

POLITECNICO DI TORINO

Master of Science in Mechatronic Engineering

Master Degree Thesis

Modelling and Control of Unmanned Aerial Vehicles



Supervisor: Prof. António Pedro Aguiar
(University of Porto)

Co-supervisor: Prof. Alessandro Rizzo
(Politecnico di Torino)

Candidate: Anuj Regmi
Matricola: s238251

Abstract

Unmanned Aerial Vehicle (UAV) is becoming an important tool for military and civilian operations due to its growing applications. Examples of such applications are delivery, sensing, mapping, surveillance and search and rescue, where the manned mission is hard.

The mathematical modelling of the UAV starts with the characterization of its kinematics, dynamics and forces and moments, which generates the 12 nonlinear ordinary differential equations. The aerodynamic flow in the design considers laminar flow in the operating velocity range. Linearization is performed around the equilibrium point which is also called the trim condition. The trim condition depends on the flight scenario. Linearization generates the state space matrix, where the modelling is carried out with the decoupling of the longitudinal and lateral mode separately. We then proceed with the autopilot design using Linear Quadratic Regulator (LQR) techniques with integral action for both the longitudinal and lateral modes. The core part of the longitudinal is the roll controller and lateral is the heading controller. Next, we present the guidance for the path following that uses the vector field method with constant k_{path} for the waypoint and straight line following.

Simulation results show the behaviour of the dynamic model. The LQR control with the integral action behaves as expected, which holds the height during the path following at constant altitude. The actuator signal is also shown based on the different values of the path constant. Simulink and MATLAB are the main software tools that were used to illustrate the performance of the resulting closed loop system for different control parameters. This model implementation with the full UAV nonlinear dynamics brings higher accuracy to the performance evaluation and it is suitable to test different controllers and guidance algorithms.

Acknowledgements

I'm deeply indebted to Prof. A. Pedro Aguiar for his continuous support and hospitality. He has always been the energy for my success. I would also like to extend my deepest gratitude to Prof. Alessandro Rizzo for his trust without which I could not have done anything. I owe my special thanks to Paulo Lopes, Sara Costa, Paulo Cunha, Diyako Ghaderiyan, Matheus Reis, Yassine Boukili, Praveen Jain, Margarida and Ahmed for all their support in the C2SR lab.

I want to thank my family who have always been the inspiration. My thanks to Marcelo and Khalid, who have always been a great team to work with during all those long hours of assignment. The warmest thank to my best friend, Maria Vittoria Morelli, knowing her was the most beautiful part, who is always beside me during the hardest time, the hours she spent helping me and with whom I share some of the best moment.

I'm incredibly grateful to Tholgapian, Mehmet, Elisabete and all my friends who have given me the love of family.

Contents

| | |
|-------------------------------------------------------|-----------|
| List of Figures | 5 |
| List of Tables | 7 |
| 1 Introduction | 8 |
| 1.1 Motivation | 8 |
| 1.2 Literature | 8 |
| 1.3 Objective and Contribution | 9 |
| 1.4 Organization of the dissertation | 10 |
| 2 Mathematical Modelling | 11 |
| 2.1 Coordinate Frames | 11 |
| 2.1.1 Coordinate Frame Transformation | 11 |
| 2.2 Aircraft frame | 12 |
| 2.2.1 Inertial Frame (\mathbf{F}^i) | 12 |
| 2.2.2 Vehicle Frame (\mathbf{F}^V) | 12 |
| 2.2.3 Vehicle Frame-1 (\mathbf{F}^{V1}) | 13 |
| 2.2.4 Vehicle Frame-2 (\mathbf{F}^{V2}) | 13 |
| 2.2.5 Body Frame (\mathbf{F}^b) | 14 |
| 2.2.6 Stability Frame (\mathbf{F}^s) | 15 |
| 2.2.7 Wind Frame (\mathbf{F}^w) | 15 |
| 2.3 Airspeed, Wind speed and Ground speed | 16 |
| 2.4 Kinematic model of Controlled Flight | 18 |
| 2.5 No wind case | 19 |
| 2.6 Kinematics | 19 |
| 2.6.1 Translation Kinematics | 19 |
| 2.6.2 Rotational Kinematics | 20 |
| 2.7 Dynamics | 21 |
| 2.7.1 Translational Dynamics | 21 |
| 2.7.2 Rotational Dynamics | 22 |
| 2.8 Summary of the Dynamics and Kinematics | 23 |
| 2.9 Forces and Moments | 23 |
| 2.9.1 Gravity Force | 23 |
| 2.9.2 Aerodynamic Force | 24 |
| 2.9.3 Propeller Thrust | 27 |
| 2.10 Summary | 28 |
| 2.10.1 Force | 28 |
| 2.10.2 Moments | 28 |
| 2.10.3 Nonlinear Equations | 29 |
| 2.11 Linearization | 29 |

| | | |
|----------|-----------------------------------|-----------|
| 2.11.1 | Equilibrium state | 29 |
| 2.11.2 | Decoupling | 30 |
| 2.11.3 | State Space Modelling | 30 |
| 2.11.4 | Longitudinal Dynamics | 31 |
| 2.11.5 | Lateral Dynamics | 32 |
| 3 | Guidance and control | 34 |
| 3.1 | LQR controller | 34 |
| 3.2 | Autopilot | 36 |
| 3.2.1 | Longitudinal Controller | 36 |
| 3.2.2 | Lateral Controller | 38 |
| 3.3 | Path Manager | 40 |
| 3.3.1 | Waypoint | 40 |
| 3.3.2 | Straight line following | 42 |
| 3.3.3 | Orbit following | 44 |
| 4 | Simulation results | 46 |
| 4.1 | Aircraft Model | 46 |
| 4.1.1 | Physical Parameter | 47 |
| 4.1.2 | Aerodynamic coefficient | 48 |
| 4.2 | State Space Matrix | 48 |
| 4.3 | Path following | 49 |
| 5 | Conclusion | 53 |
| A | Appendix | 55 |
| | Bibliography | 57 |

List of Figures

| | | |
|------|---------------------------------------------------------------------------------|----|
| 1.1 | Vayu Medical Drone, source:[18] | 8 |
| 1.2 | Zipline Drone, source:[20] | 8 |
| 2.1 | Vector Rotation about the k -axis | 11 |
| 2.2 | Vehicle Frame, F^v to Vehicle Frame, F^{v1} , source:[2] | 13 |
| 2.3 | Vehicle Frame, F^{v1} to vehicle frame2, F^{v2} , source:[2] | 13 |
| 2.4 | Vehicle Frame, F^{v2} to Body frame, F^b , source:[2] | 14 |
| 2.5 | Body Frame, F^b to stability Frame, F^s , source:[2] | 14 |
| 2.6 | Stability frame, F^s to Wind frame F^w , source:[2] | 15 |
| 2.7 | Wind Triangle source:[2] | 18 |
| 2.8 | Flight path angle, source:[2] | 18 |
| 2.9 | Block diagram of Linearization steps | 31 |
| 3.1 | Block diagram LQR | 34 |
| 3.2 | Height(m) vs time(seconds) | 37 |
| 3.3 | Velocity(V_a)(m/s) vs time | 37 |
| 3.4 | Pitch angle (θ)(in degree) based on height tracking vs time | 37 |
| 3.5 | Height tracking error vs time | 37 |
| 3.6 | Elevator deflection(δ_e)(in degree) vs time | 37 |
| 3.7 | Throttle movement(δ_t) vs time | 37 |
| 3.8 | Velocity(m/s) vs time | 38 |
| 3.9 | pitch angle(θ) vs time | 38 |
| 3.10 | Elevator deflection(δ_e)(in degree) vs time | 38 |
| 3.11 | Throttle movement(δ_t) vs time | 38 |
| 3.12 | Course angle(χ)(in degree) vs time | 39 |
| 3.13 | Heading angle error(e_ψ) based on height tracking vs time | 39 |
| 3.14 | Aileron deflection(δ_a)(in degree) vs time | 39 |
| 3.15 | Rudder deflection(δ_r)(in degree) | 39 |
| 3.16 | Architecture of the Control | 40 |
| 3.17 | Half plane approach | 41 |
| 3.18 | Straight line path | 42 |
| 3.19 | Circular path with radius ρ | 44 |
| 4.1 | Trim block in Simulink | 46 |
| 4.2 | Zagi 60 HP electric model airframe, source:[19] | 47 |
| 4.3 | Waypoint tracking with different values of k_{path} | 49 |
| 4.4 | Orbit tracking with different values of k_{orbit} | 50 |
| 4.5 | Velocity(m/s) vs time(s) for the value of orbit constant $k_{orbit}=2$ | 50 |
| 4.6 | Altitude(m) vs time(s) for the value of orbit constant $k_{orbit}=2$ | 50 |
| 4.7 | Throttle(δ_t) vs time(s) for the value of orbit constant $k_{orbit}=2$ | 50 |

| | | |
|------|-----------------------------------------------------------------------------------------------------------------|----|
| 4.8 | Elevator deflection(δ_e)(in degree) vs time(s) for the value of orbit constant $k_{orbit}=2$ | 50 |
| 4.9 | Rudder deflection(δ_r) (in degree) vs time(s) for value of orbit constant $k_{orbit}=2$ | 51 |
| 4.10 | Aileron deflection(δ_a)(in degree) vs time(s) for the value of orbit constant $k_{orbit}=2$ | 51 |
| 4.11 | Waypoint following in the inertial frame with $k_{path}=0.1$ | 51 |
| 4.12 | Altitude(m) vs time(s) to follow the given waypoint, $k_{path}=0.1$ | 52 |
| 4.13 | Velocity(m/s) vs time(s) to follow the given waypoint $k_{path}=0.1$. . . | 52 |
| 4.14 | Course angle(χ) (in degree) vs time(s) to follow the given waypoint, $k_{path}=0.1$ | 52 |
| 4.15 | Pitch angle(θ) (in degree) vs time(s) to follow the given waypoint, $k_{path}=0.1$ | 52 |

List of Tables

| | | |
|-----|---------------------------------------------|----|
| 4.1 | Geometry of the Aircraft | 47 |
| 4.2 | Moment of Inertia Data | 47 |
| 4.3 | Aerodynamic coefficient data | 48 |
| A.1 | Longitudinal coefficient equation | 56 |
| A.2 | Lateral coefficient equation | 56 |

Chapter 1

Introduction

1.1 Motivation

In the past few years, there has been a significant development in the advancement of Unmanned Aerial Vehicle(UAV) technology. Something that started with the military application, didn't take much time to reach the civilian interest. To the places where manned operation is hard to achieve a UAV is used to perform. In the recent years with the emerging of high power density batteries, long range and low power micro radio devices, cheap airframes, and powerful microprocessors and motors, UAV technology has become applicable in civilian circumstances, especially for small UAVs because they are expendable, easy to build and operate.

The use of UAVs in medical field is one of the most interesting application like shown in the figure 1.1 and figure 1.2. At present, there are many UAVs that transfer medicine and blood from different remote places to the city and vice-versa. The companies like Vayu and Zipline are revolutionizing the industry. The another application is the surveillance of the wildlife which is also very interesting.

1.2 Literature

Small UAVs have a relatively short wingspan and light weight[7]. They can be operated by only one to two people [4]. Many can even be hand-carried and hand-launched. In fact, small UAVs are designed to fly at low altitude (normally less than 1000 feet) to provide a close observation of the ground objects. This low flight altitude may make the UAVs easy to crash [5]. A robust and accurate autopilot system is indispensable for small UAVs to successfully perform the task. Autopilots



Figure 1.1: Vayu Medical Drone, source:[18]



Figure 1.2: Zipline Drone, source:[20]

are systems to guide the UAVs in flight with no assistance from human operators. Autopilots were firstly developed for missiles and later extended to aircrafts and ships since 1910's [16]. A minimal autopilot system include attitude sensors and on-board processor. Due to the high nonlinearities of the air plane dynamics, a lot of intelligent control techniques have been used in autopilot systems to guarantee a smooth desirable trajectory navigation. Nowadays, technological advances in wireless networks and micro electro-mechanical systems (MEMS) make it possible to use the inexpensive micro autopilots on small UAVs [5].

Generally, the idea of the UAVs are considered as autonomous but mostly they are semi autonomous. The mission and corresponding desired paths are defined by the operator and during the flight the operator always monitors the behaviour of the aircraft. UAVs are generally developed based on the task they need to perform. For the surveillance, one would want them to follow given waypoint and maintain loiter in the region as we would be scanning the area. To control these vehicles, one approach is radio control that requires human pilots to control the UAV through radio signals. Another method is the use of autopilots, where the autopilot can automatically keep the airplane on the desired state. The most interesting is the mixed control where the inner loop is the autopilot and the outer is controlled by the pilot. In general the autopilot controls the velocity and altitude and the pilot controls the waypoint and path.

Trajectory tracking and path following are among the basic motion control tasks that autonomous aircraft are required to execute, in order to perform some desired application. In trajectory tracking, the vehicle follows a given trajectory with time constraints, and thus, trajectory tracking controllers require the vehicle to track a time-parametrized trajectory by commanding its forward speed and orientation [1]. In classical path following, the vehicle is required to converge to and follow a given desired geometric path. Several approaches have been proposed for UAV trajectory tracking. An approach for tight tracking of curved trajectories is presented in [14]. Rather than pursuing the trajectory tracking approach, this research explores path following where the objective is to be on the path rather than at a certain point at a particular time. There are different algorithms for following the path, some of them can be summarized as the nonlinear guidance law(NLGL) [14], vector field approach [12] and many other methods. Vector field path following technique accurately follows the path than other techniques but consumes large control effort compared to the NLGL method [15]. The vectors in the field are directed toward the path to be followed and in the desired direction of flight. The vectors also serve as heading commands to the Micro Aerial Vehicle(MAV). The method is currently applicable to paths composed of straight lines and arcs [12].

1.3 Objective and Contribution

The main goal of this thesis is to develop the mathematical model, analyze and design a control system for UAV guidance. The mathematical model is developed and tested in the simulink, and for which the autopilot is also designed using the LQR method. The path following algorithm is also implemented and evaluated.

1.4 Organization of the dissertation

The whole of the thesis is divided into three main chapters:

- *Chapter 1* deals with the general introduction of the UAV and the motivation to carry the project. In this chapter we can see the different literature and concept that has been used till date.
- *Chapter 2* is the mathematical modelling, where we present the derivation of the UAV model starting from the reference frame to the forces. In this chapter we develop the corresponding nonlinear equations and also describe the linearization procedure to obtain the state space model.
- Once the state space model is generated we will proceed with the *Chapter 3*, where we will develop the autopilot using the LQR method with the integral action and also the path following algorithm.
- *Chapter 4* present the simulation results that illustrate the time evolution of the relevant signals of the plant and the controller using the simulink. We will see the behaviour of the UAV by changing different control parameters.
- *Chapter 5* is the Conclusion, which deals with the summary of the thesis along with the future perspective of the thesis.

Chapter 2

Mathematical Modelling

2.1 Coordinate Frames

The knowledge of the coordinate frame is the first step to introduce before formulating the dynamic and kinematic equation of the UAV. Different parameter works in different frames as [6]:

- Newton's equations of motion are derived relative to a fixed, inertial reference frame. However, motion is most easily described in a body-fixed frame.
- Aerodynamic forces and torques act on the aircraft body and are most easily described in a body-fixed reference frame.
- On-board sensors like accelerometers and rate gyros measure information with respect to the body frame. Alternatively, GPS measures position, ground speed, and course angle with respect to the inertial frame.
- Most mission requirements, like loiter points and flight trajectories, are specified in the inertial frame. In addition, map information is also given in an inertial frame.

2.1.1 Coordinate Frame Transformation

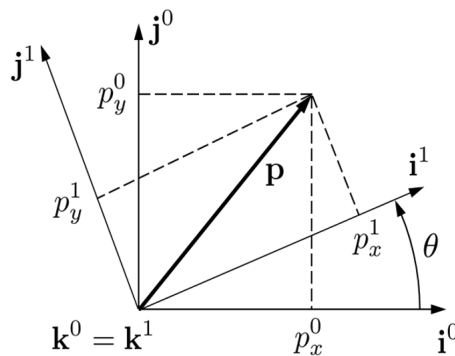


Figure 2.1: Vector Rotation about the k -axis

Any vector whose value is known in one frame can be multiplied with the rotation matrix to obtain the value in the other frame, when the origin is at the same point [3]. For example, let us consider a vector as shown in the figure 2.1 P^0 denoted in

the reference frame \mathbf{R}^0 . The value of the vector in the reference frame \mathbf{R}^1 is given by \mathbf{P}^1 with the rotation matrix \mathbf{R}_0^1 thus,

$$\mathbf{P}^1 = \mathbf{R}_0^1 \mathbf{P}^0$$

The notation \mathbf{R}_0^1 is used to denote a rotation from coordinate frame \mathbf{F}^0 to coordinate frame \mathbf{F}^1 . For the rotation angle, θ and any unit vector we would have the following different condition based on the right handed reference frame rule:

The rotation around the \mathbf{X} axis with right hand rule is given by;

$$R_0^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix}$$

The rotation around the \mathbf{Y} axis with right hand rule is given by;

$$R_0^1 = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix}$$

The rotation around the \mathbf{Z} axis with right hand rule is given by;

$$R_0^1 = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The rotation matrix \mathbf{R}_0^1 has the following properties:

$$\begin{aligned} (R_a^b)^{-1} &= (R_a^b)^T = R_b^a \\ R_b^c R_a^b &= R_a^c \\ \det(R_a^b) &= 1 \end{aligned}$$

2.2 Aircraft frame

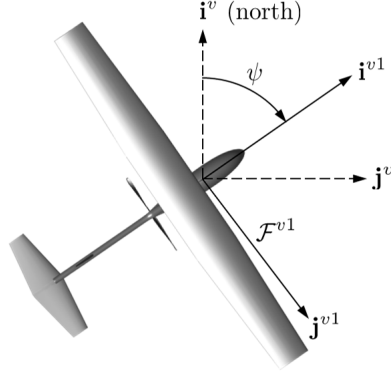
As, mentioned in the earlier section there are different frame associated with the micro aerial vehicle. All the frames are well described in the reference [2]. and are as follows:

2.2.1 Inertial Frame (\mathbf{F}^i)

This frame is also called Earth Fixed reference frame. It is a point in the surface of the earth with the \mathbf{X} axis pointing towards the north and the \mathbf{Y} axis pointing towards the east and the \mathbf{Z} axis pointing down. It is also called North East Down reference frame.

2.2.2 Vehicle Frame (\mathbf{F}^V)

The orientation of the axis in this reference frame is same as the earth fixed reference frame as shown in the figure 2.2, the origin point is acting in the center of the mass of the vehicle.


