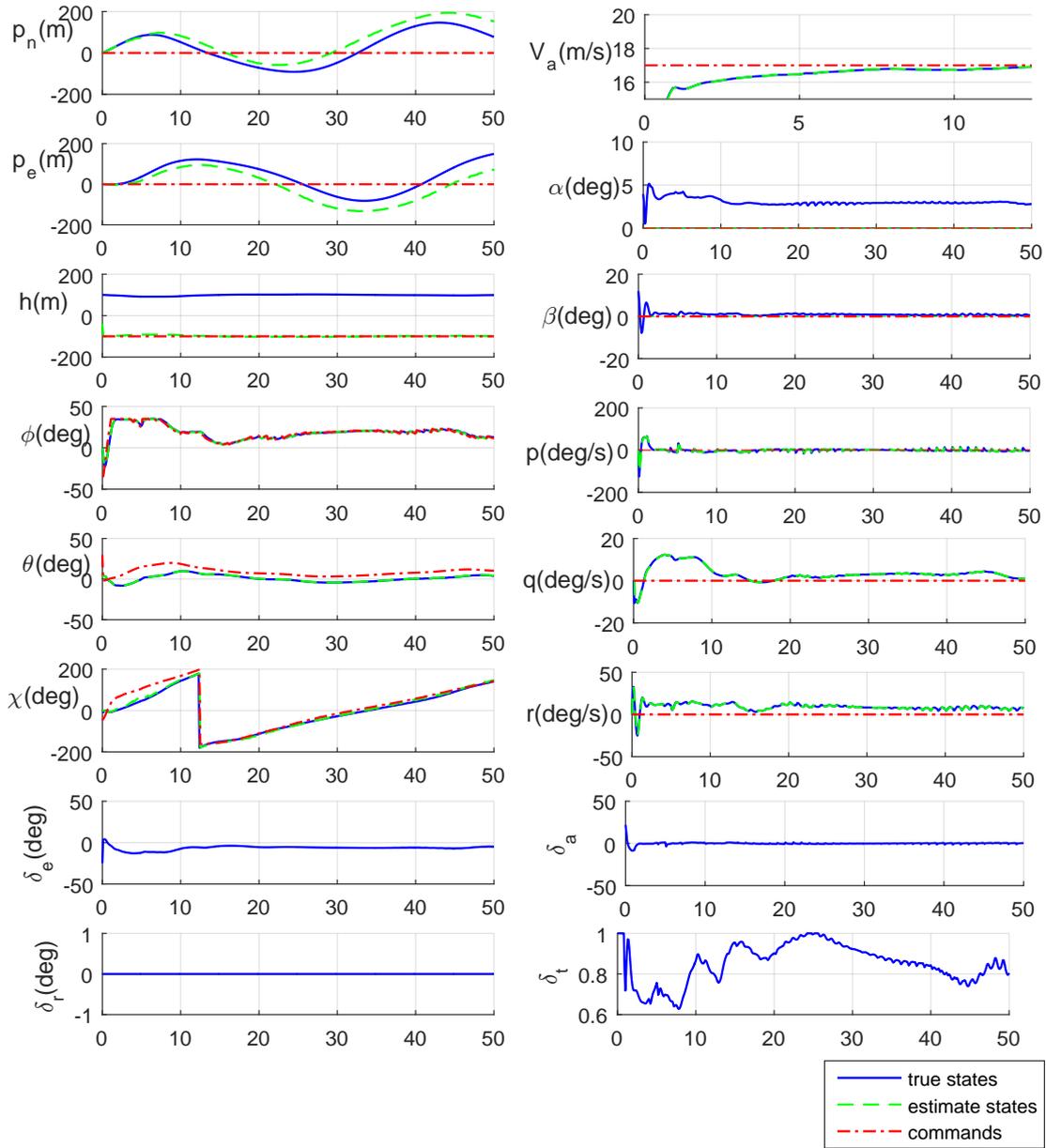


Figure 5.16: Response to circular path tracking with $wind_{n,e} = 3m/s$.

Chapter 6

Conclusion and Further Work

6.1 Conclusions

In conclusion, to enhance and facilitate the NAF UAV program, there was a need to develop a simulation model of its air vehicle. Hence, the development of the GULMA UAV non linear simulation model was accomplished using AVL and Matlab/Simulink software in this thesis. Linearization of the non linear model was successfully computed for steady flight and climbing turn velocity values of $43m/s$ and $35m/s$ respectively, zero flight path angle and altitude of $100m$. The corresponding linear models- longitudinal and lateral-directional were obtained for the UAV. Control design and modal analysis of the models provided satisfactory results using the LQR/LQG control design technique. The control problem for roll-yaw coupling existing in the lateral-directional dynamic model was addressed. The design method was to select and tune for optimal weighting matrices using a developed multi-objective PSO algorithm scheme that computed the best fit matrices for determination of the optimal LQR state feedback control gains. The result obtained from multi-objective PSO algorithm was compared to the conventional PSO algorithm which indicated a better response to a doublet input to roll angle of $< 5secs$. The response was well damped with a rise

time $< 2secs$ and an overshoot $< 10\%$. The actuator response corresponding to the doublet was maintained within the aileron permissible limits.

The coupling results for yaw output response and rudder actuator also indicated that while the cost function minimizes the control input, the optimal weighting matrices using the multi-objective PSO algorithm described guaranteed that the yaw output response value of $< 1.8^\circ$ due to the control input of $< 0.2^\circ$ was an improvement when compared to output response using the conventional PSO algorithm.

A 2-DOF model-based H_∞ and μ robust controller design and synthesis was conducted for the lateral axis GULMA UAV model that was linearized at $35m/s$ during a climb turn flight condition. Assessment of the synthesized controllers in terms of the performance objectives indicated that the performance of H_∞ technique was satisfactory for the nominal model and poor performance for the worse-case uncertain models. However, the μ controller performance was satisfactory for both nominal and uncertain models by maintaining the gain peak of $1 \leq \gamma \leq 1.2$ in the frequency domain. Furthermore in the time domain test for performance and robustness of the μ controller gave satisfactory roll and yaw output responses for a 50% increase and decrease in the commanded roll and yaw input configuration while when compared to time domain responses for the H_∞ controller, the results were unsatisfactory. This suggests that the aircraft in climbing turn flight condition showed promising robust stability for the roll to rate motion of the system. A robust stability value of 4.64 meant the system satisfied the lower bound of > 1 and a tolerance value, 464%, of the plant uncertainty. Hence, using the μ controller we can synthesize a robust controller for the GULMA UAV when conducting flight experiments.

Aircraft modelling using tools such as AVL which model the aircraft in terms of aerodynamic and flight dynamics analysis used for the calculation of stability derivatives. Guidance algorithms on their side are typically based on sensor fu-

sion using extended Kalman filtering. The LGVF algorithm provided a low cost and high performance solution which was demonstrated using the Ultra stick UAV non linear simulation model. The guidance framework used was developed by Beard and Randal for small autonomous fixed wing UAV. The LGVF technique guaranteed convergence for straight line and circular path tracking of the vehicle. Tuning of PID controllers for stabilization of the inner loop was conducted and to guarantee outer loop stability in the presence of wind disturbances in the range of $(-1 \leq wind_{n,e} \pm 3)$ m/s an integral controller was added to the loop.

A basic requirement for pre-flight checks for flight experiments and testing is that control and guidance algorithms are to be tested using a HIL (Hardware in the Loop) environment in which the navigation and flight control software is performed in real-time hardware on-board a UAV. This hardware is connected to a set of systems that emulate the flight dynamics of the aircraft as well as the sensor data that will be employed by the navigation algorithm. Positioning data could be simulated employing GPS sensor including the errors that are common to these sensors.

The Matlab/Simulink code generated using this framework could be compiled ready to be used on real-time operating system running on an on-board embedded controller such as the Arduino hardware, which brings a high degree of flexibility for adapting to changes in mission and a simple open interface for running the C codes.

6.2 Further Work

Recommendation for future work for improving GULMA UAV flight performance include the following:

- Comparing experimental flight data analysis with this developed full linear

model simulation results using model identification techniques. This comparison would enable further analysis for rapid development of FCS and testing on newly designed GULMA UAV variant structures.

- Implementation of the LGVF design on the Arduino hardware as an on-board autopilot which would be used for UAV guidance system experimental testing and analysis.
- The linearization of the GULMA simulation model was done for velocity at 35 and 43 m/s, this could be further calculated for specified range of velocities using measured thrust or engine power setting data of the air vehicle.
- Adapt the LGVF design to the NAF unmanned combat aerial vehicle program.