 Figure 2.2: Vehicle Frame, F^v to Vehicle Frame, F^{v1} , source:[2]

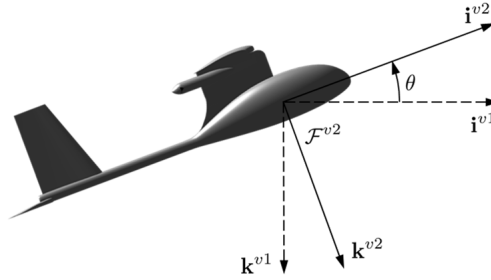
2.2.3 Vehicle Frame-1 (F^{V1})

This frame is generated by the rotation of the vehicle frame F^V with angle ψ , along the Z -axis. The angle ψ is called heading angle. The rotation matrix is given by:

$$R_v^{v1}(\psi) = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

Any vector p^{v1} , would be given by the following equation, where p^v is the vector in the vehicle frame (F^v).

$$p^{v1} = R_v^{v1}(\psi)p^v$$


 Figure 2.3: Vehicle Frame, F^{v1} to vehicle frame2, F^{v2} , source:[2]

2.2.4 Vehicle Frame-2 (F^{V2})

This frame is generated by the rotation of the vehicle frame F^{V1} with angle θ , along the Y -axis as shown in the figure 2.3. The angle θ is called pitch angle. The rotation matrix is given by:

$$R_{v1}^{v2}(\theta) = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2.2)$$

Any vector \mathbf{p}^{v2} , would be given by the following equation, where \mathbf{p}^{v1} is the vector in the vehicle frame-1 (\mathcal{F}^{v1}).

$$\mathbf{p}^{v2} = R_{v1}^{v2}(\theta)\mathbf{p}^{v1}$$

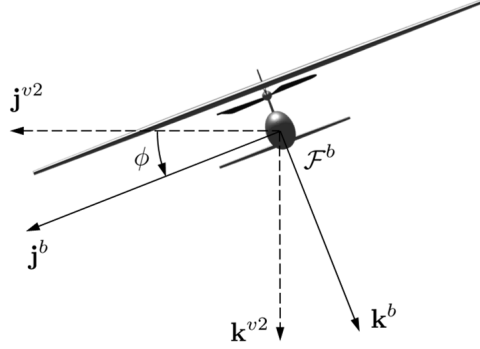


Figure 2.4: Vehicle Frame \mathcal{F}^{v2} to Body frame, \mathcal{F}^b , source:[2]

2.2.5 Body Frame (\mathcal{F}^b)

This frame is generated by the rotation of the aircraft from the vehicle frame-2 (\mathcal{F}^{v2}) along the \mathbf{X} -axis, with the angle ϕ as shown in the figure 2.4. This angle is also called roll angle or bank angle. The rotation matrix is given by:

$$\mathbf{R}_{v2}^b(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \quad (2.3)$$

Any vector \mathbf{p}^b , would be given by the following equation, where \mathbf{p}^{v2} is the vector in the vehicle frame-2 (\mathcal{F}^{v2}).

$$\mathbf{p}^b = R_{v2}^b(\phi)\mathbf{p}^{v2}$$

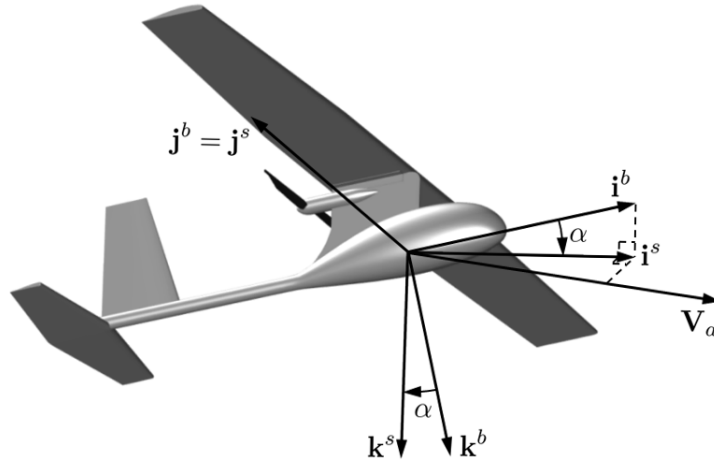


Figure 2.5: Body Frame \mathcal{F}^b to stability Frame, \mathcal{F}^s , source:[2]

2.2.6 Stability Frame (F^s)

Aerodynamic forces are generated as the airframe moves through the air surrounding it. We refer to the velocity of the aircraft relative to the surrounding air as the airspeed vector, denoted \mathbf{V}_a . The magnitude of the airspeed vector is simply referred to as the airspeed (V_a).

To generate lift, the wings of the airframe must fly at a positive angle with respect to the airspeed vector. This angle is called the angle of attack and is denoted by α . The angle of attack is defined as a left-handed rotation about j^b and is such that it aligns with the projection of V_a onto the plane spanned by i^b and k^b as shown in the figure 2.5. The need for a left handed rotation is caused by the definition of positive angle of attack, which is positive for a right-handed rotation from the stability frame is axis to the body frame i^b axis. The rotation matrix is given by

$$R_b^s(\alpha) = \begin{bmatrix} \cos \alpha & 0 & \sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \end{bmatrix}$$

Stability frame helps us rigorously define angle of attack and is useful for analyzing stability of aircraft.

Any vector represented in the body frame can be represented in the stability frame with the help of the rotation matrix R_b^s .

$$\mathbf{p}^s = R_b^s(\alpha) \mathbf{p}^b$$

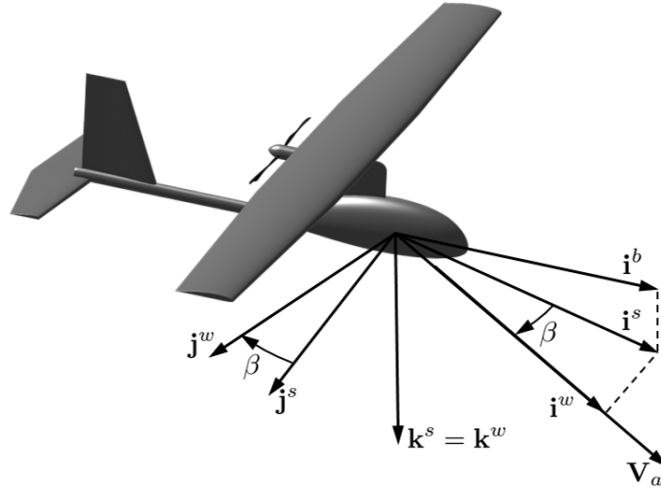


Figure 2.6: Stability frame, F^s to Wind frame F^w , source:[2]

2.2.7 Wind Frame (F^w)

The angle between the airspeed vector and the i^b - k^b plane is called the side-slip angle and is denoted by β as shown in the figure 2.6. The wind frame is obtained by rotating the stability frame by a right-handed rotation of β about k^s . The unit vector i^w is aligned with the airspeed vector V_a .

The rotation matrix is given by:

$$R_s^w(\beta) = \begin{bmatrix} \cos \beta & \sin \beta & 0 \\ -\sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Wind frame helps us rigorously define side-slip angle. Side-slip angle, is nominally zero for tailed aircraft.

The rotation from the stability frame to the wind frame is given by:

$$\mathbf{p}^w = R_s^w(\beta)\mathbf{p}^s$$

The rotation from the body frame to the wind frame is given by:

$$\begin{aligned} R_b^w(\alpha, \beta) &= R_s^w(\beta)R_b^s(\alpha) \\ &= \begin{bmatrix} \cos \beta & \sin \beta & 0 \\ -\sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \alpha & 0 & \sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \end{bmatrix} \\ &= \begin{bmatrix} \cos \beta \cos \alpha & \sin \beta & \cos \beta \sin \alpha \\ -\sin \beta \cos \alpha & \cos \beta & -\sin \beta \sin \alpha \\ -\sin \alpha & 0 & \cos \alpha \end{bmatrix} \end{aligned}$$

The rotation from the wind frame to the body frame is given by the rotation matrix:

$$R_w^b(\alpha, \beta) = (R_b^w)^T(\alpha, \beta) = \begin{bmatrix} \cos \beta \cos \alpha & -\sin \beta \cos \alpha & -\sin \alpha \\ \sin \beta & \cos \beta & 0 \\ \cos \beta \sin \alpha & -\sin \beta \sin \alpha & \cos \alpha \end{bmatrix} \quad (2.4)$$

Which is the transpose of the *equation – 1.1*.

2.3 Airspeed, Wind speed and Ground speed

When developing the dynamic equations of motion for a UAV, it is important to remember that the inertial forces experienced by the UAV are dependent on velocities and accelerations relative to a fixed (inertial) reference frame.

The aerodynamic forces, however, depend on the velocity of the airframe relative to the surrounding air. “When wind is not present, these velocities are the same.” However, wind is almost always present with UAV’s and we must carefully distinguish between airspeed.

\mathbf{V}_a is the velocity vector of the UAV represented in the wind frame and has three components $(V_a \ 0 \ 0)^T$, where V_a is the magnitude of the aircraft velocity along the \mathbf{X} -direction.

The ground speed vector (\mathbf{V}_g) represented with respect to the inertial frame. The wind velocity relative to the inertial frame is given by \mathbf{V}_w .

The ground speed vector \mathbf{V}_g^b can be represented in the body frame as: $(u \ v \ w)^T$.

Where, u v and w are the body frame velocity components. Now, \mathbf{V}_g^b is the velocity

of the UAV with respect to the inertial frame but (expressed)represented in the body frame.

Body frame component of the airspeed vector are $\mathbf{V}_a^b = (u_r \ v_r \ w_r)^T$ can be written in the body frame as:

$$\mathbf{V}_a = \mathbf{V}_g - \mathbf{V}_w \quad (2.5)$$

$$\begin{aligned} \mathbf{V}_g^b &= \begin{pmatrix} u \\ v \\ w \end{pmatrix} \\ \mathbf{V}_w^b &= \begin{pmatrix} u_w \\ v_w \\ w_w \end{pmatrix} \\ &= \mathcal{R}_v^b(\phi, \theta, \psi) \begin{pmatrix} w_n \\ w_e \\ w_d \end{pmatrix} \end{aligned}$$

R_v^b is obtained by multiplying equation, 2.1,2.2 and 2.3
The airspeed vector in wind frame is given by:

$$\mathbf{V}_a^w = \begin{pmatrix} V_a \\ 0 \\ 0 \end{pmatrix}$$

Now, we would represent the airspeed from the wind frame to the body frame by using the above equation along with the equation 2.4.

$$\mathbf{V}_a^b = \begin{pmatrix} u_r \\ v_r \\ w_r \end{pmatrix} = \begin{pmatrix} u - u_w \\ v - v_w \\ w - w_w \end{pmatrix}$$

$$\begin{aligned} \mathbf{V}_a^b &= \begin{pmatrix} u_r \\ v_r \\ w_r \end{pmatrix} \\ &= \mathcal{R}_w^b \begin{pmatrix} V_a \\ 0 \\ 0 \end{pmatrix} \end{aligned}$$

R_w^b is from equation 2.4

$$= \begin{pmatrix} \cos \beta \cos \alpha & -\sin \beta \cos \alpha & -\sin \alpha \\ \sin \beta & \cos \beta & -\sin \beta \sin \alpha \\ \cos \beta \sin \alpha & 0 & \cos \alpha \end{pmatrix} \begin{pmatrix} V_a \\ 0 \\ 0 \end{pmatrix}$$

The airspeed vector body frame component can be represented by the airspeed magnitude V_a , angle of attack (α), and the side-slip angle (β) as:

$$\begin{pmatrix} u_r \\ v_r \\ w_r \end{pmatrix} = V_a \begin{pmatrix} \cos \alpha \cos \beta \\ \sin \beta \\ \sin \alpha \cos \beta \end{pmatrix} \quad (2.6)$$

The expressions that are commonly used in formulation of the equation of motion for a UAV as the aerodynamic force and moments are expressed by:

$$V_a = \sqrt{u_r^2 + v_r^2 + w_r^2} \quad (2.7)$$

$$\alpha = \tan^{-1} \left(\frac{w_r}{u_r} \right) \quad (2.8)$$

$$\beta = \sin^{-1} \left(\frac{v_r}{\sqrt{u_r^2 + v_r^2 + w_r^2}} \right) \quad (2.9)$$

2.4 Kinematic model of Controlled Flight

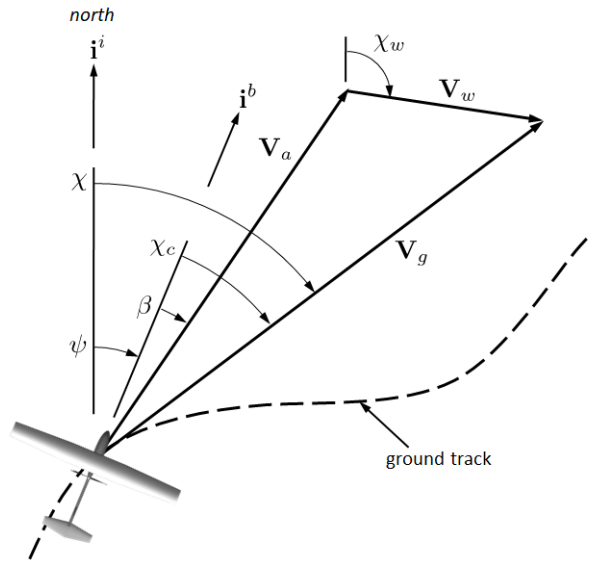


Figure 2.7: Wind Triangle source:[2]

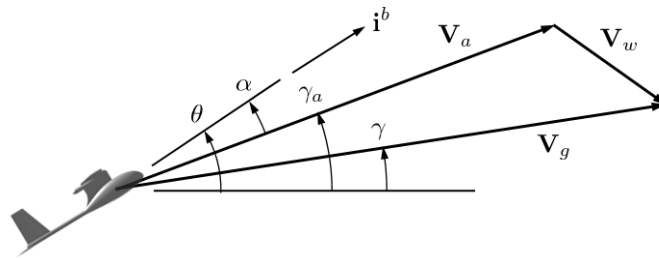


Figure 2.8: Flight path angle, source:[2]
, source:

The ground speed vector(\mathbf{V}_g) can be represented in the inertial frame by rotating it to the inertial frame as shown in the figure 2.7 and figure 2.8.

$$\mathbf{V}_g^i = \begin{pmatrix} \cos \chi & -\sin \chi & 0 \\ \sin \chi & \cos \chi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \gamma & 0 & \sin \gamma \\ 0 & 1 & 0 \\ -\sin \gamma & 0 & \cos \gamma \end{pmatrix} \begin{pmatrix} V_g \\ 0 \\ 0 \end{pmatrix} \quad (2.10)$$

$$\mathbf{V}_g^i = V_g \begin{pmatrix} \cos \chi \cos \gamma \\ \sin \chi \cos \gamma \\ -\sin \gamma \end{pmatrix}$$

which gives:

$$\begin{pmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{h} \end{pmatrix} = V_g \begin{pmatrix} \cos \chi \cos \gamma \\ \sin \chi \cos \gamma \\ \sin \gamma \end{pmatrix} \quad (2.11)$$

Similarly, the airspeed vector(\mathbf{V}_a) can be represented in the inertial frame by rotating it along from the body frame to the inertial frame as shown in the figure 2.7 and figure 2.8:

$$\mathbf{V}_a^i = \begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \gamma_a & 0 & \sin \gamma_a \\ 0 & 1 & 0 \\ -\sin \gamma_a & 0 & \cos \gamma_a \end{pmatrix} \begin{pmatrix} V_a \\ 0 \\ 0 \end{pmatrix} \quad (2.12)$$

Alternatively, using the wind triangle equation we have from equation 2.5.

$$\begin{pmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{h} \end{pmatrix} = V_a \begin{pmatrix} \cos \psi \cos \gamma_a \\ \sin \psi \cos \gamma_a \\ \sin \gamma_a \end{pmatrix} + \begin{pmatrix} w_n \\ w_e \\ -w_d \end{pmatrix}$$

2.5 No wind case

In the absence of wind we will have following simplifications:

$$\begin{aligned} V_a &= V_g & \text{Airspeed equals groundspeed} \\ \psi &= \chi & \text{Heading equals course} \\ \gamma &= \gamma_a & \text{Flight path angle equals air-mass-referenced flight path angle} \\ u &= u_r & \text{Velocity equals velocity relative to the air mass} \\ v &= v_r \\ w &= w_r \end{aligned}$$

2.6 Kinematics

2.6.1 Translation Kinematics

The translational velocity of the Micro Aerial Vehicle(MAV) is commonly expressed in terms of the velocity components along each of the axes in a body-fixed coordinate frame. The translational position of the MAV is usually measured and expressed in an inertial reference frame. Thus, we would have the rotation matrix from the body frame to the inertial frame. As in our case the vehicle frame is identical with the inertial frame, thus, we would rotate the body frame to the vehicle frame by the rotation matrix R_b^v , which is obtained by multiplying equation [2.1,2.2 and 2.3] [6]. The rate of change of the position in the north, east and down is the velocity of the aircraft in the inertial frame, and is given as[2]:

$$\begin{pmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{p}_d \end{pmatrix} \triangleq \frac{d}{dt} \begin{pmatrix} p_n \\ p_e \\ p_d \end{pmatrix} = \mathbf{v}^v = R_b^v \mathbf{v}^b = R_b^v \begin{pmatrix} u \\ v \\ w \end{pmatrix} \quad (2.13)$$

$$\begin{pmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{p}_d \end{pmatrix} = \begin{pmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} \quad (2.14)$$

Where, c_θ is $\cos \theta$ and s_θ is $\sin \theta$ and similarly for other trigonometric values, Here, $(\dot{p}_n \ \dot{p}_e \ \dot{p}_d)$ are the linear velocity in the inertial frame but the angles $(\phi \ \theta \ \psi)$ are in different Frame. The velocity $u \ v \ w$ are in the body frame.

2.6.2 Rotational Kinematics

The relationship between angular positions ϕ , θ , and ψ and the angular rates p , q , and r is also complicated by the fact that these quantities are defined in different coordinate frames.

The angular rates are defined in the body frame F^b . The angular positions (Euler angles) are defined in three different coordinate frames: the roll angle ϕ is a rotation from F^{v2} to F^b about the $i^{v2} = i^b$ axis; the pitch angle θ is a rotation from F^{v1} to F^{v2} about the $j^{v1} = j^{v2}$ axis; and the yaw angle ψ is a rotation from F^v to F^{v1} about the $k^v = k^{v1}$ axis. The body-frame angular rates can be expressed in terms of the derivatives of the Euler angles, considering a proper rotation;

$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix} + R_{v2}^b(\phi) \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + R_{v2}^b(\phi) R_{v1}^{v2}(\theta) \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix}$$

The p, q, r is the angular velocity that would be measured by the gyros fixed in the aircraft, with their axis in the body frame.

$$\begin{aligned} &= \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{pmatrix} \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{pmatrix} \begin{pmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{pmatrix} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \end{aligned}$$

Inverting the above equation, we would have:

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \quad (2.15)$$

The angular rate of the Euler angles are given by the above equation based on the inertial frame/Vehicle Frame.