Appendix A

AVL Modelling

A Performance Specifications for GULMA UAV

Critical performance specifications for GULMA UAV were obtained from CAD model provide vide reference documents and AVL analysis. The dimensional critical parameters are in Table A.1 while aerodynamic critical parameters are specified in Table A.2

Table A.1: Critical dimensional parameters.

Parameters	Values	Units
Maximum Take-off weight	47.04	Kg
Empty weight	32.1	<i>Kg</i>
Fuel weight	14.4	<i>Kg</i>
Tank Capacity	10.2	Litres
Wing Loading	36.7	<i>Kg/m²</i>
Wing reference area	1.3	<i>m²</i>
Aspect ratio	11.1	nil
Wing span	3.8	<i>m</i>
Taper ratio	0.7	nil
Root chord	0.4	<i>m</i>
Dihedral angle	0.1	deg
Cruise speed	44.5	<i>m/s</i>
Stall speed	20.6	<i>m/s</i>
Diving speed	62.3	<i>m/s</i>

Table A.2: Aerodynamic critical parameters.

Parameters	Symbol	Values
Maximum Coefficient of lift	Cl_{max}	1.38
Zero-lift Coefficient of drag	CD_0	0.0345
Induced Coefficient of drag	K	0.0357
Oswald efficiency	e	0.8
Zero-lift Coefficient of moment	CM_0	-0.0235
Lift-to-Drag	$(L/D)_{max}$	0.32
Reynolds number	Re	10^6
Neutral point	X_{np}	1.086

The specifications are required to perform an analysis of different flight conditions for the vehicle using the AVL interface. The result of the performance analysis is described in Figure A.1 is for sea level conditions and shown in Figure A.2 is for flight level of about 1000metres.

V (m/s)	CL	e	Cdi	Cdt	D(N)	L/D	Power	Cm
15	2.611	0.981	0.199	0.233	41.621	18.107	0.837	-1.065
17	2.033	0.994	0.119	0.153	35.156	18.914	0.801	-0.756
19	1.627	1.003	0.075	0.110	31.509	18.882	0.803	-0.566
21	1.332	1.009	0.050	0.085	29.675	18.139	0.836	-0.441
23	1.111	1.013	0.035	0.069	29.098	16.891	0.897	-0.355
25	0.940	1.016	0.025	0.059	29.445	15.356	0.987	-0.294
27	0.806	1.018	0.018	0.053	30.510	13.723	1.105	-0.248
29	0.699	1.020	0.014	0.048	32.156	12.122	1.251	-0.213
31	0.611	1.022	0.010	0.045	34.292	10.634	1.426	-0.186
33	0.539	1.011	0.008	0.043	36.932	9.275	1.634	-0.164
35	0.480	1.011	0.006	0.041	39.867	8.101	1.871	-0.147
37	0.429	1.006	0.005	0.040	43.176	7.076	2.142	-0.132
39	0.386	0.995	0.004	0.039	46.828	6.190	2.449	-0.120
41	0.350	0.982	0.004	0.038	50.777	5.430	2.792	-0.110
43	0.318	0.971	0.003	0.037	54.998	4.780	3.171	-0.102
45	0.290	0.976	0.002	0.037	59.416	4.228	3.586	-0.094
47	0.266	0.966	0.002	0.037	64.166	3.748	4.044	-0.088
49	0.245	0.956	0.002	0.036	69.165	3.335	4.545	-0.082
51	0.226	0.947	0.002	0.036	74.408	2.979	5.089	-0.078
53	0.209	0.939	0.001	0.036	79.894	2.670	5.678	-0.073
55	0.194	0.932	0.001	0.036	85.619	2.401	6.315	-0.070
57	0.181	0.926	0.001	0.036	91.580	2.166	7.000	-0.066
59	0.169	0.920	0.001	0.035	97.775	1.960	7.736	-0.063

Figure A.1: List of performance values for analysis at sea level at 1000 metres.

V	CL	e	Cdi	Cdt	D(N)	L/D	Power
15	4.018	0.944	0.489	0.523	60.715	15.398	1.221
17	3.129	0.968	0.289	0.323	48.189	17.118	1.099
19	2.505	0.984	0.182	0.217	40.348	18.292	1.028
21	2.050	0.994	0.121	0.155	35.331	18.900	0.995
23	1.709	1.001	0.083	0.118	32.160	18.959	0.992
25	1.447	1.006	0.059	0.094	30.274	18.529	1.015
27	1.240	1.010	0.043	0.078	29.328	17.710	1.062
29	1.075	1.013	0.033	0.067	29.100	16.617	1.132
31	0.941	1.016	0.025	0.059	29.441	15.365	1.224
33	0.830	1.011	0.019	0.054	30.318	14.016	1.342
35	0.738	1.011	0.015	0.050	31.524	12.710	1.480
37	0.660	1.006	0.012	0.047	33.111	11.447	1.643
39	0.594	0.995	0.010	0.045	35.029	10.265	1.832
41	0.538	0.982	0.008	0.043	37.211	9.192	2.046
43	0.489	0.971	0.007	0.042	39.615	8.232	2.284
45	0.446	1.039	0.005	0.040	41.759	7.463	2.520
47	0.409	1.025	0.005	0.039	44.628	6.686	2.813
49	0.377	1.013	0.004	0.039	47.682	6.002	3.133
51	0.348	1.001	0.003	0.038	50.914	5.401	3.482
53	0.322	0.990	0.003	0.037	54.319	4.871	3.861
55	0.299	0.980	0.003	0.037	57.894	4.404	4.270
57	0.278	0.971	0.002	0.037	61.635	3.992	4.711
59	0.260	0.963	0.002	0.037	65.540	3.627	5.186

Figure A.2: List of performance values for analysis at 1000 metres.

B Aerodynamic analysis for GULMA UAV

The air-foil employed in the design of the air vehicle wings was the NACA-2415 while the tail section was the NACA 0012. The respective coefficients of lift, drag and moment versus angle-of-attack of the vehicle were thus simulated using XFOIL programme in order to obtain the behaviour in critical flight conditions. Shown in Figure A.3 is the graphical representation of the results obtained from the simulation using the air-foil NACA 2415. In Figure A.4 we have the results for using the NACA 0012 for the tail section of the aircraft.

We obtained the best lift to weight ratio and power requirement for sea level as well as at 1000metres condition from the graphical representation of the lift to weight ratio versus the aircraft speed as shown in Figure A.5.

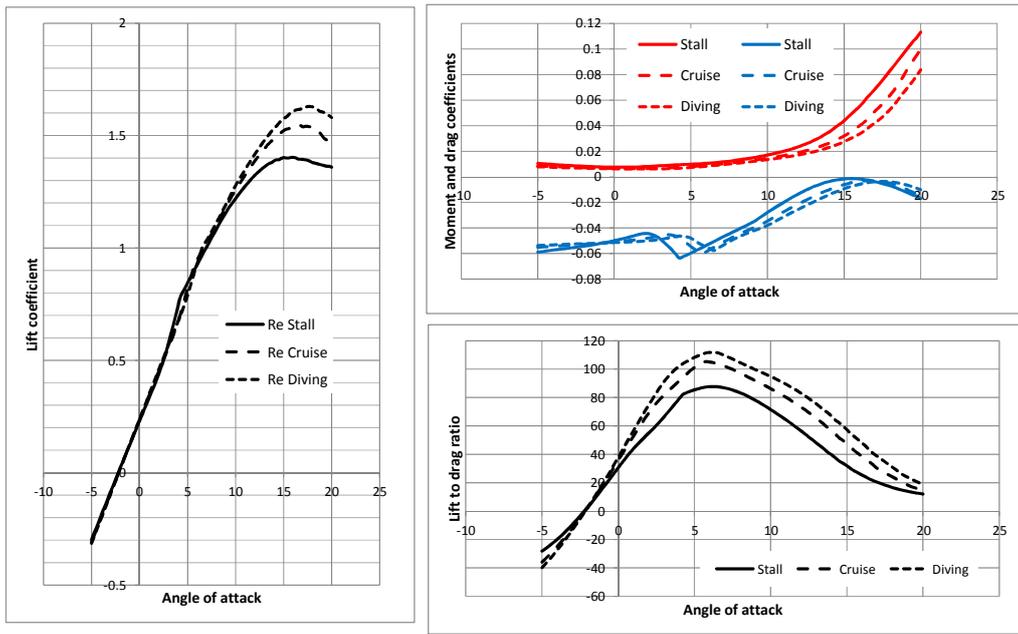


Figure A.3: Variation of coefficients for NACA 2415 air-foil.