2.6.2.1 Total Kinematics Equation

From 2.13 and 2.15 we will have Six of the 12 state equations for the UAV relating to position and velocity:

$$\begin{pmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{p}_d \end{pmatrix} = \begin{pmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix}$$

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix}$$

The remaining six equations will come from applying Newton's 2nd law to the translational and rotational motion of the aircraft.

2.7 Dynamics

2.7.1 Translational Dynamics

Newton's 2nd Law acts on the inertial frame and is given by:

$$m \frac{d\mathbf{V}_g}{dt_i} = \mathbf{f}$$

- \mathbf{f} is the sum of all external forces
- m is the mass of the aircraft
- Time derivative taken wrt inertial frame
- \mathbf{V}_g is the velocity with respect to the inertial frame.

Since, all our values are in the body frame thus we would write it in the body frame:
Using the expression

$$\frac{d\mathbf{V}_g}{dt_i} = \frac{d\mathbf{V}_g}{dt_b} + \boldsymbol{\omega}_{b/i} \times \mathbf{V}_g \text{ (cross-product)}$$

In the Body Frame we have,

$$\mathbf{V}_g^b = \begin{pmatrix} u \\ v \\ w \end{pmatrix} \quad \boldsymbol{\omega}_{b/i}^b = \begin{pmatrix} p \\ q \\ r \end{pmatrix} \quad \mathbf{f}^b = \begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix}.$$

$$m \left(\frac{d\mathbf{V}_g}{dt_b} + \boldsymbol{\omega}_{b/i} \times \mathbf{V}_g \right) = \mathbf{f}$$

Expressing $m \left(\frac{d\mathbf{V}_g}{dt_b} + \boldsymbol{\omega}_{b/i} \times \mathbf{V}_g \right) = \mathbf{f}^b$ in the body frame gives

$$m \left(\frac{d\mathbf{V}_g^b}{dt_b} + \boldsymbol{\omega}_{b/i}^b \times \mathbf{V}_g^b \right) = \mathbf{f}^b,$$

Since $\frac{d\mathbf{V}_g^b}{dt_b} = \begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix}$ we have that

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = - \begin{pmatrix} p \\ q \\ r \end{pmatrix} \times \begin{pmatrix} u \\ v \\ w \end{pmatrix} + \frac{1}{m} \begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} = \begin{pmatrix} rv - qw \\ pw - ru \\ qu - pv \end{pmatrix} + \frac{1}{m} \begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix}$$

2.7.2 Rotational Dynamics

Newton's 2nd Law states that:

$$\frac{d\mathbf{h}}{dt_i} = \mathbf{m}$$

- \mathbf{h} is the angular momentum vector
- \mathbf{m} is the sum of all external moments
- Time derivative taken wrt inertial frame

Therefore we have

$$\frac{d\mathbf{h}}{dt_i} = \frac{d\mathbf{h}}{dt_b} + \boldsymbol{\omega}_{b/i} \times \mathbf{h} = \mathbf{m}.$$

Expressing in the body frame gives

$$\frac{d\mathbf{h}^b}{dt_b} + \boldsymbol{\omega}_{b/i}^b \times \mathbf{h}^b = \mathbf{m}^b.$$

If the aircraft is symmetric about the $\mathbf{i}^b\text{-}\mathbf{k}^b$ plane, then $J_{xy} = J_{yz} = 0$ and

$$\mathbf{J} = \begin{pmatrix} J_x & 0 & -J_{xz} \\ 0 & J_y & 0 \\ -J_{xz} & 0 & J_z \end{pmatrix}$$

This symmetry assumption helps to simplify the analysis. The inverse of \mathbf{J} becomes

$$\begin{aligned} \mathbf{J}^{-1} &= \frac{\text{adj}(\mathbf{J})}{\det(\mathbf{J})} = \frac{\begin{pmatrix} J_y J_z & 0 & J_y J_{xz} \\ 0 & J_x J_z - J_{xz}^2 & 0 \\ J_{xz} J_y & 0 & J_x J_y \end{pmatrix}}{J_x J_y J_z - J_{xz}^2 J_y} \\ &= \begin{pmatrix} \frac{J_z}{\Gamma} & 0 & \frac{J_{xz}}{\Gamma} \\ 0 & \frac{1}{J_y} & 0 \\ \frac{J_{xz}}{\Gamma} & 0 & \frac{J_x}{\Gamma} \end{pmatrix} \end{aligned}$$

where

$$\Gamma \triangleq J_x J_z - J_{xz}^2$$

$$\begin{aligned} \Gamma_1 &= \frac{J_{xz}(J_x - J_y + J_z)}{\Gamma}, & \Gamma_2 &= \frac{J_z(J_z - J_y) + J_{xz}^2}{\Gamma}, & \Gamma_3 &= \frac{J_z}{\Gamma}, \\ \Gamma_4 &= \frac{J_{xz}}{\Gamma}, & \Gamma_5 &= \frac{J_z - J_x}{J_y}, & \Gamma_6 &= \frac{J_{xz}}{J_y}, \\ \Gamma_7 &= \frac{(J_x - J_y)J_x + J_{xz}^2}{\Gamma}, & \Gamma_8 &= \frac{J_x}{\Gamma}, & \Gamma &= J_x J_z - J_{xz}^2. \end{aligned}$$

Define

$$\mathbf{m}^b \triangleq \begin{pmatrix} l \\ m \\ n \end{pmatrix}$$

Then

$$\dot{\boldsymbol{\omega}}_{b/i}^b = \mathbf{J}^{-1} [-\boldsymbol{\omega}_{b/i}^b \times (\mathbf{J}\boldsymbol{\omega}_{b/i}^b) + \mathbf{m}^b]$$

can be expressed as

$$\begin{aligned}
 \begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} &= \begin{pmatrix} \frac{J_z}{\Gamma} & 0 & \frac{J_{xz}}{\Gamma} \\ 0 & \frac{1}{J_y} & 0 \\ \frac{J_{xz}}{\Gamma} & 0 & \frac{J_x}{\Gamma} \end{pmatrix} \left[\begin{pmatrix} 0 & r & -q \\ -r & 0 & p \\ q & -p & 0 \end{pmatrix} \begin{pmatrix} J_x & 0 & -J_{xz} \\ 0 & J_y & 0 \\ -J_{xz} & 0 & J_z \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} + \begin{pmatrix} l \\ m \\ n \end{pmatrix} \right] \\
 &= \begin{pmatrix} \frac{J_z}{\Gamma} & 0 & \frac{J_{xz}}{\Gamma} \\ 0 & \frac{1}{J_y} & 0 \\ \frac{J_{xz}}{\Gamma} & 0 & \frac{J_x}{\Gamma} \end{pmatrix} \left[\begin{pmatrix} J_{xz}pq + (J_y - J_z)qr \\ J_{xz}(r^2 - p^2) + (J_z - J_x)pr \\ (J_x - J_y)pq - J_{xz}qr \end{pmatrix} + \begin{pmatrix} l \\ m \\ n \end{pmatrix} \right] \\
 &= \begin{pmatrix} \Gamma_1pq - \Gamma_2qr + \Gamma_3l + \Gamma_4n \\ \Gamma_5pr - \Gamma_6(p^2 - r^2) + \frac{1}{J_y}m \\ \Gamma_7pq - \Gamma_1qr + \Gamma_4l + \Gamma_8n \end{pmatrix}
 \end{aligned}$$

where, Γ 's are functions of moments and products of inertia.

2.8 Summary of the Dynamics and Kinematics

The objective of this part is to calculate the force vector \mathbf{f}^b and moment \mathbf{m}^b , in the body frame. The equations of motion are a system of 12 first order ODE's:

$$\begin{aligned}
 \begin{pmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{p}_d \end{pmatrix} &= \begin{pmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} \\
 \begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} &= \begin{pmatrix} rv - qw \\ pw - ru \\ qu - pv \end{pmatrix} + \frac{1}{m} \begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix}, \\
 \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} &= \begin{pmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \\
 \begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} &= \begin{pmatrix} \Gamma_1pq - \Gamma_2qr \\ \Gamma_5pr - \Gamma_6(p^2 - r^2) \\ \Gamma_7pq - \Gamma_1qr \end{pmatrix} + \begin{pmatrix} \Gamma_3l + \Gamma_4n \\ \frac{1}{J_y}m \\ \Gamma_4l + \Gamma_8n \end{pmatrix}
 \end{aligned}$$

2.9 Forces and Moments

The external forces are a combination of gravitational, aerodynamic, and propulsion:

$$\mathbf{f} = \mathbf{f}_g + \mathbf{f}_a + \mathbf{f}_p.$$

The external moments are a combination of aerodynamic, and propulsion:

$$\mathbf{m} = \mathbf{m}_a + \mathbf{m}_p.$$

2.9.1 Gravity Force

The gravity vector expressed in the vehicle frame is

$$\mathbf{f}_g^v = \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix}.$$

Expressed in the body frame we have

$$\begin{aligned}
 \mathbf{f}_g^b &= \mathcal{R}_v^b \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} \\
 &= \begin{pmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ s_\phi s_\theta c_\psi - c_\phi s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & s_\phi c_\theta \\ c_\phi s_\theta c_\psi + s_\phi s_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi & c_\phi c_\theta \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} \\
 &= \begin{pmatrix} -mg \sin \theta \\ mg \cos \theta \sin \phi \\ mg \cos \theta \cos \phi \end{pmatrix}
 \end{aligned}$$

2.9.2 Aerodynamic Force

Aerodynamics is the force generated due to the shape of the body passing through the air. In this case, we will only consider all the forces are generated due to the effect of the wing assuming laminar flow. The general expression for Lift, Drag and Moment acting on the aircraft is given by:

$$\begin{aligned}
 F_{\text{lift}} &= \frac{1}{2} \rho V_a^2 S C_L \\
 F_{\text{drag}} &= \frac{1}{2} \rho V_a^2 S C_D \\
 m &= \frac{1}{2} \rho V_a^2 S c C_m
 \end{aligned}$$

where, C_L, C_D, C_m are non dimensional aerodynamic coefficients along the Z-direction, X-direction, Y-direction respectively, .

To better understand the aerodynamic effect on the aircraft we can decouple the longitudinal and lateral aerodynamic force, thus simplifying the equations.

2.9.2.1 Longitudinal Aerodynamics

The longitudinal aerodynamics creates force and moment in the $\mathbf{i}^b\text{-}\mathbf{k}^b$ plane, also called the pitch plane. The forces are in the \mathbf{i}^b and \mathbf{k}^b plane also called the drag force and lift force respectively and the moment about the \mathbf{j}^b axis.

It is heavily influenced by the angle of attack (α), pitch rate (q) and the elevator deflection (δ_e), In the general nonlinear case we have:

$$\begin{aligned}
 F_{\text{lift}} &\approx \frac{1}{2} \rho V_a^2 S C_L(\alpha, q, \delta_e) \\
 F_{\text{drag}} &\approx \frac{1}{2} \rho V_a^2 S C_D(\alpha, q, \delta_e) \\
 m &\approx \frac{1}{2} \rho V_a^2 S c C_m(\alpha, q, \delta_e)
 \end{aligned}$$

Expanding C_L as a Taylor series and keeping only the first order (linear) terms gives

$$\begin{aligned} F_{\text{lift}} &= \frac{1}{2} \rho V_a^2 S [C_{L_0} + C_{L_\alpha} \alpha + C_{L_q} q + C_{L_{\delta_e}} \delta_e] \\ &= \frac{1}{2} \rho V_a^2 S \left[C_{L_0} + C_{L_\alpha} \alpha + C_{L_q} \frac{c}{2V_a} q + C_{L_{\delta_e}} \delta_e \right] \end{aligned}$$

where the coefficients C_{L_0} , $C_{L_\alpha} \triangleq C_L \alpha$ (stability derivative), $C_{L_q} \triangleq C_L \frac{qc}{2V_a}$ (stability derivative), and $C_{L_{\delta_e}} \triangleq C_L \delta_e$ (control derivative) are dimensionless quantities.

$$\begin{aligned} F_{\text{drag}} &= \frac{1}{2} \rho V_a^2 S \left[C_{D_0} + C_{D_\alpha} \alpha + C_{D_q} \frac{c}{2V_a} q + C_{D_{\delta_e}} \delta_e \right] \\ m &= \frac{1}{2} \rho V_a^2 S c \left[C_{m_0} + C_{m_\alpha} \alpha + C_{m_q} \frac{c}{2V_a} q + C_{m_{\delta_e}} \delta_e \right] \end{aligned}$$

For the given above equation we would have:

$$C_L(\alpha) = (1 - \sigma(\alpha)) [C_{L_0} + C_{L_\alpha} \alpha] + \sigma(\alpha) [2 \operatorname{sign}(\alpha) \sin^2 \alpha \cos \alpha]$$

The blending function is given by:

$$\sigma(\alpha) = \frac{1 + e^{-M(\alpha - \alpha_0)} + e^{M(\alpha + \alpha_0)}}{(1 + e^{-M(\alpha - \alpha_0)}) (1 + e^{M(\alpha + \alpha_0)})}$$

Then,

$$\begin{aligned} C_D(\alpha) &= \text{parasitic drag} + \text{induced drag}; \\ C_D(\alpha) &= C_{D_p} + \frac{(C_{L_0} + C_{L_\alpha} \alpha)^2}{\pi e A R} \end{aligned}$$

where e is the Oswald efficiency factor

Now let us focus on the linear Lift and Drag Model which would be valid for the small deviation of the angle of attack from trimming condition, in this case:

$$\begin{aligned} C_L(\alpha) &= C_{L_0} + C_{L_\alpha} \alpha \\ C_D(\alpha) &= C_{D_0} + C_{D_\alpha} \alpha \end{aligned}$$

As the aerodynamic force (Lift and Drag) are acting on the stability frame, we have,

$$\mathbf{f}^s = \begin{pmatrix} f_x \\ f_z \end{pmatrix}$$

thus, we need to bring it to the body frame through the rotation by the angle of attack forming the rotation matrix, R_s^b .

$$\mathbf{f}_s^b = R_s^b \mathbf{f}^s$$

$$\begin{pmatrix} f_x \\ f_z \end{pmatrix} = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} -F_{\text{drag}} \\ -F_{\text{lift}} \end{pmatrix}$$

Thus, the total longitudinal force acting on the body would be given as;

$$\begin{pmatrix} f_x \\ f_z \end{pmatrix} = \frac{1}{2}\rho V_a^2 S \begin{pmatrix} [-C_D(\alpha) \cos \alpha + C_L(\alpha) \sin \alpha] \\ + [-C_{D_q} \cos \alpha + C_{L_q} \sin \alpha] \frac{c}{2V_a} q + [-C_{D_{\delta_e}} \cos \alpha + C_{L_{\delta_e}} \sin \alpha] \delta_e \\ [-C_D(\alpha) \sin \alpha - C_L(\alpha) \cos \alpha] \\ + [-C_{D_q} \sin \alpha - C_{L_q} \cos \alpha] \frac{c}{2V_a} q + [-C_{D_{\delta_e}} \sin \alpha - C_{L_{\delta_e}} \cos \alpha] \delta_e \end{pmatrix}$$

and the pitching moment would be given as:

$$\mathbf{m} = \frac{1}{2}\rho V_a^2 S c \left[C_{m_0} + C_{m_\alpha} \alpha + C_{m_q} \frac{c}{2V_a} q + C_{m_{\delta_e}} \delta_e \right]$$

2.9.2.2 Lateral Aerodynamics

The Lateral aerodynamics creates force and moment in the \mathbf{i}^b - \mathbf{j}^b plane, also called the pitch plane. The forces are in the \mathbf{i}^b and \mathbf{j}^b direction and the moments about the \mathbf{i}^b and \mathbf{k}^b axis.

$$\begin{aligned} f_y &= \frac{1}{2}\rho V_a^2 S C_Y(\beta, p, r, \delta_a, \delta_r) \\ l &= \frac{1}{2}\rho V_a^2 S b C_l(\beta, p, r, \delta_a, \delta_r) \\ n &= \frac{1}{2}\rho V_a^2 S b C_n(\beta, p, r, \delta_a, \delta_r) \end{aligned}$$

$$\begin{aligned} f_y &\approx \frac{1}{2}\rho V_a^2 S \left[C_{Y_0} + C_{Y_\beta} \beta + C_{Y_p} \frac{b}{2V_a} p + C_{Y_r} \frac{b}{2V_a} r + C_{Y_{\delta_a}} \delta_a + C_{Y_{\delta_r}} \delta_r \right] \\ l &\approx \frac{1}{2}\rho V_a^2 S b \left[C_{l_0} + C_{l_\beta} \beta + C_{l_p} \frac{b}{2V_a} p + C_{l_r} \frac{b}{2V_a} r + C_{l_{\delta_a}} \delta_a + C_{l_{\delta_r}} \delta_r \right] \\ n &\approx \frac{1}{2}\rho V_a^2 S b \left[C_{n_0} + C_{n_\beta} \beta + C_{n_p} \frac{b}{2V_a} p + C_{n_r} \frac{b}{2V_a} r + C_{n_{\delta_a}} \delta_a + C_{n_{\delta_r}} \delta_r \right] \end{aligned}$$

where for symmetric aircraft, $C_{Y_0} = C_{l_0} = C_{n_0} = 0$.

C_{m_α} , C_{ℓ_β} , C_{n_β} , C_{m_q} , C_{ℓ_p} , C_{n_r} are called the *stability derivatives* because their values determine the stability of the aircraft.

Static Stability Derivatives

- C_{m_α} - longitudinal static stability derivative. Must be ≤ 0 for stability: increase in α causes and downward pitching moment.
- C_{ℓ_β} - roll static stability derivative. Associated with dihedral in wings. Must be ≤ 0 for stability: positive roll ϕ causes a restoring moment.
- C_{n_β} - yaw static stability derivative. Weathercock stability derivative. Influenced by design of tail. Causes airframe to align with the wind vector. Must be greater than or equal to zero for stability: cocks airframe into wind driving β to zero.

Dynamic Stability Derivatives

- C_{m_q} , C_{ℓ_p} , C_{n_r} are known as the pitch damping derivative, roll damping derivative, and yaw damping derivative, respectively. They quantify the level of damping associated with angular motion of the airframe.

Control Derivatives

- $C_{m_{\delta_e}}$, $C_{\ell_{\delta_a}}$, and $C_{n_{\delta_r}}$ are the primary control derivatives and quantify the effect on the control surfaces on their primary intended axes of influence.
- $C_{\ell_{\delta_r}}$ and $C_{n_{\delta_a}}$ are the cross-control derivatives.

The Aerodynamic data are generally given by the computation fluid dynamics software(CFD) or wind tunnel analysis of the design. One of the interesting open source software is the XFLR5 software [8].

2.9.3 Propeller Thrust

The propulsive force is acting along the \mathbf{i}^w direction of the wind frame, thus, we need to rotate it to the body frame.

The pressure entering the prop would be given as:

$$P_{\text{upstream}} = P_0 + \frac{1}{2}\rho V_a^2$$

The pressure exiting the prop would be given as:

$$P_{\text{downstream}} = P_0 + \frac{1}{2}\rho V_{\text{exit}}^2$$

where, P_0 is the static pressure, ρ is the density of air, V_{exit} is the speed of the air as it leaves the propeller.

The exit velocity:

$$V_{\text{exit}} = (k_{\text{motor}}\delta_t - V_a)$$

where, k_{motor} is the motor efficiency constant and δ_t is the control input which is the throttle deflection.

Total Force generated by the prop is given by:

$$\begin{aligned} F_{x_p} &= S_{\text{prop}} C_{\text{prop}} (P_{\text{downstream}} - P_{\text{upstream}}) \\ &= \frac{1}{2} \rho S_{\text{prop}} C_{\text{prop}} [(k_{\text{motor}}\delta_t)^2 - V_a^2] \\ \mathbf{f}_p &= \frac{1}{2} \rho S_{\text{prop}} C_{\text{prop}} \begin{pmatrix} (k_{\text{motor}}\delta_t)^2 - V_a^2 \\ 0 \\ 0 \end{pmatrix} \end{aligned}$$

Where, S_{prop} is the area swept out by the propeller, C_{prop} is the aerodynamic coefficient for the propeller. The propeller torque is modeled as:

$$\begin{aligned} T_p &= -k_{T_p} (k_{\Omega}\delta_t)^2 \\ \mathbf{m}_p &= \begin{pmatrix} -k_{T_p} (k_{\Omega}\delta_t)^2 \\ 0 \\ 0 \end{pmatrix} \end{aligned}$$

$$\text{Propeller speed}(\Omega) = k_{\Omega}\delta_t$$

where, k_{T_p} is the constant for the propeller determined by the experiment. Mostly we would consider the effects of this propeller torque are usually relatively minor. If unaccounted for, the propeller torque will cause a slow rolling motion in the direction opposite the propeller rotation. It is easily corrected by applying a small aileron deflection, which generates a rolling moment to counteract the propeller torque [2].

2.10 Summary

2.10.1 Force

$$\begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} = \begin{pmatrix} -mg \sin \theta \\ mg \cos \theta \sin \phi \\ mg \cos \theta \cos \phi \end{pmatrix} + \frac{1}{2} \rho V_a^2 S \begin{pmatrix} C_X(\alpha) + C_{X_q}(\alpha) \frac{c}{2V_a} q + C_{X_{\delta_e}}(\alpha) \delta_e \\ C_{Y_0} + C_{Y_\beta} \beta + C_{Y_p} \frac{b}{2V_a} p + C_{Y_r} \frac{b}{2V_a} r + C_{Y_{\delta_a}} \delta_a + C_{Y_{\delta_r}} \delta_r \\ C_Z(\alpha) + C_{Z_q}(\alpha) \frac{c}{2V_a} q + C_{Z_{\delta_e}}(\alpha) \delta_e \end{pmatrix} + \frac{1}{2} \rho S_{\text{prop}} C_{\text{prop}} \begin{pmatrix} (k_{\text{motor}} \delta_t)^2 - V_a^2 \\ 0 \\ 0 \end{pmatrix}$$

Where,

$$C_X(\alpha) \triangleq -C_D(\alpha) \cos \alpha + C_L(\alpha) \sin \alpha$$

$$C_{X_q}(\alpha) \triangleq -C_{D_q} \cos \alpha + C_{L_q} \sin \alpha$$

$$C_{X_{\delta_e}}(\alpha) \triangleq -C_{D_{\delta_e}} \cos \alpha + C_{L_{\delta_e}} \sin \alpha$$

$$C_Z(\alpha) \triangleq -C_D(\alpha) \sin \alpha - C_L(\alpha) \cos \alpha$$

$$C_{Z_q}(\alpha) \triangleq -C_{D_q} \sin \alpha - C_{L_q} \cos \alpha$$

$$C_{Z_{\delta_e}}(\alpha) \triangleq -C_{D_{\delta_e}} \sin \alpha - C_{L_{\delta_e}} \cos \alpha$$

2.10.2 Moments

$$\begin{pmatrix} l \\ m \\ n \end{pmatrix} = \frac{1}{2} \rho V_a^2 S \begin{pmatrix} b \left[C_{l_0} + C_{l_\beta} \beta + C_{l_p} \frac{b}{2V_a} p + C_{l_r} \frac{b}{2V_a} r + C_{l_{\delta_a}} \delta_a + C_{l_{\delta_r}} \delta_r \right] \\ c \left[C_{m_0} + C_{m_\alpha} \alpha + C_{m_q} \frac{c}{2V_a} q + C_{m_{\delta_e}} \delta_e \right] \\ b \left[C_{n_0} + C_{n_\beta} \beta + C_{n_p} \frac{b}{2V_a} p + C_{n_r} \frac{b}{2V_a} r + C_{n_{\delta_a}} \delta_a + C_{n_{\delta_r}} \delta_r \right] \end{pmatrix} + \begin{pmatrix} -k_{T_p} (k_\Omega \delta_t)^2 \\ 0 \\ 0 \end{pmatrix}$$

2.10.3 Nonlinear Equations

The nonlinear equation are as follows:

$$\begin{aligned}
 \dot{p}_n &= (\cos \theta \cos \psi)u + (\sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi)v + (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi)w \\
 \dot{p}_e &= (\cos \theta \sin \psi)u + (\sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi)v + (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi)w \\
 \dot{h} &= u \sin \theta - v \sin \phi \cos \theta - w \cos \phi \cos \theta \\
 \dot{u} &= rv - qw - g \sin \theta + \frac{\rho V_a^2 S}{2m} \left[C_X(\alpha) + C_{X_q}(\alpha) \frac{cq}{2V_a} + C_{X_{\delta_e}}(\alpha) \delta_e \right] + \frac{\rho S_{\text{prop}} C_{\text{prop}}}{2m} [(k_{\text{motor}} \delta_t)^2 - V_a^2] \\
 \dot{v} &= pw - ru + g \cos \theta \sin \phi + \frac{\rho V_a^2 S}{2m} \left[C_{Y_0} + C_{Y_\beta} \beta + C_{Y_p} \frac{bp}{2V_a} + C_{Y_r} \frac{br}{2V_a} + C_{Y_{\delta_a}} \delta_a + C_{Y_{\delta_r}} \delta_r \right] \\
 \dot{w} &= qu - pv + g \cos \theta \cos \phi + \frac{\rho V_a^2 S}{2m} \left[C_Z(\alpha) + C_{Z_q}(\alpha) \frac{cq}{2V_a} + C_{Z_{\delta_e}}(\alpha) \delta_e \right] \\
 \dot{\phi} &= p + q \sin \phi \tan \theta + r \cos \phi \tan \theta \\
 \dot{\theta} &= q \cos \phi - r \sin \phi \\
 \dot{\psi} &= q \sin \phi \sec \theta + r \cos \phi \sec \theta \\
 \dot{p} &= \Gamma_1 pq - \Gamma_2 qr + \frac{1}{2} \rho V_a^2 S b \left[C_{p_0} + C_{p_\beta} \beta + C_{p_p} \frac{bp}{2V_a} + C_{p_r} \frac{br}{2V_a} + C_{p_{\delta_a}} \delta_a + C_{p_{\delta_r}} \delta_r \right] \\
 \dot{q} &= \Gamma_5 pr - \Gamma_6 (p^2 - r^2) + \frac{\rho V_a^2 S c}{2J_y} \left[C_{m_0} + C_{m_\alpha} \alpha + C_{m_q} \frac{cq}{2V_a} + C_{m_{\delta_e}} \delta_e \right] \\
 \dot{r} &= \Gamma_7 pq - \Gamma_1 qr + \frac{1}{2} \rho V_a^2 S b \left[C_{r_0} + C_{r_\beta} \beta + C_{r_p} \frac{bp}{2V_a} + C_{r_r} \frac{br}{2V_a} + C_{r_{\delta_a}} \delta_a + C_{r_{\delta_r}} \delta_r \right]
 \end{aligned}$$

2.11 Linearization

Model linearization is based on the small disturbance theory. According to this theory, analysis is done under small perturbations of motion characteristics [10]. Before the linearization it is important to understand the equilibrium state of the physical system also known as the trim condition of the aircraft. The basic block model is shown in the figure 2.9. There are different flight envelope and thus the linearization is needed to produce respecting all the flight condition. The steps involved for the computation of the state space model of the aircraft include:

- Compute the trim state
- Decouple the model
- For each mode consider the other mode to zero, for example for longitudinal mode, later model would become zero.
- Compute the Jacobin of each mode

2.11.1 Equilibrium state

Stability is a property of an equilibrium state. To discuss stability we must first define what is meant by equilibrium. If an airplane is to remain in steady uniform flight, the resultant force as well as the resultant moment about the center of gravity must both be equal to 0. An airplane satisfying this requirement is said to be in a state of equilibrium or flying at a trim condition [13].

For a small unmanned aerial vehicle we would have the following approximation:

- Flat earth
- Mass of the aircraft is constant
- No change in air density(ρ) due to altitude
- All the angular and linear acceleration equals to zero

$$\{\dot{p}, \dot{q}, \dot{r}, \dot{u}, \dot{v}, \dot{w}, \dot{V}_a\}$$

- Rate of change of angle of attack($\dot{\alpha}$) and rate of change of side-slip angle($\dot{\beta}$) is equal to zero.

The aircraft is supposed to operate in different flight condition and every flight envelope has its own trimming points thus, based on each flight envelope we would have different consideration. The aircraft scenarios are as follows:

- Steady Straight and Level flight

$$\{\dot{\phi}, \dot{\theta}, \dot{\psi}\} = 0$$

- Steady Turn

$$\{\dot{\theta}, \dot{\phi}\} = 0$$

$\dot{\psi}$ is non zero and called turn rate

- Steady Level Climb.

$$\{\dot{\phi}, \dot{\psi}\} = 0$$

$\dot{\theta}$ is called climb rate

- Climbing Turn.

$$\dot{\phi} = 0$$

$\dot{\psi}$ is called turn rate and $\dot{\theta}$ is called climb rate and are non zero.

2.11.2 Decoupling

For an aircraft it is common to assume that the longitudinal modes are decoupled from the lateral modes. The key assumption is that the fuselage is slender that is, the length is much larger than the width and the height of the aircraft. It is also assumed that the longitudinal velocity is much larger than the vertical and transversal velocities [9].

In order to decouple the rigid-body kinetics in longitudinal and lateral modes it will be assumed that the states (v, p, r, ϕ) are negligible and assumed zero in the longitudinal channel while (u, w, q, θ) are negligible when considering the lateral channel.

2.11.3 State Space Modelling

For any nonlinear equations given by $\dot{x} = f(x, u)$ where, x is the state and u is the input. The trim condition is $\dot{x}^* = f(x^*, u^*)$.

Let $\bar{x} = x - x^*$ be the deviation from trim. Then,

$$\begin{aligned} \dot{\bar{x}} &= \dot{x} - \dot{x}^* \\ &= f(x, u) - f(x^*, u^*) \\ &= f(x + x^* - x^*, u + u^* - u^*) - f(x^*, u^*) \\ &= f(x^* + \bar{x}, u^* + \bar{u}) - f(x^*, u^*) \end{aligned}$$

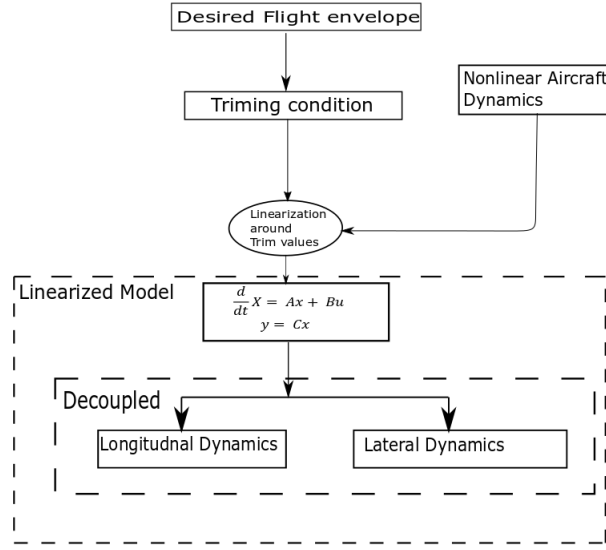


Figure 2.9: Block diagram of Linearization steps

Using a Taylor series expansion around trim gives

$$\begin{aligned}
 \dot{\bar{x}} &= f(x^*, u^*) + \frac{\partial f(x^*, u^*)}{\partial x} \bar{x} + \frac{\partial f(x^*, u^*)}{\partial u} \bar{u} + H.O.T - f(x^*, u^*) \\
 &\approx \frac{\partial f(x^*, u^*)}{\partial x} \bar{x} + \frac{\partial f(x^*, u^*)}{\partial u} \bar{u} \\
 &\triangleq A\bar{x} + B\bar{u}
 \end{aligned}$$

Thus, we have the state matrix **A** and input matrix **B** for the system. Since the system is LTI the matrices **A** and **B** have constant elements

2.11.4 Longitudinal Dynamics

Now formulating the longitudinal dynamics equation we would have (all the lateral consideration are zero).

$$\text{states}(\dot{\mathbf{x}}_{lon}) = \begin{pmatrix} u \\ w \\ q \\ \theta \\ h \end{pmatrix} \quad \text{Input}(\mathbf{u}_{lon}) = \begin{pmatrix} \delta_e \\ \delta_t \end{pmatrix}$$

The given nonlinear equations of motion for the longitudinal mode is given by:

$$\begin{aligned}
 \dot{u} &= -qw - g \sin \theta + \frac{\rho(u^2 + w^2)S}{2m} \left[C_{X_0} + C_{X_\alpha} \tan^{-1} \left(\frac{w}{u} \right) + C_{X_{\delta_e}} \delta_e \right] \\
 &\quad + \frac{\rho\sqrt{u^2 + w^2}S}{4m} C_{X_q} cq + \frac{\rho S_{\text{prop}}}{2m} C_{\text{prop}} [(k\delta_t)^2 - (u^2 + w^2)] \\
 \dot{w} &= qu + g \cos \theta + \frac{\rho(u^2 + w^2)S}{2m} \left[C_{Z_0} + C_{Z_\alpha} \tan^{-1} \left(\frac{w}{u} \right) + C_{Z_{\delta_e}} \delta_e \right] \\
 &\quad + \frac{\rho\sqrt{u^2 + w^2}S}{4m} C_{Z_q} cq \\
 \dot{q} &= \frac{1}{2J_y} \rho(u^2 + w^2) c S \left[C_{m_0} + C_{m_\alpha} \tan^{-1} \left(\frac{w}{u} \right) + C_{m_{\delta_e}} \delta_e \right] + \frac{1}{4J_y} \rho\sqrt{u^2 + w^2} S C_{m_q} c^2 q \\
 \dot{\theta} &= q \\
 \dot{h} &= u \sin \theta - w \cos \theta
 \end{aligned}$$

where we have used $\alpha = \tan^{-1} \left(\frac{w}{u} \right)$ and $V_a = \sqrt{u^2 + w^2}$.

Now, we would compute the Jacobin and thus would obtain the linearized state space model.

2.11.4.1 Jacobian Matrices

$$A_{\text{lon}} = \frac{\partial f_{\text{lon}}}{\partial x_{\text{lon}}} = \begin{pmatrix} \frac{\partial \dot{u}}{\partial u} & \frac{\partial \dot{u}}{\partial w} & \frac{\partial \dot{u}}{\partial q} & \frac{\partial \dot{u}}{\partial \theta} & \frac{\partial \dot{u}}{\partial h} \\ \frac{\partial \dot{w}}{\partial u} & \frac{\partial \dot{w}}{\partial w} & \frac{\partial \dot{w}}{\partial q} & \frac{\partial \dot{w}}{\partial \theta} & \frac{\partial \dot{w}}{\partial h} \\ \frac{\partial \dot{q}}{\partial u} & \frac{\partial \dot{q}}{\partial w} & \frac{\partial \dot{q}}{\partial q} & \frac{\partial \dot{q}}{\partial \theta} & \frac{\partial \dot{q}}{\partial h} \\ \frac{\partial \dot{\theta}}{\partial u} & \frac{\partial \dot{\theta}}{\partial w} & \frac{\partial \dot{\theta}}{\partial q} & \frac{\partial \dot{\theta}}{\partial \theta} & \frac{\partial \dot{\theta}}{\partial h} \\ \frac{\partial \dot{h}}{\partial u} & \frac{\partial \dot{h}}{\partial w} & \frac{\partial \dot{h}}{\partial q} & \frac{\partial \dot{h}}{\partial \theta} & \frac{\partial \dot{h}}{\partial h} \end{pmatrix} \quad B_{\text{lon}} = \frac{\partial f_{\text{lon}}}{\partial u_{\text{lon}}} = \begin{pmatrix} \frac{\partial \dot{u}}{\partial \delta_e} & \frac{\partial \dot{u}}{\partial \delta_t} \\ \frac{\partial \dot{w}}{\partial \delta_e} & \frac{\partial \dot{w}}{\partial \delta_t} \\ \frac{\partial \dot{q}}{\partial \delta_e} & \frac{\partial \dot{q}}{\partial \delta_t} \\ \frac{\partial \dot{\theta}}{\partial \delta_e} & \frac{\partial \dot{\theta}}{\partial \delta_t} \\ \frac{\partial \dot{h}}{\partial \delta_e} & \frac{\partial \dot{h}}{\partial \delta_t} \end{pmatrix}$$

2.11.4.2 Longitudinal State-Space Equation

$$\begin{pmatrix} \dot{\bar{u}} \\ \dot{\bar{w}} \\ \dot{\bar{q}} \\ \dot{\bar{\theta}} \\ \dot{\bar{h}} \end{pmatrix} = \begin{pmatrix} X_u & X_w & X_q & -g \cos \theta^* & 0 \\ Z_u & Z_w & Z_q & -g \sin \theta^* & 0 \\ M_u & M_w & M_q & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \sin \theta^* & -\cos \theta^* & 0 & u^* \cos \theta^* + w^* \sin \theta^* & 0 \end{pmatrix} \begin{pmatrix} \bar{u} \\ \bar{w} \\ \bar{q} \\ \bar{\theta} \\ \bar{h} \end{pmatrix} + \begin{pmatrix} X_{\delta_e} & X_{\delta_t} \\ Z_{\delta_e} & 0 \\ M_{\delta_e} & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \bar{\delta}_e \\ \bar{\delta}_t \end{pmatrix}$$

The equation of the coefficient above can be found in the table A.1.