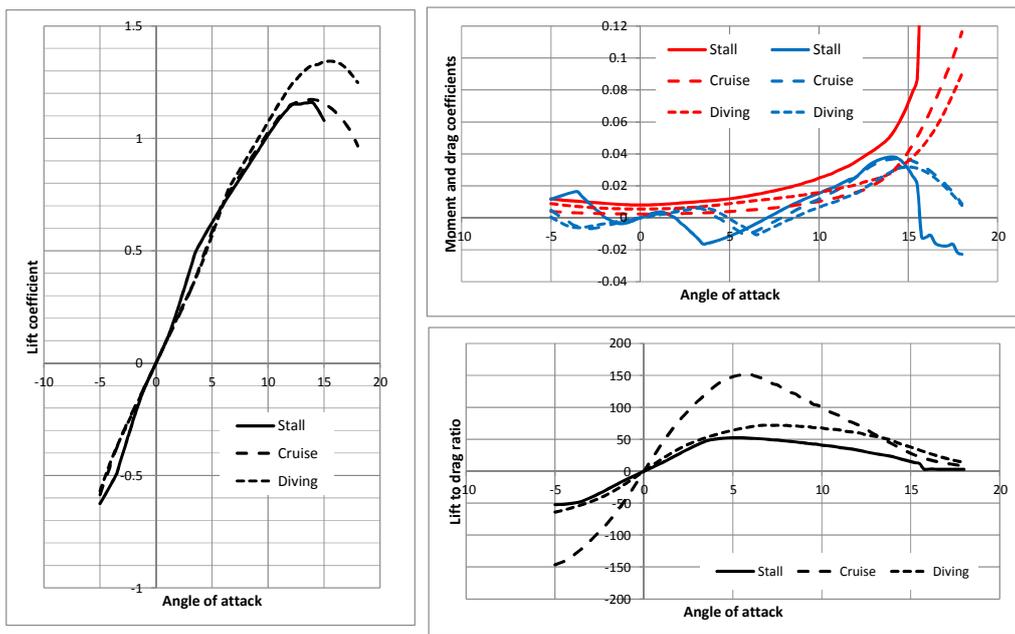


Figure A.4: Variation of coefficients for NACA 0012 air-foil.

C Weight analysis

In computing the mass matrix AVL requires the input of the mass distribution of the components, sub-components and other accessories as well as payload in

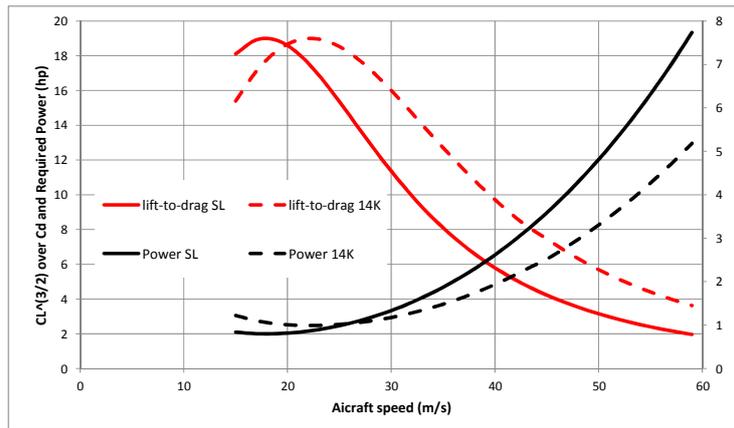


Figure A.5: Lift-Drag ratio versus airspeed for GULMA UAV.

terms of the their product of inertia. Therefore all components were weighed to build up the excel mas sheets. This is shown in Figure A.6.

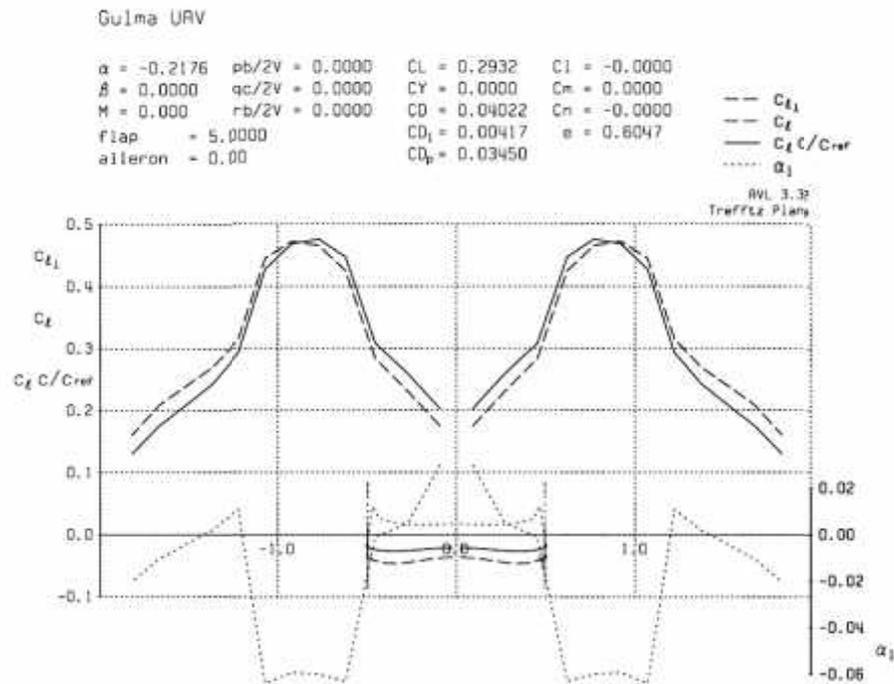
Components	Mass (kg)	x	y	z
Structural Components				
Fuselage +Wiring + Painting	5.194	0.944	0.000	0.08640
Right Wing + Wiring + Painting	4.875	1.091	-0.950	0.17700
Left Wing + Wiring + Painting	4.875	1.091	0.950	0.17660
Left Aileron + Painting	0.000	0.000	0.000	0.00000
Right Aileron + Painting	0.000	0.000	0.000	0.00000
MLG strut + tyres	2.242	1.147	0.000	-0.20000
NLG (Tyre+Strut)	0.593	0.224	0.000	-0.09084
Tail boom(left)	0.343	1.663	0.507	0.18064
Tail boom(right)	0.343	1.663	-0.507	0.18064
Right Fin + wiring + Painting	0.596	2.165	-0.515	0.26700
Left Fin + wiring + Painting	0.596	2.165	0.515	0.26700
Right rudder + Painting	0.000	0.000	0.000	0.00000
Left rudder + Painting	0.000	0.000	0.000	0.00000
Horizontal Tailplane + Painting	1.313	2.192	0.000	0.47800
Elevator + Painting	0.000	0.000	0.000	0.00000
SYSTEM AND ACCESSORIES				
Engine and Engine accessories	6.612	1.584	0	0.134
Propeller +hub	0.000	0	0	0
Alternator	0.000	0	0	0
Fuel Tank + Piping	15.133	0.975	0.023	0.0636
AVIONICS SYSTEMS AND ASSESSORIES				
Battery	1.81	0.345	0	0.035637
Surveillance Camera	1.347	0.489	0	-0.0603
Forward Looking Camera	0.013	0.160	0	-0.05
Flight Control Systems	0.600	0.660	0	0.028054
Avionics	0.14	0.66	0	0.028054
Rear Camera	0.013	2.2	0	0.45
Telemetry Equipment	0.40	0.66	0	0.028054
Ballast	0	0.20	0.00	0.00
OEM	47.042	Summation of OEM Mass Moments		

Figure A.6: Mass distribution sheet for GULMA UAV.

D Trim results from AVL

The trimmed results are presented in graphical form for some selected cases. This is because the volume of simulations required to for various trim conditions. However the selected the results for the trim procedure are in Figure A.7.