2.11.5 Lateral Dynamics

Now formulating the lateral dynamics equation we would have (all the longitudinal consideration are zero).

$$\text{States}(\dot{\mathbf{x}}_{\text{lat}}) = \begin{pmatrix} v \\ p \\ r \\ \phi \\ \psi \end{pmatrix} \quad \text{Input}(\mathbf{u}_{\text{lat}}) = \begin{pmatrix} \delta_a \\ \delta_r \end{pmatrix}$$

The given nonlinear equations of motion for the Lateral Dynamics is given by:

$$\begin{aligned}
 \dot{v} &= pw - ru + g \cos \theta \sin \phi + \frac{\rho \sqrt{u^2 + v^2 + w^2} S b}{2m} \left[C_{Y_p} p + C_{Y_r} r \right] \\
 &\quad + \frac{\rho(u^2 + v^2 + w^2) S}{2m} \left[C_{Y_0} + C_{Y_\beta} \tan^{-1} \left(\frac{v}{\sqrt{u^2 + w^2}} \right) + C_{Y_{\delta_a}} \delta_a + C_{Y_{\delta_r}} \delta_r \right] \\
 \dot{p} &= \Gamma_1 pq - \Gamma_2 qr + \frac{\rho \sqrt{u^2 + v^2 + w^2} S b^2}{2} \left[C_{p_p} p + C_{p_r} r \right] \\
 &\quad + \frac{1}{2} \rho(u^2 + v^2 + w^2) S b \left[C_{p_0} + C_{p_\beta} \tan^{-1} \left(\frac{v}{\sqrt{u^2 + w^2}} \right) + C_{p_{\delta_a}} \delta_a + C_{p_{\delta_r}} \delta_r \right] \\
 \dot{r} &= \Gamma_7 pq - \Gamma_1 qr + \frac{\rho \sqrt{u^2 + v^2 + w^2} S b^2}{2} \left[C_{r_p} p + C_{r_r} r \right] \\
 &\quad + \frac{1}{2} \rho(u^2 + v^2 + w^2) S b \left[C_{r_0} + C_{r_\beta} \tan^{-1} \left(\frac{v}{\sqrt{u^2 + w^2}} \right) + C_{r_{\delta_a}} \delta_a + C_{r_{\delta_r}} \delta_r \right] \\
 \dot{\phi} &= p + q \sin \phi \tan \theta + r \cos \phi \tan \theta \\
 \dot{\psi} &= q \sin \phi \sec \theta + r \cos \phi \sec \theta
 \end{aligned}$$

where we have used $\beta = \tan^{-1} \left(\frac{v}{\sqrt{u^2 + w^2}} \right)$ and $V_a = \sqrt{u^2 + v^2 + w^2}$.

Now, as similar to the longitudinal case we would perform the Jacobin and thus we would obtain linearized state space model.

2.11.5.1 Jacobian Matrices

$$A_{\text{lat}} = \frac{\partial f_{\text{lat}}}{\partial x_{\text{lat}}} = \begin{pmatrix} \frac{\partial \dot{v}}{\partial v} & \frac{\partial \dot{v}}{\partial p} & \frac{\partial \dot{v}}{\partial r} & \frac{\partial \dot{v}}{\partial \phi} & \frac{\partial \dot{v}}{\partial \psi} \\ \frac{\partial \dot{p}}{\partial v} & \frac{\partial \dot{p}}{\partial p} & \frac{\partial \dot{p}}{\partial r} & \frac{\partial \dot{p}}{\partial \phi} & \frac{\partial \dot{p}}{\partial \psi} \\ \frac{\partial \dot{r}}{\partial v} & \frac{\partial \dot{r}}{\partial p} & \frac{\partial \dot{r}}{\partial r} & \frac{\partial \dot{r}}{\partial \phi} & \frac{\partial \dot{r}}{\partial \psi} \\ \frac{\partial \dot{\phi}}{\partial v} & \frac{\partial \dot{\phi}}{\partial p} & \frac{\partial \dot{\phi}}{\partial r} & \frac{\partial \dot{\phi}}{\partial \phi} & \frac{\partial \dot{\phi}}{\partial \psi} \\ \frac{\partial \dot{\psi}}{\partial v} & \frac{\partial \dot{\psi}}{\partial p} & \frac{\partial \dot{\psi}}{\partial r} & \frac{\partial \dot{\psi}}{\partial \phi} & \frac{\partial \dot{\psi}}{\partial \psi} \end{pmatrix} B_{\text{lat}} = \frac{\partial f_{\text{lat}}}{\partial u_{\text{lat}}} = \begin{pmatrix} \frac{\partial \dot{v}}{\partial \delta_a} & \frac{\partial \dot{v}}{\partial \delta_r} \\ \frac{\partial \dot{p}}{\partial \delta_a} & \frac{\partial \dot{p}}{\partial \delta_r} \\ \frac{\partial \dot{r}}{\partial \delta_a} & \frac{\partial \dot{r}}{\partial \delta_r} \\ \frac{\partial \dot{\phi}}{\partial \delta_a} & \frac{\partial \dot{\phi}}{\partial \delta_r} \\ \frac{\partial \dot{\psi}}{\partial \delta_a} & \frac{\partial \dot{\psi}}{\partial \delta_r} \end{pmatrix}$$

2.11.5.2 Lateral Space-Space Model

$$\begin{pmatrix} \dot{\bar{v}} \\ \dot{\bar{p}} \\ \dot{\bar{r}} \\ \dot{\bar{\phi}} \\ \dot{\bar{\psi}} \end{pmatrix} = \begin{pmatrix} Y_v & Y_p & Y_r & g \cos \theta^* \cos \phi^* & 0 \\ L_v & L_p & L_r & 0 & 0 \\ N_v & N_p & N_r & 0 & 0 \\ 0 & 1 & \cos \phi^* \tan \theta^* & q^* \cos \phi^* \tan \theta^* - r^* \sin \phi^* \tan \theta^* & 0 \\ 0 & 0 & \cos \phi^* \sec \theta^* & p^* \cos \phi^* \sec \theta^* - r^* \sin \phi^* \sec \theta^* & 0 \end{pmatrix} \begin{pmatrix} \bar{v} \\ \bar{p} \\ \bar{r} \\ \bar{\phi} \\ \bar{\psi} \end{pmatrix} + \begin{pmatrix} Y_{\delta_a} & Y_{\delta_r} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \bar{\delta}_a \\ \bar{\delta}_r \end{pmatrix}$$

The equation of the coefficient can be found in table A.2

Chapter 3

Guidance and control

After the development of the state space model in the previous chapter, now we will address the controller design. We will first design the autopilot and then work with the path following algorithm.

3.1 LQR controller

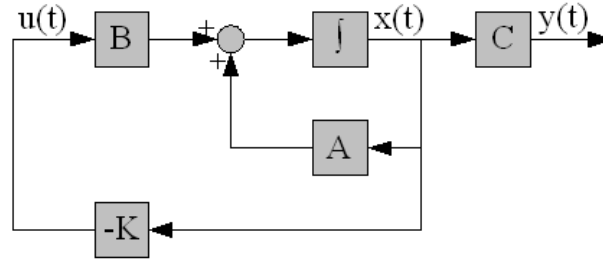


Figure 3.1: Block diagram LQR

Given the state space equation,

$$\dot{x} = Ax + Bu \quad (3.1)$$

and the cost function for the continuous infinite time horizon:

$$J = \int_0^\infty x^T Q x + u^T R u \quad (3.2)$$

where, $Q \geq 0$, is symmetric positive semi-definite matrix and $R > 0$, is symmetric positive definite matrix, in the Linear Quadratic Regulator (LQR) problem, the goal is to find the feedback control signal u as shown in the figure 3.1 that minimizes J . In equation 3.2 Q and R are called weight matrices, where Q focus on how important the state is, and R focuses on the how expensive the control cost is.

The optimal solution for the cost function 3.2 is

$$u = -Kx$$

where $K = R^{-1}B^T S$ and S satisfies the *Algebraic Riccati Equation* (ARE).

The general procedure for the LQR would be:

1. A and B matrix known from the plant model which is in the form of equation 3.1.

2. Choose Q and R matrices

3. Solve ARE for S

$$A^T S + SA - SBR^{-1}B^T S + Q = 0$$

4. Compute $K = R^{-1}B^T S$

5. Choose the K solution for the stable system

In Matlab we would directly use the function $lqr()$ and thus the controller gain is,

$$K_{lqr} = lqr(A, B, Q, R)$$

The linear feedback control signal would be

$$u = -K_{lqr}x$$

A major problem that needs to be addressed is the selection of the diagonal weights Q and R, which directly interfere in the control signal(u) in the system. It is the trail and error method that would give the best value of the Q and R matrices.

LQR with Integral Action

The objective of the integral is to remove the steady state error from the system. Given, the state space as equation:

$$\dot{x} = Ax + Bu \quad (3.3)$$

$$y = Cx \quad (3.4)$$

$$(3.5)$$

where, y is the output that we would want to drive to the reference signal (r). Then, we would consider the integral error as,

$$z = \int_0^t r(\tau) - y(\tau) d\tau \quad (3.6)$$

$$\dot{z} = r - Cx \quad (3.7)$$

the dynamic equation 3.3 and 3.7 can be written as:

$$\begin{pmatrix} \dot{x} \\ \dot{z} \end{pmatrix} = \begin{pmatrix} A & 0 \\ -C & 0 \end{pmatrix} \begin{pmatrix} x \\ z \end{pmatrix} + \begin{pmatrix} B \\ 0 \end{pmatrix} u + \begin{pmatrix} 0 \\ I \end{pmatrix} r \quad (3.8)$$

Design of tracking is to make the system stabilize, if $x(\infty), z(\infty)$ and $u(\infty)$ approach constant value, then $\dot{z} = 0$, so $y(\infty) = r$. In steady state condition equation 3.8 becomes:

$$\begin{pmatrix} \dot{x}(\infty) \\ \dot{z}(\infty) \end{pmatrix} = \begin{pmatrix} A & 0 \\ -C & 0 \end{pmatrix} \begin{pmatrix} x(\infty) \\ z(\infty) \end{pmatrix} + \begin{pmatrix} B \\ 0 \end{pmatrix} u(\infty) + \begin{pmatrix} 0 \\ I \end{pmatrix} r(\infty) \quad (3.9)$$

considering r to be constant, then $r(\infty) = r(t) = r$ is constant value, for $t > 0$. subtracting Equation 3.8 with 3.10, then we get:

$$\begin{aligned} \begin{pmatrix} \dot{x}(t) - \dot{x}(\infty) \\ \dot{z}(t) - \dot{z}(\infty) \end{pmatrix} &= \begin{pmatrix} A & 0 \\ -C & 0 \end{pmatrix} \begin{pmatrix} x(t) - x(\infty) \\ z(t) - z(\infty) \end{pmatrix} + \begin{pmatrix} B \\ 0 \end{pmatrix} (u(t) - u(\infty)) \\ &\quad + \begin{pmatrix} 0 \\ I \end{pmatrix} (r(t) - r(\infty)) \end{aligned} \quad (3.10)$$

For,

$$x(t) - x(\infty) = x_e(t)$$

$$z(t) - z(\infty) = z_e(t)$$

$$u(t) - u(\infty) = u_e(t)$$

thus, the error equation can be written as:

$$\begin{pmatrix} \dot{x}_e \\ \dot{z}_e \end{pmatrix} = \underbrace{\begin{pmatrix} A & 0 \\ -C & 0 \end{pmatrix}}_{\bar{A}} \begin{pmatrix} x_e \\ z_e \end{pmatrix} + \underbrace{\begin{pmatrix} B \\ 0 \end{pmatrix}}_{\bar{B}} u_e \quad (3.11)$$

The control signal for the above equation is given by:

$$u_e(t) = -Kx_e(t) + k_I z_e(t) \quad (3.12)$$

Here, it formulated by considering the augmented matrix \bar{A} and \bar{B} , It is important to consider the dimension of the \bar{B} as it has the rows equal to the \bar{A} and column equal to the number of control input(u) and Q being diagonally equal to the number of rows of the \bar{A} matrix. The rest is similar with the LQR design method.

With the matlab command being;

$$K_{lqr} = lqr(\bar{A}, \bar{B}, Q, R)$$

3.2 Autopilot

The autopilot is responsible for all the flight envelope(take off, ascent, cruise, decent, landing). As, we have already decoupled the longitudinal and lateral mode, thus we would design the autopilot separately for both the condition.

3.2.1 Longitudinal Controller

The longitudinal autopilot is responsible to track the height and the velocity of the aircraft. As the state space matrix for the longitudinal controller is given in the section 2.11.4.2, where we would take the state $x_{lon} = (u, w, q, \theta, h)^T$ and the control input, $u = (\delta_e, \delta_t)$. The objective of the longitudinal controller is to drive the altitude(h) to the commanded altitude(h^c) and the airspeed(V_a) to commanded airspeed (V_a^c). Therefore we augment the state with:

$$\begin{aligned} \text{Error, } z_\theta &= \int_0^t \begin{pmatrix} h^c(\tau) - h(\tau) \\ V_a^c(\tau) - V_a(\tau) \end{pmatrix} d\tau \\ \dot{z} &= \begin{pmatrix} h^c \\ V_a^c \end{pmatrix} - H_{lon} x_{lon} \\ H_{lon} &= \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ \frac{u^*}{V_a^*} & \frac{w^*}{V_a^*} & 0 & 0 & 0 \end{pmatrix} \end{aligned}$$

Finally, we would have the augmented matrix as:

$$\bar{A}_{lon} = \begin{bmatrix} A_{lon} & [0] \\ -H_{lon} & [0] \end{bmatrix} \quad \bar{B}_{lon} = \begin{bmatrix} B_{lon} \\ [0] \end{bmatrix}$$

3.2.1.1 Height tracking

In the figure 3.2, we would see the behaviour of the nonlinear model of the aircraft to track the given height. The figure 3.3 and 3.4 shows the change in velocity and pitch

angle respectively, with figure 3.5 showing the height tracking error. The actuator movement is shown in figure 3.7 and 3.6 which corresponds to throttle deflection and elevator deflection respectively.

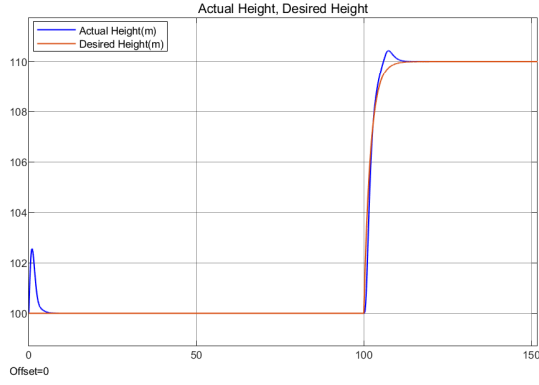


Figure 3.2: Height(m) vs time(seconds)

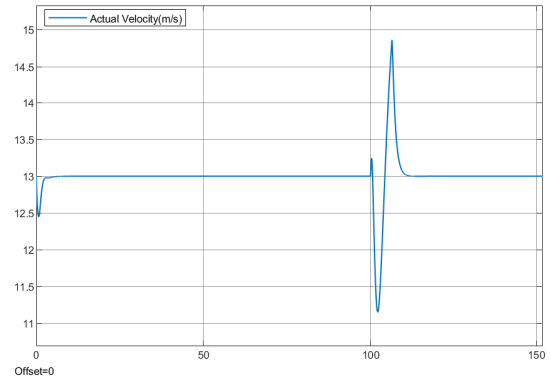
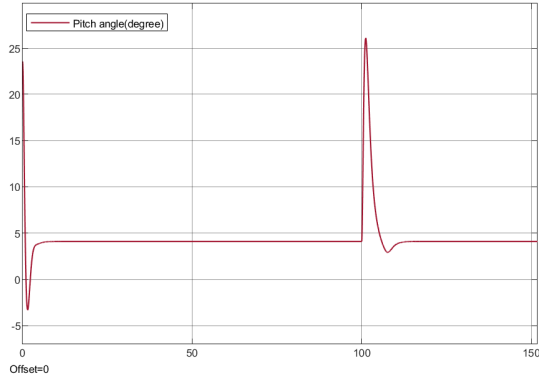
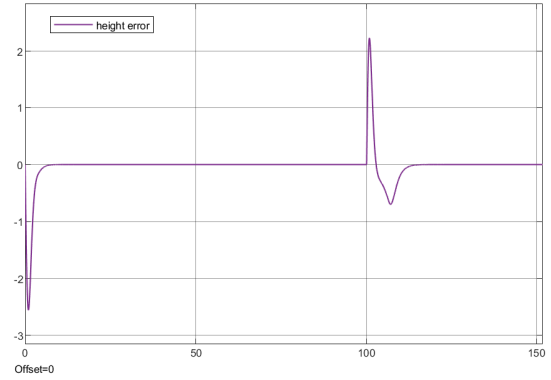
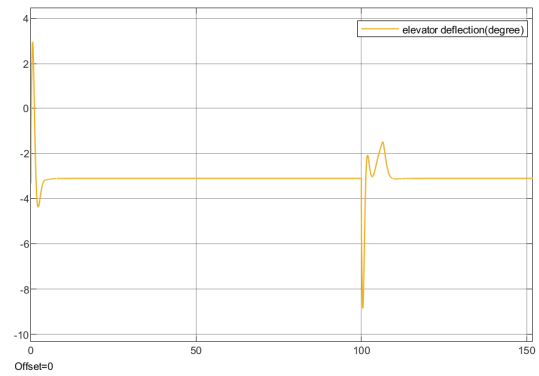
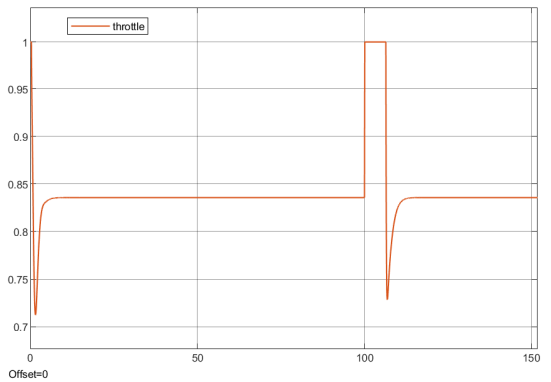

 Figure 3.3: Velocity(V_a)(m/s) vs time

 Figure 3.4: Pitch angle (θ)(in degree) based on height tracking vs time


Figure 3.5: Height tracking error vs time


 Figure 3.6: Elevator deflection(δ_e)(in degree) vs time

 Figure 3.7: Throttle movement(δ_t) vs time

3.2.1.2 Velocity tracking

Figure 3.8, displays velocity tracking behaviour of the aircraft. The figure 3.9 shows the change in pitch angle to maintain the velocity. It is very clear that, at a constant

altitude we could see the increase in velocity would decrease the pitch angle. The elevator deflection is shown in the figure 3.10 and the throttle movement is shown in the figure 3.11.

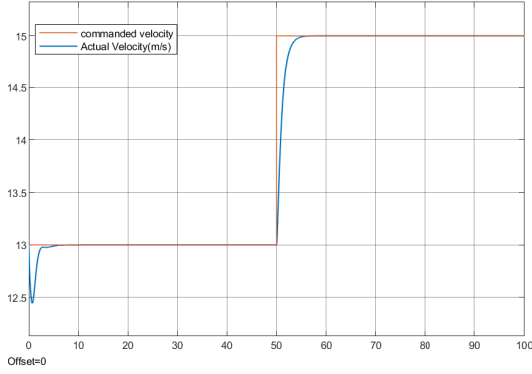


Figure 3.8: Velocity(m/s) vs time

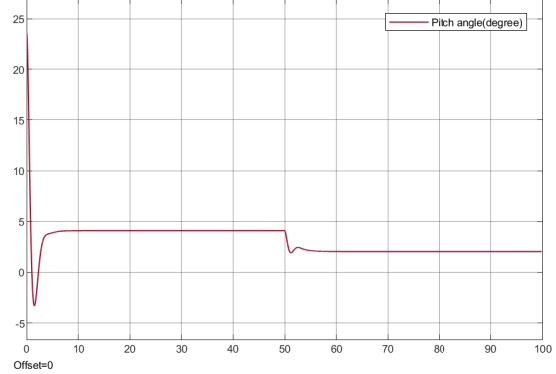


Figure 3.9: pitch angle(θ) vs time

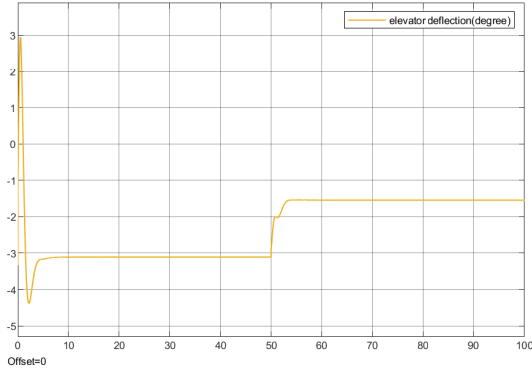


Figure 3.10: Elevator deflection(δ_e)(in degree) vs time

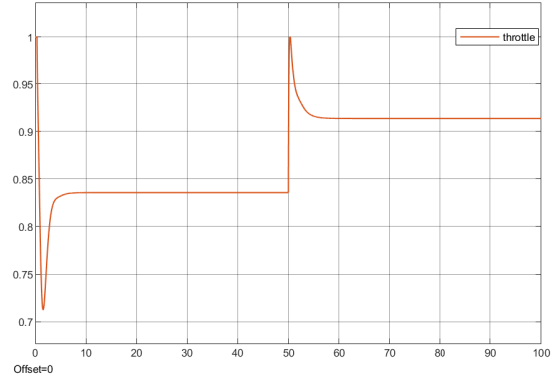


Figure 3.11: Throttle movement(δ_t) vs time

3.2.2 Lateral Controller

As derived in section 2.11.5.2, we have the state space equation as:

$$\dot{x}_{lat} = A_{lat}x + B_{lat}u$$

where, $x_{lat} = (v, p, r, \phi, \psi)^T$, and $u_{lat} = (\delta_a, \delta_r)^T$.

The objective of the lateral controller is to hold the course angle(χ) to the desired course angle(χ^c). This is achieved by assuming,

$$\psi = \chi$$

The integral error is given by:

$$z_\chi = \int_0^t \chi^c(\tau) - \chi(\tau) d\tau$$

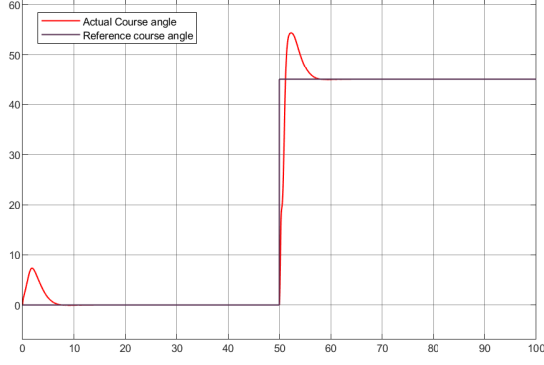
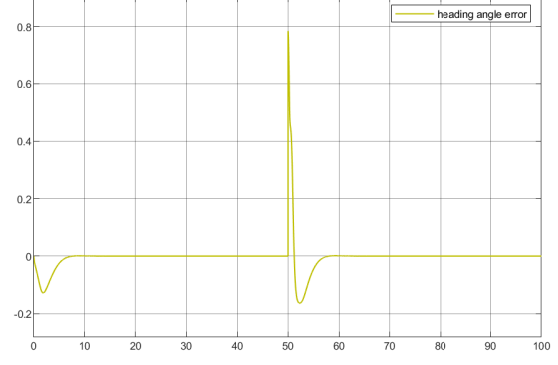
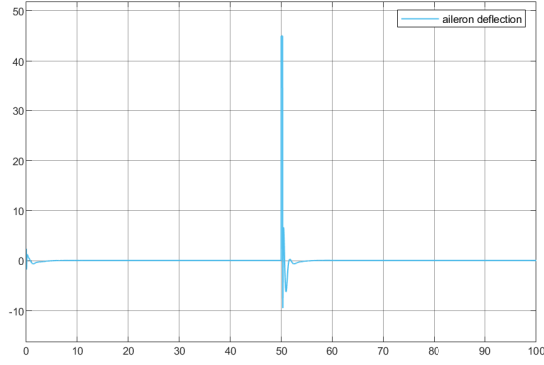
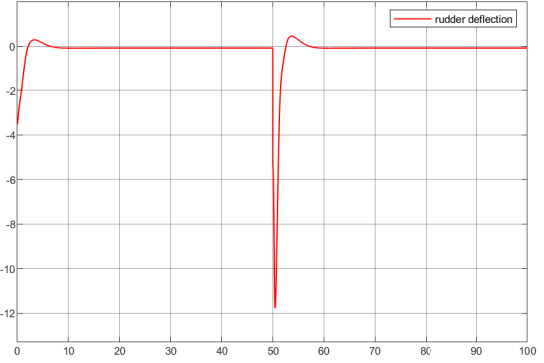
$$\dot{z}_\chi = \chi^c - C_{lat}x_{lat}$$

$$C_{lat} = [0 \ 0 \ 0 \ 0 \ 1]$$

Now the lateral augmented state space equation would be

$$\bar{A}_{lat} = \begin{bmatrix} A_{lat} & 0 \\ -C_{lat} & 0 \end{bmatrix} \quad \bar{B}_{lat} = \begin{bmatrix} B_{lat} \\ 0 \end{bmatrix}$$

Here, the the course controller or the heading controller would track the heading angle as shown in the figure 3.12 and the error is shown by the figure 3.13. The aileron and rudder movement required to track the given heading is shown in the figure 3.14 and 3.15 respectively.


 Figure 3.12: Course angle(χ)(in degree) vs time

 Figure 3.13: Heading angle error(e_{ψ}) based on height tracking vs time

 Figure 3.14: Aileron deflection(δ_a)(in degree) vs time

 Figure 3.15: Rudder deflection(δ_r)(in degree)

3.3 Path Manager

This section describes the vector field guidance strategies which deals with the straight-line and orbit following which can be used to follow a series of waypoints. The path manager is responsible to generate the desired course angle(χ^c), desired velocity(V_a^c) and desired altitude(h^c), which is then sent to the autopilot.

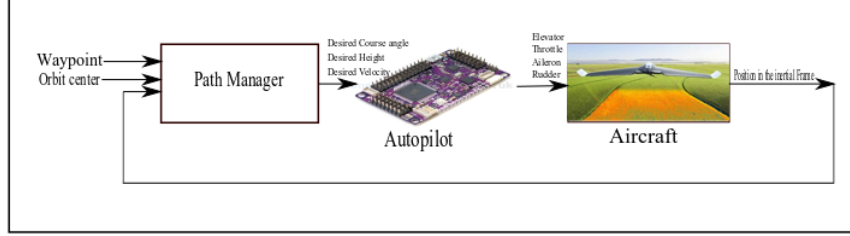


Figure 3.16: Architecture of the Control

3.3.1 Waypoint

A waypoint has three components one in each north, east and down direction. In this work we only consider the lateral case where the height is kept constant by the autopilot. Waypoint following is a common method of trajectory following in which several waypoints are selected on the reference trajectory and the vehicle is guided to pass through all waypoints [17] thus creating a path. We will use the notation:

$$\mathcal{W} = \{w_1, w_2, w_3, w_4, w_5 \dots w_n\}$$

where, \mathcal{W} , is the waypoint path with w_1, \dots, w_n being the waypoint point in the inertial frame.

Any two points in space could be joined by the straight line. Thus we could say that between the waypoint we would have straight line that connects them. The vehicle is required to accurately fly past the waypoints before reaching the final target. The trajectory to the second waypoint would be affected by the heading angle at the first waypoint. Similarly, the heading angle at the last waypoint determines the trajectory to the final target. Thus, an unreasonable heading angle at the current waypoint may result in the violation of constraints for the next waypoint or the final target [11].

Waypoint Switching

The problem related to the waypoint is to search for the proper method to switch between the waypoint in order to follow the path properly. To this end, we would consider the half plane method as mentioned in the reference [2].

The aircraft tracks straight-line path from \mathbf{w}_{i-1} to \mathbf{w}_i until it enters $\mathcal{H}(\mathbf{w}_i, \mathbf{n}_i)$, at which point it will track straight-line path from \mathbf{w}_i to \mathbf{w}_{i+1} .

For example, let us say that i is 2 and thus we would have three waypoint w_1 , w_2 and w_3 , the plane would be divided by the point w_2 , as shown in the figure 3.17, where the aircraft would be starting to track the straight line made with point w_1 and w_2 , and the function $\mathcal{H}(\mathbf{r}, \mathbf{n})$ which would define the plane would be negative

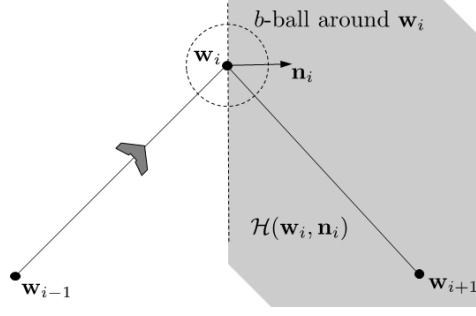


Figure 3.17: Half plane approach

till it crosses the middle waypoint w_2 and become positive as reached close to the waypoint w_2 and then i becomes 3 and now it starts to follow the line made with the points w_2 and w_3 and so on the code keeps running till the final value of the index is found.

Given point $\mathbf{r} \in \mathbb{R}^3$ and normal vector $\mathbf{n} \in \mathbb{R}^3$, define half plane

$$\mathcal{H}(\mathbf{r}, \mathbf{n}) \triangleq \{\mathbf{p} \in \mathbb{R}^3 : (\mathbf{p} - \mathbf{r})^T \mathbf{n} \geq 0\}$$

Define unit vector pointing in direction of line $\overline{\mathbf{w}_i \mathbf{w}_{i+1}}$ as

$$\mathbf{q}_i \triangleq \frac{\mathbf{w}_{i+1} - \mathbf{w}_i}{\|\mathbf{w}_{i+1} - \mathbf{w}_i\|}$$

Unit normal to the 3-D half plane that separates the line $\overline{\mathbf{w}_{i-1} \mathbf{w}_i}$ from the line $\overline{\mathbf{w}_i \mathbf{w}_{i+1}}$ is given by

$$\mathbf{n}_i \triangleq \frac{\mathbf{q}_{i-1} + \mathbf{q}_i}{\|\mathbf{q}_{i-1} + \mathbf{q}_i\|}$$

The final output of the waypoint algorithm would generate the unit directional vector \mathbf{q} , which contains the component in the north, east and down direction and \mathbf{r} , which is the origin for the straight line and also the component in the inter inertial frame.

The algorithm mentioned above is:

Waypoint path, $\mathbf{w} = \{w_1, w_2, w_3\}$

Initialize the waypoint index: $\mathbf{i} = 2$

$$\mathbf{r} = \mathbf{w}_{i-1}$$

$$\mathbf{q}_{i-1} \triangleq \frac{\mathbf{w}_i - \mathbf{w}_{i-1}}{\|\mathbf{w}_i - \mathbf{w}_{i-1}\|}$$

$$\mathbf{q}_i \triangleq \frac{\mathbf{w}_{i+1} - \mathbf{w}_i}{\|\mathbf{w}_{i+1} - \mathbf{w}_i\|}$$

$$\mathbf{n}_i \triangleq \frac{\mathbf{q}_{i-1} - \mathbf{q}_i}{\|\mathbf{q}_{i-1} - \mathbf{q}_i\|} \mathbf{H} = (\mathbf{p} - \mathbf{r})^T * \mathbf{n}$$

if,

$$\mathbf{H} \leq 0$$

if,

$$\mathbf{i} \leq \mathbf{N} - 1$$

$$\mathbf{i} = \mathbf{i} + 1$$

end
end

$$\mathbf{out} = [\mathbf{r}^T, \mathbf{q}_{i-1}^T]^T$$

We next describe the vector-field-based trajectory-following guidance laws that use a construction of vector fields that surround a reference trajectory and guide the vehicle toward it[12].

3.3.2 Straight line following

A straight line can be defined by two vectors in 3 dimensional plane one being the position and other being the direction.

$$\mathcal{P}_{line}(\mathbf{r}, \mathbf{q}) = \{\mathbf{x} \in \mathbb{R}^3 : \mathbf{x} = \mathbf{r} + \lambda \mathbf{q}, \lambda \in \mathbb{R}\}$$

In the above equation, \mathbf{r} is the origin of the line and \mathbf{q} is the unit vector that points to the direction of the line in the 3-D plane.

The course angle of the line as measured from the north is given by;

$$\chi_q \triangleq \text{atan2} \frac{q_e}{q_n} \quad (3.13)$$

where, $\mathbf{q} = (q_n, q_e, q_d)^T$, stating the north east and down component of the unit vector.

Line following problem is solved generally considering the frame relative to the straight line. Considering the rotation along the z-axis (which is aligned along the inertial frame), we will have the transformation from inertial frame to path frame as

$$\mathcal{R}_i^{\mathcal{P}} \triangleq \begin{pmatrix} \cos \chi_q & \sin \chi_q & 0 \\ -\sin \chi_q & \cos \chi_q & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.14)$$

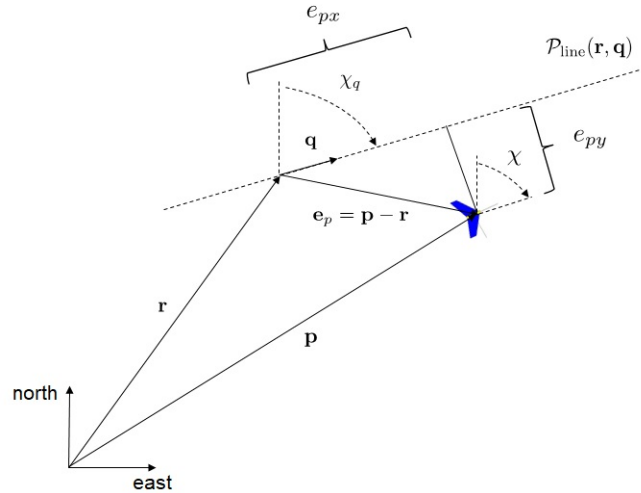


Figure 3.18: Straight line path

Path error(\mathbf{e}_p) as expressed in the path frame can be given as:

$$\mathbf{e}_p = \begin{pmatrix} e_{px} \\ e_{py} \\ e_{pz} \end{pmatrix} \triangleq \mathcal{R}_i^{\mathcal{P}} (\mathbf{p}^i - \mathbf{r}^i)$$

\mathbf{r} is a constant position in inertial frame and whose derivative is zero and \mathbf{p} is expressed in the inertial frame which is $[p_n, p_e, p_d]^T$ and the derivative is given by equation 2.11.

Let us consider the kinematics of the relative error dynamics in path frame with constant altitude:

$$\begin{aligned} \begin{pmatrix} \dot{e}_{px} \\ \dot{e}_{py} \end{pmatrix} &= \begin{pmatrix} \cos \chi_q & \sin \chi_q \\ -\sin \chi_q & \cos \chi_q \end{pmatrix} \begin{pmatrix} V_g \cos \chi \\ V_g \sin \chi \end{pmatrix} \\ &= V_g \begin{pmatrix} \cos(\chi - \chi_q) \\ \sin(\chi - \chi_q) \end{pmatrix} \end{aligned}$$

Regulate the cross-track error e_{py} to zero by commanding the course angle:

$$\dot{e}_{py} = V_g \sin(\chi - \chi_q)$$

The lateral straight line path following problem is to select χ^c so that e_{py} tends to zero when the course angle of the line χ_q is known. The strategy in this section will be to construct a desired course angle at every spatial point relative to the straight-line path that results in the UAV moving toward the path. The set of desired course angles at every point will be called a vector field because the desired course angle specifies a vector (relative to the straight line) with a magnitude of unity.

Finally the objective is to construct the vector field so that, when e_{py} is large, the aircraft approaches to the path with the course angle $\chi^\infty \in (0, \frac{\pi}{2}]$. The desired course the aircraft need to follow is based on the cross track error, So,

$$\chi_d(e_{py}) = -\chi^\infty \frac{2}{\pi} \tan^{-1}(k_{path} e_{py})$$

where, k_{path} is a positive constant that influences on how fast the aircraft has to approach the desired line, the rate of transition from the χ^∞ to zero.

Large values of k_{path} result in short, abrupt transitions, while small values of k_{path} cause long, smooth transitions in the desired course. The command for lateral path following is the sum of two terms given by:

$$\chi^c(t) = \chi_q + \chi^d = \chi_q - \chi^\infty \frac{2}{\pi} \tan^{-1}(k_{path} e_{py}) \quad (3.15)$$

The equation 3.15 is affected by the value of χ_q as derived in the equation 3.13, where atan2 returns the value in the $\pm\pi$. Slight change in χ_q would bring sudden large change in the course angle, which is not expected thus, we would have:

$$\chi_q = \text{atan2}(q_e, q_n) + 2\pi m \quad (3.16)$$

where $m = \{1, -1\}$ and depends as

$$-\pi \leq \chi_q - \chi \leq \pi$$

The algorithm to follow the straight line would be given as;

$$\begin{aligned}
 & \chi_q = \text{atan}(q_e, q_n) \\
 & \text{while } \chi_q - \chi \leq -\pi \\
 & \quad \chi_q = \chi_q + 2\pi \\
 & \text{while } \chi_q - \chi \geq \pi \\
 & \quad \chi_q = \chi_q - 2\pi \\
 & \text{end} \\
 & e_{py} = \sin \chi_q (p_n - r_n) + \cos \chi_q (p_e - r_e) \\
 & \chi^c = \chi_q - \chi^\infty \frac{2}{\pi} \tan^{-1}(k_{path} e_{py})
 \end{aligned}$$

3.3.3 Orbit following

We can define an orbit path as

$$\mathcal{P}_{\text{orbit}}(\mathbf{c}, \rho, \lambda) = \left\{ \mathbf{r} \in \mathbb{R}^3: \mathbf{r} = \mathbf{c} + \lambda \rho \begin{pmatrix} \cos \varphi & \sin \varphi & 0 \end{pmatrix}^\top, \varphi \in [0, 2\pi) \right\}$$

where, \mathbf{c} is the center of the circle in the inertial frame, ρ is the radius of the circle, λ is 1 for the clockwise direction and -1 for the anti-clockwise direction, and φ is the phase angle of the relative position as seen in the figure 3.19

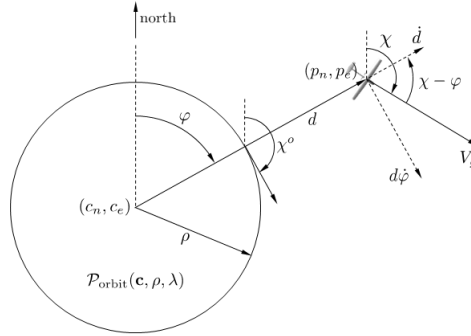


Figure 3.19: Circular path with radius ρ

It is easier to analyze in polar coordinates and we will consider constant altitude which would give the flight path angle (γ) to zero and thus the equation 2.11 can be rewritten as:

$$\begin{pmatrix} \dot{p}_n \\ \dot{p}_e \end{pmatrix} = \begin{pmatrix} V_g \cos \chi \\ V_g \sin \chi \end{pmatrix}$$

Geometrically from the figure 3.19, we would have the rate of distance from the center normally and tangentially in the polar coordinates would be given as:

$$\begin{aligned}
 \dot{d} &= V_g \cos(\chi - \varphi) \\
 \dot{\varphi} &= \frac{V_g}{d} \sin(\chi - \varphi)
 \end{aligned}$$

If we have the aircraft in the orbit we define, the χ^0 as below

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

The final objective for us is to bring the aircraft to the orbit by bringing the perpendicular distance to the radius of the circle and the course angle to the course angle of the orbit.