Case 2: Trim condition, cruise state, flaps 5 deg, Wto mass and CG position



E Eigenmode Analysis

Loading for Case 2

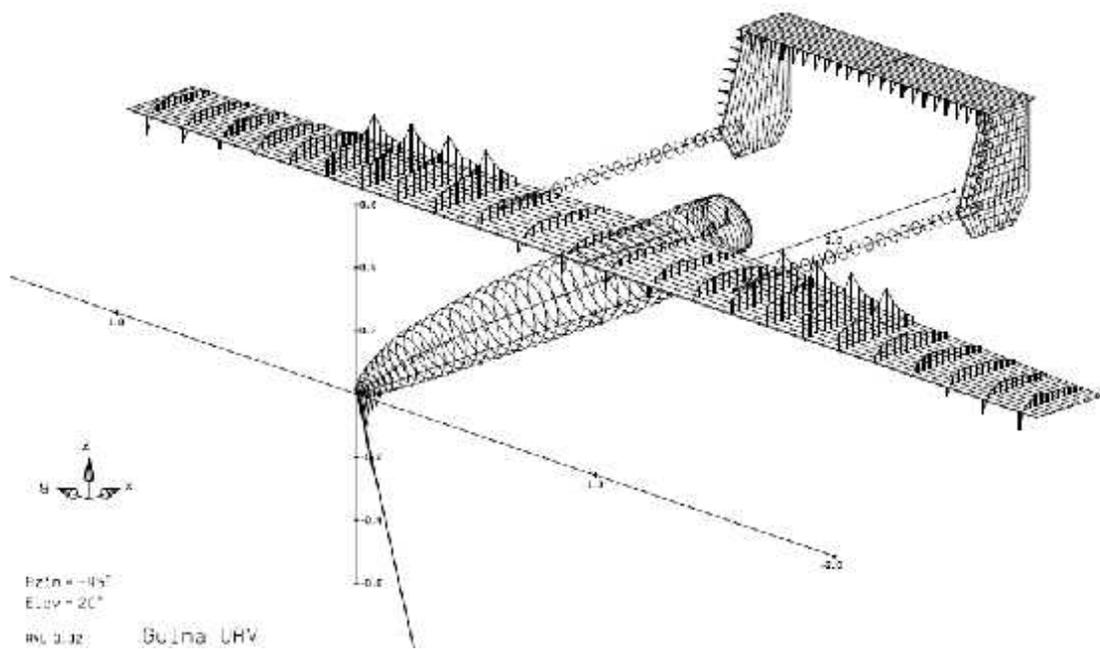
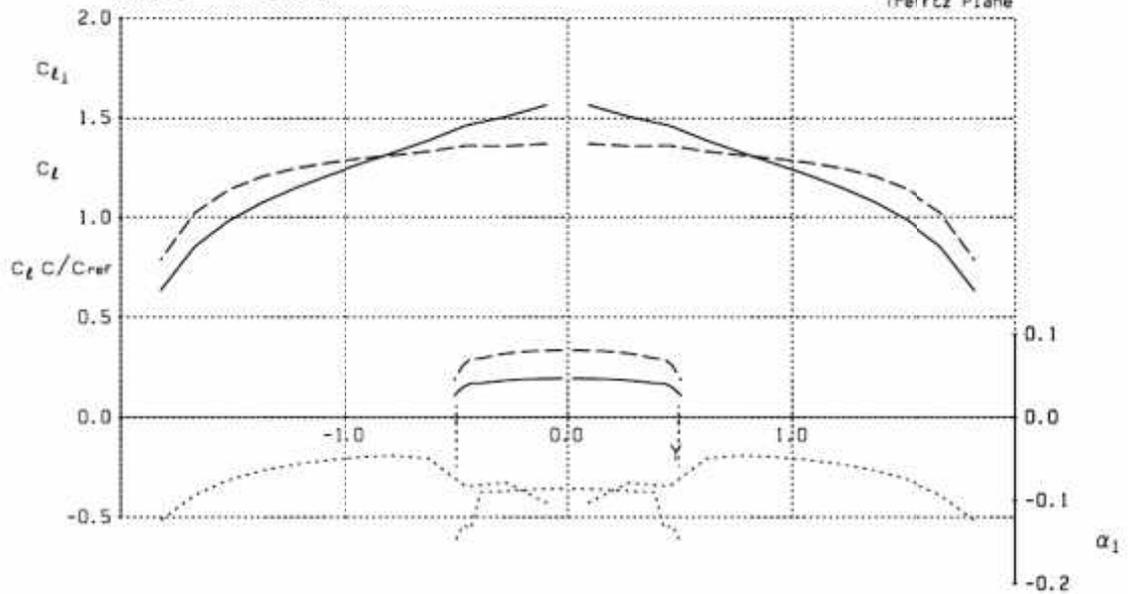


Figure A.7: Trim for loading case 2.

Gulma UAV

$\alpha = 9.9477$	$pb/2V = 0.0000$	$CL = 1.3000$	$Cl = -0.0000$
$\beta = 0.0000$	$qc/2V = 0.0000$	$CY = 0.0000$	$Cm = -0.0000$
$M = 0.000$	$rb/2V = 0.0000$	$CD = 0.07117$	$Cn = -0.0000$
flap = 0.0000		$CD_1 = 0.04718$	$e = 1.0206$
aileron = 0.0000		$CD_p = 0.03450$	
Elevator = -2.5749			
Rudder = 0.0000			

--- C_{L1}
 --- C_L
 --- $C_L C / C_{ref}$
 α_1
 AVL 3.32
 Trefftz Plane



Loading Case 4

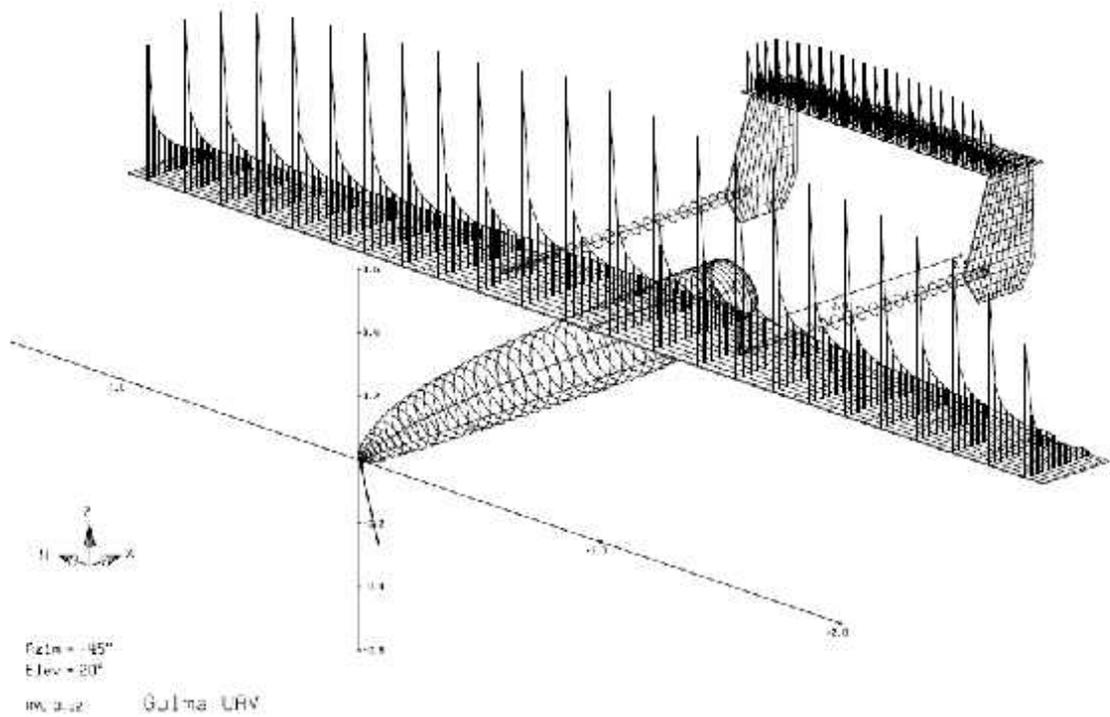


Figure A.8: Trim for loading case 4.

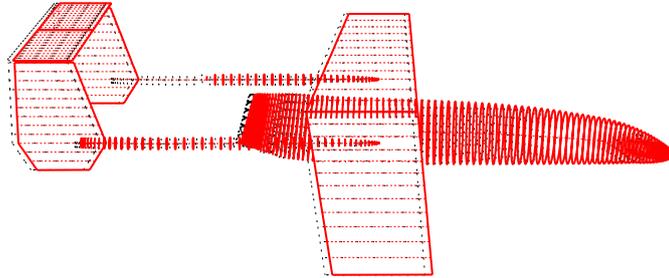


Figure A.9: Eigenmode view of model in flight level condition.