We would have two conditions for this case:

- When the distance d from the center is large than radius ρ then, we would have desired course angle(χ_d) $\approx \chi^o + \lambda \frac{\pi}{2}$.
- when distance d is equal the radius ρ , then, we would have desired course angle(χ_d) equal to χ^o .

Therefore, let the desired course angle be computed as:

$$\chi_d(d - \rho, \lambda) = \chi^o + \lambda \tan^{-1} \left(k_{\text{orbit}} \left(\frac{d - \rho}{\rho} \right) \right)$$

Then we would have the commanded course angle generated as:

$$\chi^c(t) = \varphi + \lambda \left[\frac{\pi}{2} + \tan^{-1} \left(k_{\text{orbit}} \left(\frac{d - \rho}{\rho} \right) \right) \right]$$

The orbit angle(φ) must be computed considering the effect of sudden jump of 2π while transiting from π to $-\pi$., thus,

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

The algorithm for the lateral circular path is given as follows:

$$\begin{aligned} d &= \sqrt{(p_n - c - n)^2 + (p_e - c_e)^2} \\ \varphi &= \text{atan2}(p_e - c_e, p_n - c_n) \\ \text{while } \varphi - \chi &\leq -\pi \\ \varphi &= \varphi + 2\pi \\ \text{while } \varphi - \chi &\geq -\pi \\ \varphi &= \varphi - 2\pi \\ \text{end} \\ \chi^c(t) &= \varphi + \lambda \left[\frac{\pi}{2} + \tan^{-1} \left(k_{\text{orbit}} \left(\frac{d - \rho}{\rho} \right) \right) \right] \end{aligned}$$

Chapter 4

Simulation results

This chapter describes the implementation and simulation of the UAV model and the proposed controllers for several test scenarios. For the Linearization we have used the Simulink and MATLAB directly. We create the Matlab function for the force and moments that has the state and actuator control as the input and generates the force as the output as shown in the figure 4.1.

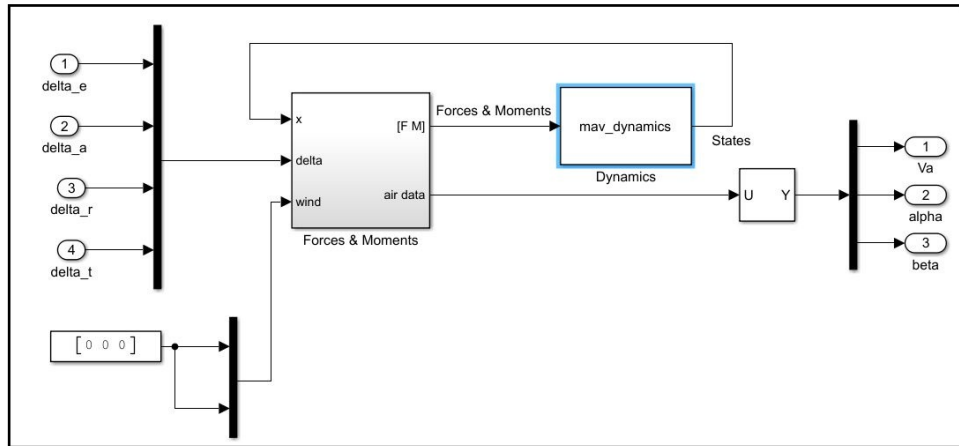


Figure 4.1: Trim block in Simulink

The model designed in the simulink includes the nonlinear model with all the nonlinear equations and thus we are now required to trim the model and linearize the model using the trim values. For this we use the *trim* function in the matlab and also use the *linmod* function that outputs the linearization around the generated trim values. The matlab code is given as:

$$\text{trim}(\text{filename}, x0, u0, y0, ix, iu, iy, dx0, idx)$$

$$[A, B, C, D] = \text{linmod}(\text{filename}, x_{\text{trim}}, u_{\text{trim}})$$

Now, once the state space matrix is generated we then separate the longitudinal and lateral state space matrix and thus obtain system dynamic matrices necessary for generating the LQR controller.

4.1 Aircraft Model

The next tables show the values of the model parameters used for simulation. These values were taken from [2].



Figure 4.2: Zagi 60 HP electric model airframe, source:[19]

4.1.1 Physical Parameter

Table 4.1: Geometry of the Aircraft

| Parameter | Description | Value |
|-------------|-------------------------------|-------------------------|
| m | mass of the aircraft | 1.56Kg |
| S | wing surface area | 0.2589 m^2 |
| b | wing span | 1.4224 m |
| c | chord length | 0.3302 m |
| S_{prop} | Surface area of the propeller | 0.0314 m^2 |
| ρ | Air density | 1.2682 kg/m^3 |
| k_{motor} | motor coefficient | 20 |
| e | oswald efficiency | 0.9 |

Table 4.2: Moment of Inertia Data

| Moment | Value |
|----------|-------------------------|
| J_X | 0.1147 Kg m^2 |
| J_Y | 0.0576 Kg m^2 |
| J_Z | 0.1712 Kg m^2 |
| J_{XZ} | 0.0015 Kg m^2 |

4.1.2 Aerodynamic coefficient

Table 4.3: Aerodynamic coefficient data

| Longitudinal coefficient | Value | Lateral coefficient | Value |
|--------------------------|----------|---------------------|----------|
| C_{L_0} | 0.09167 | C_{Y_0} | 0 |
| C_{D_0} | 0.01631 | C_{l_0} | 0 |
| C_{m_0} | -0.02338 | C_{n_0} | 0 |
| C_{L_α} | 3.5016 | C_{Y_β} | -0.07359 |
| C_{D_α} | 0.2108 | C_{l_β} | -0.02854 |
| C_{m_α} | -0.5675 | C_{n_β} | -0.00040 |
| C_{L_q} | 2.8932 | C_{Y_p} | 0 |
| C_{D_q} | 0 | C_{l_p} | -0.3209 |
| C_{m_q} | -1.3990 | C_{n_p} | -0.01297 |
| $C_{L_{\delta_e}}$ | 0.2724 | C_{Y_r} | 0 |
| $C_{D_{\delta_e}}$ | 0.3045 | C_{l_r} | 0.03066 |
| $C_{m_{\delta_e}}$ | -0.3254 | C_{n_r} | -0.00434 |
| C_{prop} | 1.0 | $C_{Y_{\delta_a}}$ | 0 |
| M | 50 | $C_{l_{\delta_a}}$ | 0.1682 |
| α_0 | 0.4712 | C_{n_a} | 0.1682 |
| ϵ | 0.1592 | | |
| C_{D_p} | 0.0254 | | |

4.2 State Space Matrix

The state space matrix obtained from MATLAB is given as:

Longitudinal Mode:

$$A_{lon} = \begin{pmatrix} -0.454653673118648 & 0.0917411892577683 & -0.864212811454343 & -8.98777814898664 & 0 \\ -1.06376723875552 & -4.90784812684167 & 12.9712426627808 & -3.73402061060811 & 0 \\ 0.360792817681557 & -4.63539544994353 & -7.27172280590767 & 0 & 0 \\ 0 & 0 & 0.949792954946042 & 0 & 0 \\ 0.400754824687156 & -0.870186377875349 & 0 & 12.2130111947080 & 0 \end{pmatrix}$$

$$B_{lon} = \begin{pmatrix} -0.504983700194440 & 8.63929252197672 \\ 6.38262550062489 & 0 \\ -79.5245276238215 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

Lateral Mode:

$$A_{lat} = \begin{pmatrix} -1.34070933501494 & 0.772363528094415 & -12.9595017505407 & 9.29902987648858 & 0 \\ -2.06971828950236 & -3.27412887764330 & 1.62740860315965 & 0 & 0 \\ 4.40544321372776 & 0.260851137160438 & -4.39246644894759 & 0 & 0 \\ 0 & 1.00000000000001 & 0.0714932279410995 & -1.31851562471148e-13 & 0 \\ 0 & 0 & 0.953275152609940 & 0 & 0 \end{pmatrix}$$

$$B_{lat} = \begin{pmatrix} 0 & -3.02343635753036 \\ 18.6251312139587 & 24.1095346723967 \\ 14.0750177252039 & -7.05874272555448 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

4.3 Path following

In the simulation, we could see the behaviour of the aircraft moving in the inertial frame while following the waypoint and orbit. For the waypoint as shown in figure 4.3, we have several different values of the path constant(k_{path}). We start with the value $k_{path} = 0.01$ where it can be seen that the aircraft does not follow the path but tries to orient in the direction. With the value of $k_{path} = 0.05$, we could see there is an improvement in following the path but, during the curve we will have higher deviation. Note that this error decreases as the value increases to $k_{path} = 0.1$. However, for values from 0.05 to 0.1 the aircraft does not touch the waypoint 1 and waypoint 4. Notice that if we increase the $k_{path} = 0.5$ we would see very good behaviour at first while tracking the waypoint 1 till the waypoint 3, but during the waypoint 4 there is an oscillation that soon converges to the path. For the value $k_{path} = 2.0$ we see higher oscillation from starting and would also end with oscillation. This is due to large values of k_{path} results in short, abrupt transitions, while small values of k_{path} cause long, smooth transitions in the desired course. Further we also have much higher values of the actuator and more higher changes in the state associated with the higher values of the path constant.

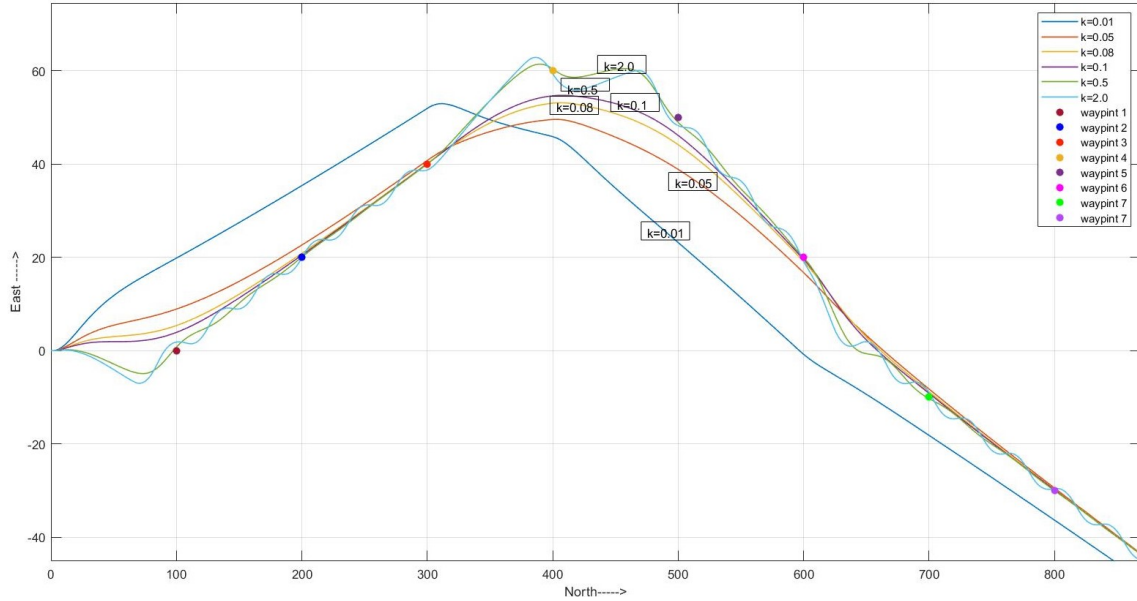


Figure 4.3: Waypoint tracking with different values of k_{path}

Figure 4.4 shows the orbit following for different values of k_{orbit} . We compare the behaviour of the aircraft and the error associated to follow the given orbit with the value of $k_{orbit} = 0.1$ and $k_{orbit} = 2.0$, where the initial behaviour of the aircraft is almost the same but the final locus of the position is different. In general, we can conclude stating that higher value of the k_{orbit} brings low error. Figure 4.5 shows the velocity behaviour of the aircraft and the figure 4.6 shows the altitude. During the initial entry to the orbit we can see the fluctuation due to the nonlinear behaviour of the aircraft but later it is constant. This behaviour brings the change in the control

signal as shown in the figure 4.9 and figure 4.10.

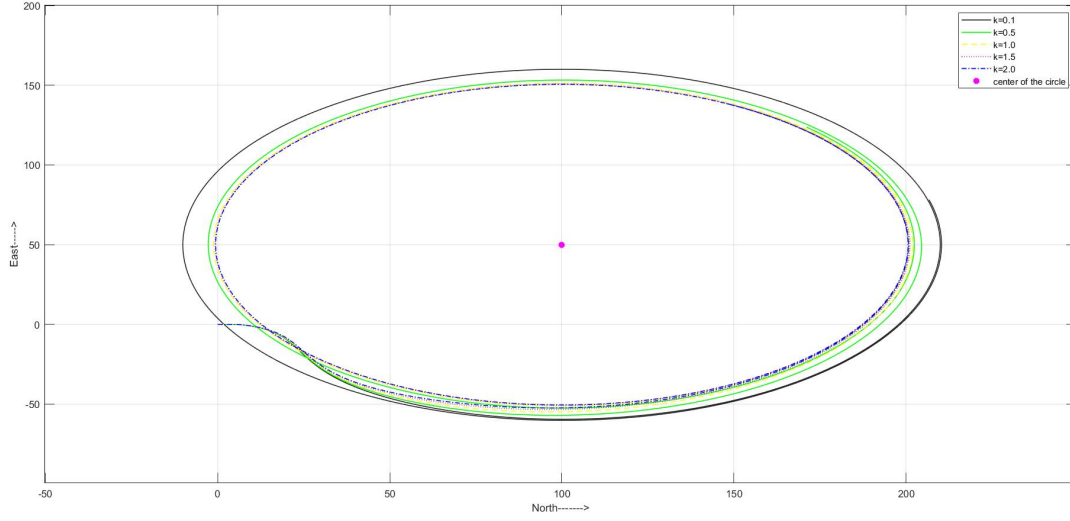


Figure 4.4: Orbit tracking with different values of k_{orbit}

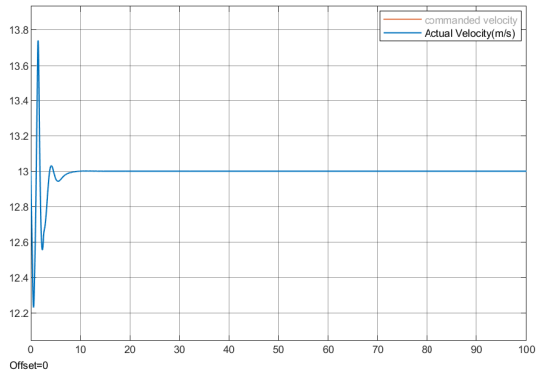


Figure 4.5: Velocity(m/s) vs time(s) for the value of orbit constant $k_{orbit}=2$

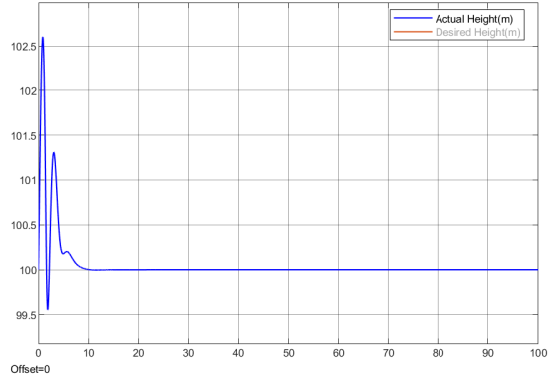


Figure 4.6: Altitude(m) vs time(s) for the value of orbit constant $k_{orbit}=2$

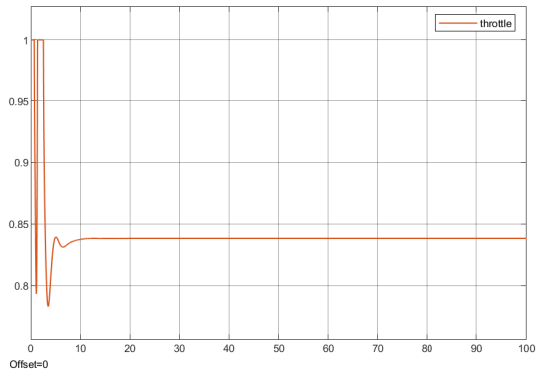


Figure 4.7: Throttle(δ_t) vs time(s) for the value of orbit constant $k_{orbit}=2$

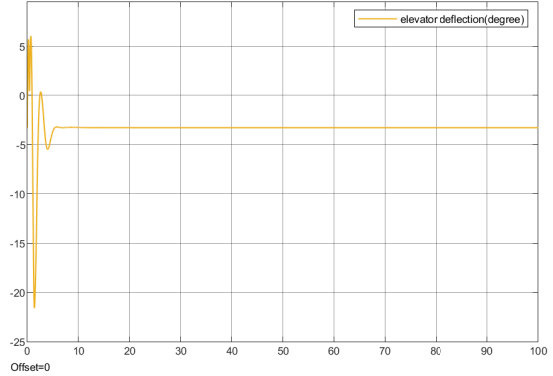


Figure 4.8: Elevator deflection(δ_e)(in degree) vs time(s) for the value of orbit constant $k_{orbit}=2$

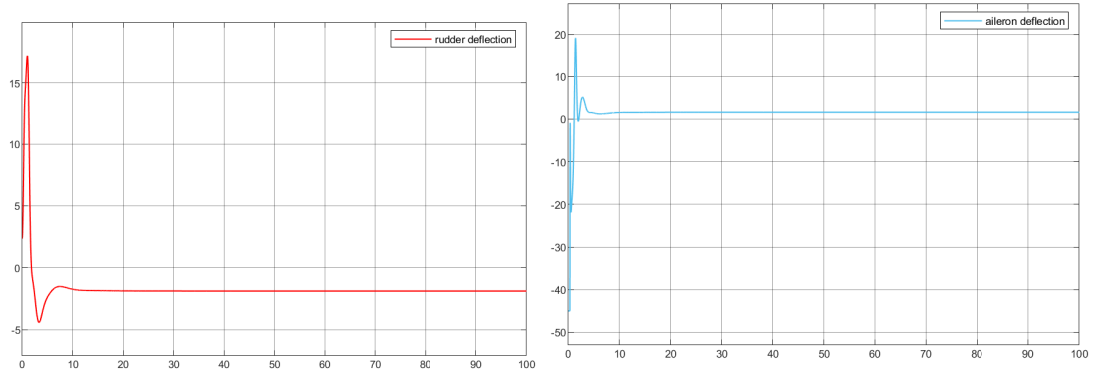


Figure 4.9: Rudder deflection(δ_r) (in degree) vs time(s) for value of orbit constant $k_{orbit}=2$ Figure 4.10: Aileron deflection(δ_a)(in degree) vs time(s) for the value of orbit constant $k_{orbit}=2$

The figure 4.11 shows the waypoint tracking in the inertial frame with $k_{path} = 0.1$, starting from the initial coordinate[0 0 100]. We could see the altitude and velocity behaviour of the aircraft to track the given waypoint in the figure 4.12 and 4.13 respectively. Figure 4.15 shows the pitch angle behaviour and figure 4.14 shows the course angle. As the vector field guidance strategy creates straight line between the waypoints which the aircraft is commanded to follow. This strategy shows during the change in the altitude the aircraft nonlinear behaviour adds up and thus there is slight fall in altitude before the climb.