Mode 2
 $f = 0.0000$ cycles/Tunit
 $\zeta = 1.000000$
 $t = -0.37$ Tunit
 $\omega t = 0^\circ$
 $x = 0\text{m}$
 $y = 0\text{m}$
 $z = 0\text{m}$
 $\phi = -10.1^\circ$
 $\theta = 0.0^\circ$
 $\psi = -7.0^\circ$
 $V = 1\text{m/Tunit}$
 $\alpha = 6.21^\circ$
 $\beta = 4.75^\circ$

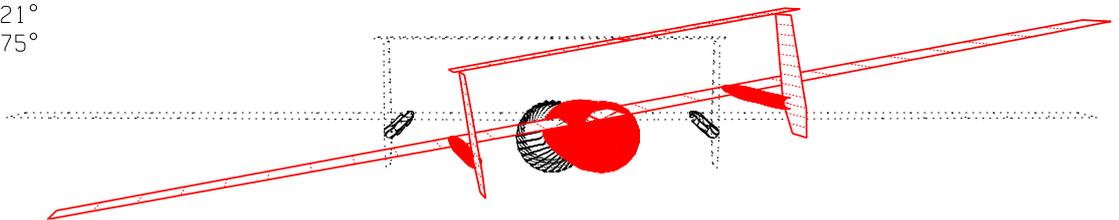


Figure A.10: Eigenmode side slip motion diagram.

 stability-axis derivatives...

	alpha		beta							
z' force CL	CLa =	6.586709	CLb =	-0.000001						
y' force CY	CYa =	0.000000	CYb =	-0.414475						
x' mom. Cl'	CLa =	-0.000001	CLb =	-0.038718						
y' mom. Cm	Cma =	-0.025620	Cmb =	0.000001						
z' mom. Cn'	Cna =	0.000000	Cnb =	0.072175						
	roll rate p'			pitch rate q'			yaw rate r'			
z' force CL	CLp =	0.000002	CLq =	8.324544	CLr =	0.000000				
y' force CY	Cyp =	0.017197	CYq =	0.000002	CYr =	0.166292				
x' mom. Cl'	CLp =	-0.658516	CLq =	0.000000	CLr =	0.090591				
y' mom. Cm	Cmp =	0.000001	Cmq =	-14.261755	Cmr =	-0.000001				
z' mom. Cn'	Cnp =	-0.021642	Cnq =	-0.000001	Cnr =	-0.060726				
	flap	d1	aileron	d2	Elevator	d3	Rudder	d4		
z' force CL	CLd1 =	0.025709	CLd2 =	0.000000	CLd3 =	0.007774	CLd4 =	-0.000817		
y' force CY	CYd1 =	0.000000	CYd2 =	0.000190	CYd3 =	0.000000	CYd4 =	0.000000		
x' mom. Cl'	CLd1 =	0.000000	CLd2 =	-0.006667	CLd3 =	0.000000	CLd4 =	0.000000		
y' mom. Cm	Cmd1 =	0.001358	Cmd2 =	0.000000	Cmd3 =	-0.022940	Cmd4 =	0.002381		
z' mom. Cn'	Cnd1 =	0.000000	Cnd2 =	-0.000107	Cnd3 =	0.000000	Cnd4 =	0.000000		
Trefftz drag	CDffd1 =	0.000464	CDffd2 =	0.000000	CDffd3 =	0.000094	CDffd4 =	-0.000072		
span eff.	ed1 =	-0.007992	ed2 =	0.000000	ed3 =	0.015182	ed4 =	0.022242		

Neutral point xnp = 1.091346

clb cnr / clr cnb = 0.359597 (> 1 if spirally stable)

Figure A.11: AVL stability derivatives results for case 2.

 stability-axis derivatives...

	alpha		beta							
z' force CL	CLa =	6.282118	CLb =	-0.000000						
y' force CY	CYa =	0.000000	CYb =	-0.469090						
x' mom. Cl'	CLa =	0.006545	CLb =	-0.205364						
y' mom. Cm	Cma =	-1.606691	Cmb =	-0.000000						
z' mom. Cn'	Cna =	-0.000381	Cnb =	0.059964						
	roll rate p'			pitch rate q'			yaw rate r'			
z' force CL	CLp =	-0.013171	CLq =	7.463471	CLr =	0.005418				
y' force CY	Cyp =	-0.109184	CYq =	0.000000	CYr =	0.264625				
x' mom. Cl'	CLp =	-0.660855	CLq =	0.007776	CLr =	0.321342				
y' mom. Cm	Cmp =	0.003348	Cmq =	-15.351664	Cmr =	-0.000844				
z' mom. Cn'	Cnp =	-0.108914	Cnq =	-0.000341	Cnr =	-0.075525				
	flap	d1	aileron	d2	Elevator	d3	Rudder	d4		
z' force CL	CLd1 =	0.024835	CLd2 =	-0.000000	CLd3 =	0.007535	CLd4 =	-0.000776		
y' force CY	CYd1 =	-0.000000	CYd2 =	-0.002268	CYd3 =	0.000000	CYd4 =	0.000000		
x' mom. Cl'	CLd1 =	0.000026	CLd2 =	-0.006569	CLd3 =	0.000008	CLd4 =	-0.000001		
y' mom. Cm	Cmd1 =	-0.001868	Cmd2 =	-0.000000	Cmd3 =	-0.023172	Cmd4 =	0.002432		
z' mom. Cn'	Cnd1 =	0.000002	Cnd2 =	-0.000454	Cnd3 =	0.000000	Cnd4 =	-0.000000		
Trefftz drag	CDffd1 =	0.001646	CDffd2 =	-0.000000	CDffd3 =	0.000804	CDffd4 =	-0.000105		
span eff.	ed1 =	0.003916	ed2 =	0.000000	ed3 =	-0.005188	ed4 =	0.001004		

Neutral point xnp = 1.155650

clb cnr / clr cnb = 0.804932 (> 1 if spirally stable)

Figure A.12: AVL stability derivatives results for case 4.

Appendix B

UAV Models

A Linearized models

The non-linear model described in Chapter 3 of is linearized by assuming small perturbations from a steady, level trim condition. For simplicity, state variables is the same notation as notation as perturbation of states in the linear model. The longitudinal dynamics are decoupled from the lateral-directional axis and the thrust is assumed to be constant. The longitudinal model are described in Equations 3.1 and 3.2 of Chapter 3. The resulting system for the longitudinal system is:

$$A_{lon} = \begin{bmatrix} X_u & X_w & X_q - W_e & -g\cos\theta_e \\ Z_u & Z_w & Z_q + U_e & -g\sin\theta_e \\ M_u & M_w & M_q & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} B_{lon} = \begin{bmatrix} X_{\delta_e} \\ Z_{\delta_e} \\ M_{\delta_e} \\ 0 \end{bmatrix} \quad (\text{B.1})$$

where the terms W_e , U_e and θ_e are the trim conditions for linearization. The X, Z and M terms with are the dimensional aerodynamic derivatives computed from AVL. A_{lon} contains the longitudinal static derivatives and B_{lon} matrix has the control derivatives that are also computed from AVL. The linearized acceleration measurements are also computed from the non-linear model auxiliary sub-system

given as:

$$a_{xx} = \dot{u} + qW_e + g \cos \theta_e \theta + g \sin \theta_e \quad (\text{B.2})$$

$$a_{zz} = \dot{w} - qU_e + g \sin \theta_e \theta - g \cos \theta_e \quad (\text{B.3})$$

The Lateral directional axis dynamics are described in Equations 3.2 and 3.3. The Y, L and N are described by the linear functions of lateral velocity, roll and yaw angular rates, as well as the ailerons and rudder deflections. The resulting system is given by the following:

$$A_{latt} = \begin{bmatrix} Y_v & Y_q + W_e & Y_r - U_e & -g \cos \theta_e & 0 \\ L_v & L_p & -L_r I_x / I_{xz} & 0 & 0 \\ N_v & -N_p I_z / I_{xz} & N_r & 0 & 0 \\ 0 & 1 & \tan \theta_e & 0 & 0 \\ 0 & 0 & \sec \theta_e & 0 & 0 \end{bmatrix} B_{lon} = \begin{bmatrix} Y_{\delta_a} & Y_{\delta_r} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (\text{B.4})$$

The linearized lateral acceleration measurement is given as:

$$a_{yy} = \dot{v} + pW_e + rU_e - g \cos \theta_e \phi \quad (\text{B.5})$$

The MATLAB/Simulink diagram for the models used are shown in below:

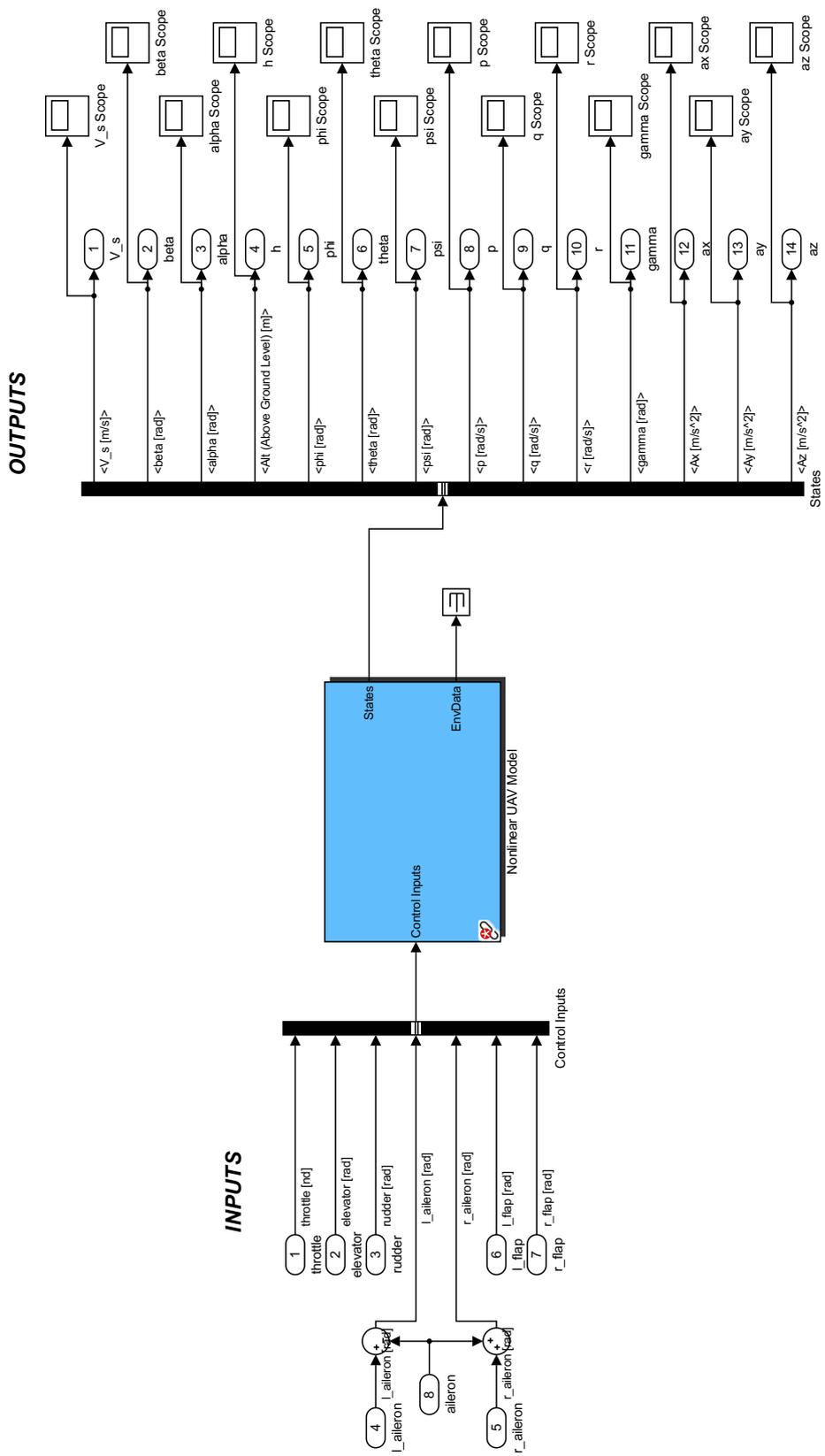


Figure B.1: UAV Non-linear Simulink Test Platform [56].

**UMN UAV
SOFTWARE-IN-THE-LOOP SIMULATION
(SIL)**

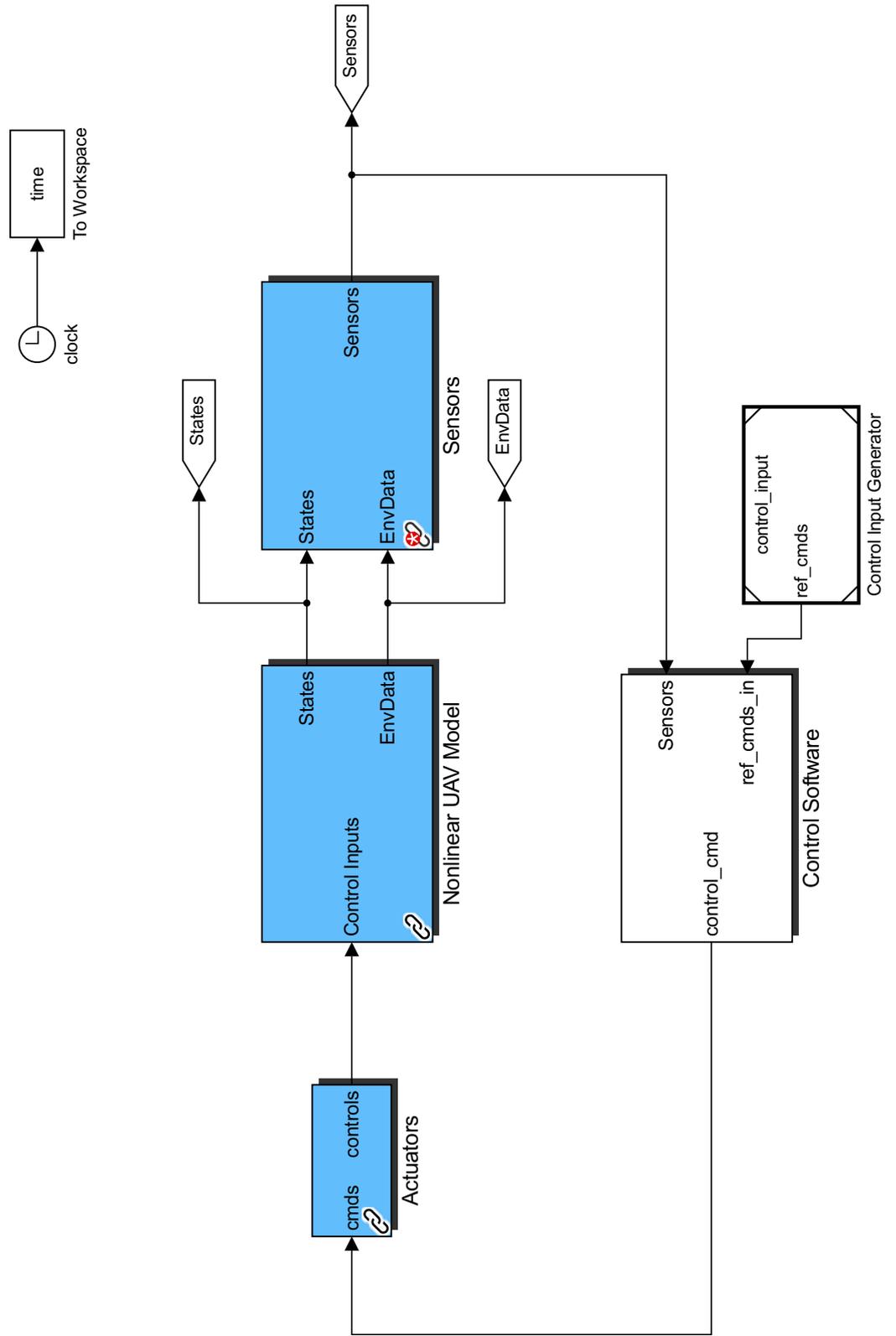


Figure B.2: UAV Non-linear Simulink Test Platform [56].