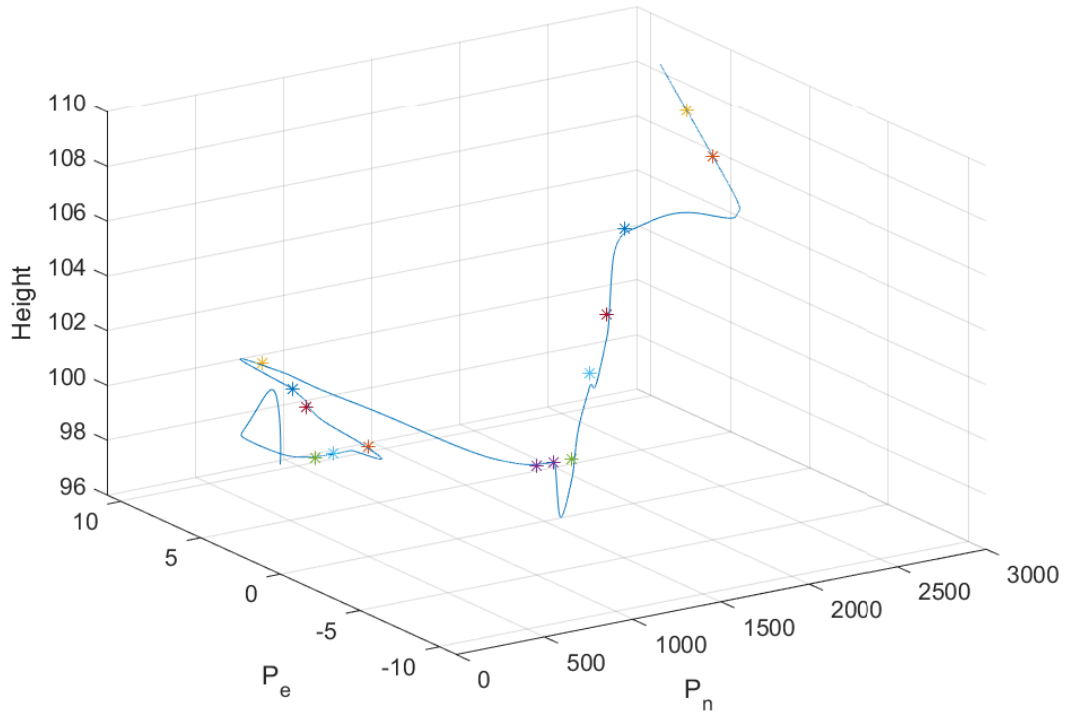


Figure 4.11: Waypoint following in the inertial frame with $k_{path}=0.1$

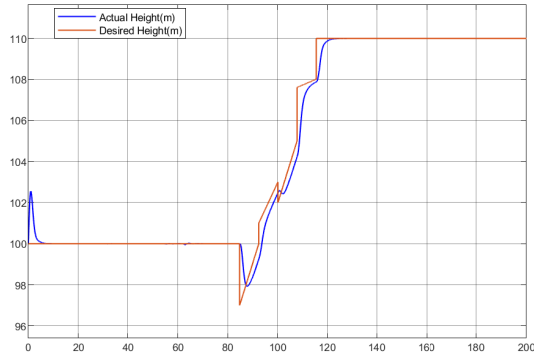


Figure 4.12: Altitude(m) vs time(s) to follow the given waypoint, $k_{path}=0.1$

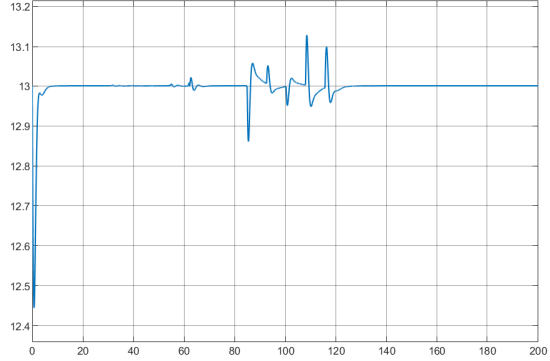


Figure 4.13: Velocity(m/s) vs time(s) to follow the given waypoint $k_{path}=0.1$

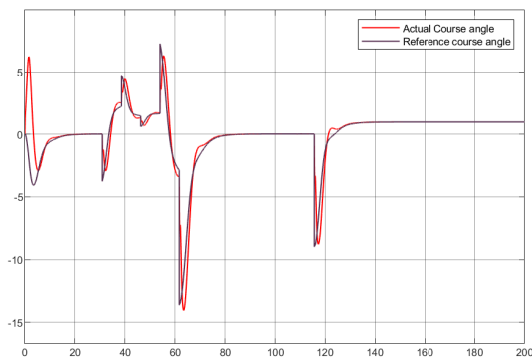


Figure 4.14: Course angle(χ) (in degree) vs time(s) to follow the given waypoint, $k_{path}=0.1$

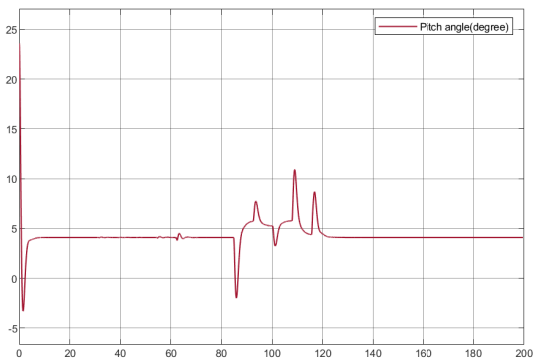


Figure 4.15: Pitch angle(θ) (in degree) vs time(s) to follow the given waypoint, $k_{path}=0.1$

Chapter 5

Conclusion

In this thesis, we studied the modelling and control of a class of fixed wing Unmanned Aerial Vehicles (UAVs). The work was structured into various chapters. The first chapter was dedicated to the mathematical modelling of the aircraft where we started with the introduction of the several reference frames associated with the aircraft. In the kinematics modelling, we studied the relation between the velocities (linear and angular) and the position and orientation. The kinematic equations were divided into translation and rotational kinematics. After the kinematics we moved to the dynamic equation, that captures the translations dynamic and rotational dynamic as a function of force and moments, resulting in 12 nonlinear differential equations with 6 Degree of freedom along with the control input.

The aircraft is a nonlinear model, so we first proceed with its linearization using the trim condition. The trim condition is available for different flight envelopes. Based on the cruise case where the aircraft is at a constant altitude, we formulated a linear model as in section 2.11. To make the design easy we decoupled the longitudinal and lateral modes. The longitudinal autopilot consisted of the height and velocity controller and the lateral autopilot consisted of the heading controller assuming the heading(ψ) angle is equal to the course angle(χ). LQR control strategy was used to design the autopilot. Next, we devoted to the design of the path following algorithm with the waypoint following (section 3.3.2) and the orbit following(section 3.3.3). The main goal is to make the UAV to be able to autonomously follow a predefined path. Path following algorithms ensure that the UAV will follow the predefined path in two or three dimensions. A requirement for these path-following algorithms is that they must be accurate and robust to wind disturbances. Control techniques are popular for path-following applications. They provide the robustness to wind disturbances mention before. There are several control techniques applicable in the UAV including linear control, nonlinear control, intelligent control, hybrid control and robust control. In section 3.3.2, we used the vector field method and thus simulated the behavior of the aircraft with different values.

The nonlinear behavior of the aircraft had a direct effect on the control signals, namely the fact if the actuators saturate one would easily stall the aircraft. In this thesis, we illustrated the performance of the standard vector field method of path following that had higher efficiency for the waypoints that were further compared to the waypoints that were closer. Since knowledge of the plant model is one of the most important parts for control design, the method used proved to be efficient for modelling that uses Simulink to linearize which is easy to implement and allows

us to deal with any form of small fixed-wing UAV. The future aspect of the thesis is to extend the work to test and compare different control algorithm based on performance and simplicity. For the future task, we will use the plant model to design the outer loop nonlinear guidance controller for path following and object tracking (moving and stationary). This type of control algorithms seem promise as they can deal with the error created in the inner loop. This thesis would also be extended to create a hardware in the loop simulation using the nonlinear model.

Appendix A

Appendix

Table A.1: Longitudinal coefficient equation

| Longitudinal | Formula |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| X_u | $\frac{u^* \rho S}{m} [C_{X_0} + C_{X_\alpha} \alpha^* + C_{X_{\delta_e}} \delta_e^*] - \frac{\rho S w^* C_{X_\alpha}}{2m} + \frac{\rho S c C_{X_q} u^* q^*}{4m V_a^*} - \frac{\rho S_{prop} C_{prop} u^*}{m}$ |
| X_w | $-q^* + \frac{w^* \rho S}{m} [C_{X_0} + C_{X_\alpha} \alpha^* + C_{X_{\delta_e}} \delta_e^*] + \frac{\rho S c C_{X_q} w^* q^*}{4m V_a^*} + \frac{\rho S C_{X_\alpha} u^*}{2m} - \frac{\rho S_{prop} C_{prop} w^*}{m}$ |
| X_q | $-w^* + \frac{\rho V_a^* S C_{X_q} c}{4m}$ |
| X_{δ_e} | $\frac{\rho V_a^{*2} S C_{X_{\delta_e}}}{2m}$ |
| X_{δ_t} | $\frac{\rho S_{prop} C_{prop} k^2 \delta_t^*}{m}$ |
| Z_u | $q^* + \frac{u^* \rho S}{m} [C_{Z_0} + C_{Z_\alpha} \alpha^* + C_{Z_{\delta_e}} \delta_e^*] - \frac{\rho S C_{Z_\alpha} w^*}{2m} + \frac{u^* \rho S C_{Z_q} c q^*}{4m V_a^*}$ |
| Z_w | $\frac{w^* \rho S}{m} [C_{Z_0} + C_{Z_\alpha} \alpha^* + C_{Z_{\delta_e}} \delta_e^*] + \frac{\rho S C_{Z_\alpha} u^*}{2m} + \frac{\rho w^* S c C_{Z_q} q^*}{4m V_a^*}$ |
| Z_q | $u^* + \frac{\rho V_a^* S C_{Z_q} c}{4m}$ |
| Z_{δ_e} | $\frac{\rho V_a^{*2} S C_{Z_{\delta_e}}}{2m}$ |
| M_u | $\frac{u^* \rho S c}{J_y} [C_{m_0} + C_{m_\alpha} \alpha^* + C_{m_{\delta_e}} \delta_e^*] - \frac{\rho S c C_{m_\alpha} w^*}{2J_y} + \frac{\rho S c^2 C_{m_q} q^* u^*}{4J_y V_a^*}$ |
| M_w | $\frac{w^* \rho S c}{J_y} [C_{m_0} + C_{m_\alpha} \alpha^* + C_{m_{\delta_e}} \delta_e^*] + \frac{\rho S c C_{m_\alpha} u^*}{2J_y} + \frac{\rho S c^2 C_{m_q} q^* w^*}{4J_y V_a^*}$ |
| M_q | $\frac{\rho V_a^* S c^2 C_{m_q}}{4J_y}$ |
| M_{δ_e} | $\frac{\rho V_a^{*2} S c C_{m_{\delta_e}}}{2J_y}$ |

Table A.2: Lateral coefficient equation

| Lateral | Formula |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Y_v | $\frac{\rho S b v^*}{4m V_a^*} [C_{Y_p} p^* + C_{Y_r} r^*] + \frac{\rho S v^*}{m} [C_{Y_0} + C_{Y_\beta} \beta^* + C_{Y_{\delta_a}} \delta_a^* + C_{Y_{\delta_r}} \delta_r^*] + \frac{\rho S C_{Y_\beta}}{2m} \sqrt{u^{*2} + w^{*2}}$ |
| Y_p | $w^* + \frac{\rho V_a^* S b}{4m} C_{Y_p}$ |
| Y_r | $-u^* + \frac{\rho V_a^* S b}{4m} C_{Y_r}$ |
| Y_{δ_a} | $\frac{\rho V_a^{*2} S}{2m} C_{Y_{\delta_a}}$ |
| Y_{δ_r} | $\frac{\rho V_a^{*2} S}{2m} C_{Y_{\delta_r}}$ |
| L_v | $\frac{\rho S b^2 v^*}{4V_a^*} [C_{p_p} p^* + C_{p_r} r^*] + \rho S b v^* [C_{p_0} + C_{p_\beta} \beta^* + C_{p_{\delta_a}} \delta_a^* + C_{p_{\delta_r}} \delta_r^*] + \frac{\rho S b C_{p_\beta}}{2} \sqrt{u^{*2} + w^{*2}}$ |
| L_p | $\Gamma_1 q^* + \frac{\rho V_a^* S b^2}{4} C_{p_p}$ |
| L_r | $-\Gamma_2 q^* + \frac{\rho V_a^* S b^2}{4} C_{p_r}$ |
| L_{δ_a} | $\frac{\rho V_a^{*2} S b}{2} C_{p_{\delta_a}}$ |
| L_{δ_r} | $\frac{\rho V_a^{*2} S b}{2} C_{p_{\delta_r}}$ |
| N_v | $\frac{\rho S b^2 v^*}{4V_a^*} [C_{r_p} p^* + C_{r_r} r^*] + \rho S b v^* [C_{r_0} + C_{r_\beta} \beta^* + C_{r_{\delta_a}} \delta_a^* + C_{r_{\delta_r}} \delta_r^*] + \frac{\rho S b C_{r_\beta}}{2} \sqrt{u^{*2} + w^{*2}}$ |
| N_p | $\Gamma_7 q^* + \frac{\rho V_a^* S b^2}{4} C_{r_p}$ |
| N_r | $-\Gamma_1 q^* + \frac{\rho V_a^* S b^2}{4} C_{r_r}$ |
| N_{δ_a} | $\frac{\rho V_a^{*2} S b}{2} C_{r_{\delta_a}}$ |
| N_{δ_r} | $\frac{\rho V_a^{*2} S b}{2} C_{r_{\delta_r}}$ |

Bibliography

- [1] A Pedro Aguiar and Joao P Hespanha. Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty. *IEEE transactions on automatic control*, 52(8):1362–1379, 2007.
- [2] R.W. Beard and T.W. McLain. Small unmanned aircraft: Theory and practice, 2012.
- [3] Basilio Bona. Dynamic modelling of mechatronic system, 2013.
- [4] Stephen A Cambone, Keinheth J Krieg, Peter Pace, and Linton Wells. Usa’s unmanned aircraft roadmap, 2005-2030. *National Defense*, 2005.
- [5] Haiyang Chao, Yongcan Cao, and YangQuan Chen. Autopilots for small fixed-wing unmanned air vehicles: A survey. In *2007 International Conference on Mechatronics and Automation*, pages 3144–3149. IEEE, 2007.
- [6] MV Cook. Flight dynamics principles, 2007.
- [7] Martin E Dempsey and S Rasmussen. Eyes of the army—us army roadmap for unmanned aircraft systems 2010–2035. *US Army UAS Center of Excellence, Ft. Rucker, Alabama*, 9, 2010.
- [8] André Deperrois. About stability analysis using xflr5. *Presentation document, a. http://xflr5.sourceforge.net/docs/XFLR5_and_Stability_analysis.pdf*, 2010.
- [9] Thor I Fossen. Mathematical models for control of aircraft and satellites. *Department of Engineering Cybernetics Norwegian University of Science and Technology*, 2011.
- [10] Hassan K Khalil and Jessy W Grizzle. *Nonlinear systems*, volume 3. Prentice hall Upper Saddle River, NJ, 2002.
- [11] Zixuan Liang, Qingdong Li, and Zhang Ren. Waypoint constrained guidance for entry vehicles. *Aerospace Science and Technology*, 52:52–61, 2016.
- [12] Derek R Nelson, D Blake Barber, Timothy W McLain, and Randal W Beard. Vector field path following for miniature air vehicles. *IEEE Transactions on Robotics*, 23(3):519–529, 2007.
- [13] Robert C Nelson et al. *Flight stability and automatic control*, volume 2. WCB/McGraw Hill New York, 1998.
- [14] Sanghyuk Park, John Deyst, and Jonathan How. A new nonlinear guidance logic for trajectory tracking. In *AIAA guidance, navigation, and control conference and exhibit*, page 4900, 2004.

- [15] PB Sujit, Srikanth Saripalli, and Joao Borges Sousa. Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicles. *IEEE Control Systems Magazine*, 34(1):42–59, 2014.
- [16] Jeffrey M Sullivan. Evolution or revolution? the rise of uavs. *IEEE Technology and Society Magazine*, 25(3):43–49, 2006.
- [17] Antonios Tsourdos, Brian White, and Madhavan Shanmugavel. *Cooperative path planning of unmanned aerial vehicles*, volume 32. John Wiley & Sons, 2010.
- [18] VAYU. <https://www.vayu.us/about-us/>, 2019 (accessed 27 June, 2019).
- [19] Zagi. <http://www.zagi.com>, 2019 (accessed 27 June, 2019).
- [20] ZIPLINE. <https://flyzipline.com/about/>, 2019 (accessed 27 June, 2019).