Bibliography

- [1] Godwin Abbe. *NAF GULMA UAV: Design, Manufacture and Technical Specifications*. Alex Badeh Centre for Innovation and Technology: Unpublished manuscript, 2014.
- [2] Reg Austin. *Unmanned aircraft systems: UAVS design, development and deployment*, volume 54. John Wiley & Sons, 2011.
- [3] Gary J Balas, John C Doyle, Keith Glover, Andy Packard, and Roy Smith. μ -analysis and synthesis toolbox. *MATLAB*, 1993.
- [4] Péter Bauer, Paw Yew Chai, Luigi Iannelli, Rohit Pandita, Gergely Regula, Bálint Vanek, Gary J Balas, Luigi Glielmo, and József Bokor. Uav lab open research platform for unmanned aerial vehicles. In *Advances in Aerospace Guidance, Navigation and Control*, pages 175–186. Springer, 2011.
- [5] Randal W Beard, Derek Kingston, Morgan Quigley, Deryl Snyder, Reed Christiansen, Walt Johnson, Timothy McLain, and Michael Goodrich. Autonomous vehicle technologies for small fixed-wing uavs. *Journal of Aerospace Computing, Information and Communication*, 2(1):92–108, 2005.
- [6] Randal W Beard and Timothy W McLain. *Small unmanned aircraft: Theory and practice*. Princeton University Press, 2012.
- [7] N Berman and U Shaked. H-infinity nonlinear filtering. *International Journal of Robust and Nonlinear Control*, 6(4):281–296, 1996.

- [8] John T Betts. Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control and Dynamics*, 21(2):193–207, 1998.
- [9] David N Borys and Richard Colgren. Advances in intelligent autopilot systems for unmanned aerial vehicles. AIAA, 2005.
- [10] Stephen P Boyd and Craig H Barratt. *Linear controller design: Limits of performance*. Prentice-Hall Englewood Cliffs, NJ, 1991.
- [11] Gregg Buskey, Gordon Wyeth, and Jonathan Roberts. Autonomous helicopter hover using an artificial neural network. volume 2, pages 1635–1640. IEEE, 2001.
- [12] Hongda Chen, Kuochu Chang, and Craig S Agate. Uav path planning with tangent-plus-lyapunov vector field guidance and obstacle avoidance. *IEEE Transactions on Aerospace and Electronic Systems*, 49(2):840–856, 2013.
- [13] Teng D Chollom, Nkemdilim Ofodile, and Osichinaka Ubadike. Application techniques of multi-objective particle swarm optimization: Aircraft flight control. In *UKACC 11th International Conference on Control*, pages 1–6. IEEE, 2016.
- [14] Reece A Clothier, Jennifer L Palmer, Rodney A Walker, and Neale L Fulton. Definition of an airworthiness certification framework for civil unmanned aircraft systems. *Safety Science Journal*, 49(6):871–885, 2011.
- [15] Carlos Coello, Gregorio Toscano Pulido, and M Salazar Lechuga. Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8(3):256–279, 2004.
- [16] Michael V Cook. *Flight dynamics principles: A linear systems approach to aircraft stability and control*. Butterworth-Heinemann, 2012.

- [17] C Cuerno-Rejado and R Martinez-Val. Unmanned aircraft systems in the civil airworthiness regulatory frame: A case study. *Journal of Aircraft*, 48(4):1351–1359, 2011.
- [18] Navid Dadkhah and Bérénice Mettler. Survey of motion planning literature in the presence of uncertainty: Considerations for uav guidance. *Journal of Intelligent & Robotic Systems*, 65(1-4):233–246, 2012.
- [19] Konstantinos Dalamagkidis, Kimon P Valavanis, and Les A Piegl. On unmanned aircraft systems issues, challenges and operational restrictions preventing integration into the national airspace system. *Progress in Aerospace Sciences*, 44(7):503–519, 2008.
- [20] Satchidananda Dehuri, Ashish Ghosh, and Sung-Bae Cho. Particle swarm optimised polynomial neural network for classification: A multi-objective view. *International Journal of Intelligent Defence Support Systems*, 1(3):225–253, 2008.
- [21] Joaquín Derrac, Salvador García, Daniel Molina, and Francisco Herrera. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Journal of Swarm and Evolutionary Computation*, 1(1):3–18, 2011.
- [22] Marco Dorigo, Mauro Birattari, and Thomas Stutzle. Ant colony optimization. *IEEE computational intelligence magazine*, 1(4):28–39, 2006.
- [23] Andrei Dorobantu. *Test Platforms for Model-Based Flight Research*. PhD thesis, 2013.
- [24] Andrei Dorobantu, Wayne Johnson, F Adhika Lie, Brian Taylor, Austin Murch, Yew Chai Paw, Demoz Gebre-Egziabher, and Gary Balas. An airborne experimental test platform: From theory to flight. pages 659–673. IEEE, ACC, 2013.

- [25] Andrei Dorobantu, Austin Murch, B er enice Mettler, and Gary Balas. System identification for small, low-cost, fixed-wing unmanned aircraft. *Journal of Aircraft*, 50(4):1117–1130, 2013.
- [26] M Drela and H Youngren. Athena vortex lattice user manual, version 3.26, 2006.
- [27] Gabriel Hugh Elkaim, Fidelis Adhika Pradipta Lie, and Demoz Gebre-Egziabher. Principles of guidance, navigation, and control of uavs. In *Handbook of Unmanned Aerial Vehicles*, pages 347–380. Springer, 2015.
- [28] Andries P Engelbrecht. *Fundamentals of computational swarm intelligence*. John Wiley & Sons, 2006.
- [29] Thor I Fossen. *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.
- [30] Vladislav Gavrillets. Dynamic model for a miniature aerobatic helicopter. In *Handbook of Unmanned Aerial Vehicles*, pages 279–306. Springer, 2015.
- [31] Ashish Ghosh and Satchidananda Dehuri. Evolutionary algorithms for multi-criteria optimization: A survey. *International Journal of Computing and Information Sciences*, 2005.
- [32] Da-Wei Gu, Petko Hr Petkov, and Mihail M Konstantinov. *Robust control design with MATLAB®*, volume 1. Springer Science & Business Media, 2005.
- [33] Chen Guimin, Jia Jianyuan, and Han Qi. Study on the strategy of decreasing inertia weight in particle swarm optimization algorithm. *Journal-Xi'an Jiaotong University*, 40(1):53, 2006.

- [34] J Hamidi. Control system design using particle swarm optimization (ps0). *International Journal of Soft Computing and Engineering*, 1(6):116–119, 2012.
- [35] Larry Hardesty. Optimizing optimization algorithms. {<https://news.mit.edu/2015/optimizing-optimization-algorithms-0121>}, 2015. Online; accessed 2015-01-21.
- [36] Kaveh Hassani and Won-Sook Lee. Optimal tuning of linear quadratic regulators using quantum particle swarm optimization. pages 1–8. IEEE, 2014.
- [37] Sonia Hernandez and Maruthi R Akella. Lyapunov-based guidance for orbit transfers and rendezvous in levi-civita coordinates. *Journal of Guidance, Control, and Dynamics*, 37(4):1170–1181, 2014.
- [38] Leong Hou In, Rylan Jager, Shahriar Keshmiri, and Richard Colgren. Development of a pilot training platform for uavs using a 6dof nonlinear model with flight test validation. *AIAA*, 2008.
- [39] Jonathan How, Yoshiaki Kuwata, and Ellis King. Flight demonstrations of cooperative control for uav teams. In *AIAA 3rd" Unmanned Unlimited" Technical Conference, Workshop and Exhibition*, page 6490. AIAA, 2004.
- [40] Sunday Iliya, Ferrante Neri, Dylan Menzies, Pip Cornelius, and Lorenzo Picinali. Differential evolution schemes for speech segmentation: A comparative study. In *IEEE Symposium on Differential Evolution*, pages 1–8. IEEE, 2014.
- [41] B Kada and Y Ghazzawi. Robust pid controller design for an uav flight control system. volume 2, pages 19–21. WCECS, 2011.

- [42] Farid Kendoul. Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems. *Journal of Field Robotics*, 29(2):315–378, 2012.
- [43] Farid Kendoul, Zhenyu Yu, and Kenzo Nonami. Guidance and nonlinear control system for autonomous flight of minirotorcraft unmanned aerial vehicles. *Journal of Field Robotics*, 27(3):311–334, 2010.
- [44] James Kennedy. Swarm intelligence. In *Handbook of nature-inspired and innovative computing*, pages 187–219. Springer Science & Business Media, 2006.
- [45] H Jin Kim, Mingu Kim, Hyon Lim, Chulwoo Park, Seungho Yoon, Daewon Lee, Hyunjin Choi, Gyeongtaek Oh, Jongho Park, and Youdan Kim. Fully autonomous vision-based net-recovery landing system for a fixed-wing uav. *IEEE/ASME Transactions On Mechatronics*, 18(4):1320–1333, 2013.
- [46] Joshua D Knowles and David W Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary computation*, 8(2):149–172, 2000.
- [47] Jean N Koster, Alec Velazco, Mikhail Kosyan, Scott Balaban, and Lydia McDowell. The hyperion 2 green aircraft project. 878:2012, 2012.
- [48] Aymeric Kron, Jean de Lafontaine, and Daniel Alazard. Robust 2dof h-infinity controller for highly flexible aircraft: Design methodology and numerical results. *Canadian Aeronautics and Space Journal*, 49(1):19–29, 2003.
- [49] Martin E Kügler and Florian Holzapfel. Parameterization and computation of automatic take-off and landing trajectories for fixed-wing uav. page 3421. AIAA, 2017.

- [50] Jang-Ho Lee, Byoung-Mun Min, and Eung-Tai Kim. Autopilot design of tilt-rotor uav using particle swarm optimization method. pages 1629–1633. IEEE, IEEE, 2007.
- [51] Norman Lehtomaki, Nils R Sandell Jr, Michael Athans, et al. Robustness results in linear-quadratic gaussian based multivariable control designs. *IEEE Transactions on Automatic Control*, 26(1):75–93, 1981.
- [52] Jinxing Li, Berta Esteban-Fernández de Ávila, Wei Gao, Liangfang Zhang, and Joseph Wang. Micro/nanorobots for biomedicine: Delivery, surgery, sensing, and detoxification. *Journal of Science Robotics*, 2(4):eaam6431, 2017.
- [53] Yong Li, Jianchang Liu, and Yu Wang. Design approach of weighting matrices for lqr based on multi-objective evolution algorithm. In *International Conference on Information and Automation*, pages 1188–1192. IEEE, 2008.
- [54] Brian London. *A model-based systems engineering framework for concept development*. PhD thesis, 2012.
- [55] Bernal Luis and Dorman Paul. Study of the aerodynamics of a small uav using avl software. 2006.
- [56] MATLAB. *version 7.10.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts, 2010.
- [57] Michalis Mavrovouniotis, Changhe Li, and Shengxiang Yang. A survey of swarm intelligence for dynamic optimization: Algorithms and applications. *Journal of Swarm and Evolutionary Computation*, 33:1–17, 2017.
- [58] Pat H McIngvale. Unmanned aerial vehicle automatic landing system, 1998. US Patent 5,716,032.

- [59] Tomas Melin. A vortex lattice matlab implementation for linear aerodynamic wing applications. *Master's Thesis, Department of Aeronautics, Royal Institute of Technology (KTH), Stockholm, Sweden*, 2000.
- [60] Seyedali Mirjalili, Pradeep Jangir, and Shahrzad Saremi. Multi-objective ant-lion optimizer: A multi-objective optimization algorithm for solving engineering problems. *Journal of Applied Intelligence*, 46(1):79–95, 2017.
- [61] Austin M Murch. *Flight Control System Architecture for the NASA AirSTAR Flight Test Infrastructure*. AIAA, 2008.
- [62] Austin M Murch, Yew Chai Paw, Rohit Pandita, Zhefeng Li, and Gary J Balas. A low cost small uav flight research facility. In *Advances in Aerospace Guidance, Navigation and Control*, pages 29–40. Springer, 2011.
- [63] Kenzo Nonami. Prospect and recent research & development for civil use autonomous unmanned aircraft as uav and mav. *Journal of System, Design and Dynamics*, 1(2):120–128, 2007.
- [64] Sidhartha Panda and Narayana Prasad Padhy. Comparison of particle swarm optimization and genetic algorithm for facts-based controller design. *Journal of Applied Soft computing*, 8(4):1418–1427, 2008.
- [65] Yew Chai Paw. *Synthesis and validation of flight control for UAV*. PhD thesis, University of Minnesota, 2009.
- [66] Jiankun Peng, Hongwen He, and Rui Xiong. Rule based energy management strategy for a series-parallel plug-in hybrid electric bus optimized by dynamic programming. *Applied Energy*, 185:1633–1643, 2017.
- [67] Mauro Pontani and Brace A Conway. Particle swarm optimization applied to space trajectories. *AIAA Guidance, Control and Dynamics*, 33(5):1429–1441, 2010.

- [68] Pratyusha Rakshit, Amit Konar, and Swagatam Das. Noisy evolutionary optimization algorithms: A comprehensive survey. *Journal of Swarm and Evolutionary Computation*, 33:18–45, 2017.
- [69] David Royer, Shahriar Keshmiri, Benjamin Sweeten, and Valerie Jones. Modeling and sensitivity analysis of the meridian unmanned aircraft. *AIAA Infotech Aerospace*, 2010.
- [70] Nasser R Sabar, Jemal Abawajy, and John Yearwood. Heterogeneous cooperative co-evolution memetic differential evolution algorithm for big data optimization problems. *IEEE Transactions on Evolutionary Computation*, 21(2):315–327, 2017.
- [71] Amir Saeed, Syed Ussama Ali, and M Zamurad Shah. Linear control techniques application and comparison for a research uav altitude control. pages 126–133. IEEE, 2016.
- [72] Edgar N Sanchez, Hector M Becerra, and Carlos M Velez. Combining fuzzy, pid and regulation control for an autonomous mini-helicopter. *Journal of Information Sciences*, 177(10):1999–2022, 2007.
- [73] Ruud Schoonderwoerd, Owen E Holland, Janet L Bruten, and Leon JM Rothkrantz. Ant-based load balancing in telecommunications networks. *Journal of Adaptive Behavior*, 5(2):169–207, 1997.
- [74] Neryahu A Shneydor. *Missile guidance and pursuit: Kinematics, Dynamics and Control*. Elsevier, 1998.
- [75] Shashiprakash Singh. *Autonomous landing of unmanned aerial vehicles*. PhD thesis, Indian Institute of Science, 2010.
- [76] Sigurd Skogestad and Ian Postlethwaite. *Multivariable feedback control: Analysis and design*, volume 2. Wiley, New York, 2007.

- [77] Cary R Spitzer, Daniel A Martinec, Cornelius T Leondes, Vyacheslav Tuzlukov, Won-Sik Yoon, and Yong Deak Kim. Avionics systems. *Journal of Systems, Controls, Embedded Systems, Energy and Machines*, 2016.
- [78] Nidamarthi Srinivas and Kalyanmoy Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, 2(3):221–248, 1994.
- [79] Thomas Stastny, Adyasha Dash, and Roland Siegwart. Nonlinear mpc for fixed-wing uav trajectory tracking: Implementation and flight experiments. AIAA, 2017.
- [80] Brian L Stevens and Frank L Lewis. *Aircraft control and simulation*. John Wiley & Sons, 2003.
- [81] WH Tang, Shan He, Emmanuel Prempain, QH Wu, and J Fitch. A particle swarm optimiser with passive congregation approach to thermal modelling for power transformers. In *IEEE Congress on Evolutionary Computation*, volume 3, pages 2745–2751. IEEE, 2005.
- [82] Ashish Tewari. *Automatic Control of Atmospheric and Space Flight Vehicles: Design and Analysis with MATLAB® and Simulink®*. Springer Science & Business Media, 2011.
- [83] Frans Van den Bergh and Andries Petrus Engelbrecht. A study of particle swarm optimization particle trajectories. *Journal of Information sciences*, 176(8):937–971, 2006.
- [84] Dyke Weatherington and U Deputy. Unmanned aircraft systems roadmap. 2005.
- [85] Takeshi Yamasaki, Hirotohi Sakaida, Keisuke Enomoto, Hiroyuki Takano, and Yoriaki Baba. Robust trajectory-tracking method for uav guidance using proportional navigation. pages 1404–1409. IEEE, 2007.

- [86] Lotfi A Zadeh. Making computers think like people: Fuzzy set theory. *IEEE Spectrum*, 21(8):26–32, 1984.
- [87] Majid Zamani, Nasser Sadati, and Masoud Karimi Ghartemani. Design of an h-inifinty pid controller using particle swarm optimization. *International Journal of Control, Automation and Systems*, 7(2):273–280, 2009.