Chapter 1

Introduction

1.1 System Architecture

The objective of this book is to prepare the reader to do research in the exciting and rapidly developing field of autonomous navigation, guidance, and control of unmanned air vehicles. The focus is on the design of the software algorithms required for autonomous and semi-autonomous flight. To work in this area, researchers must be familiar with a wide range of topics, including coordinate transformations, aerodynamics, autopilot design, state estimation, path planning, and computer vision. The aim of this book is to cover these essential topics, focusing in particular on their application to small and miniature air vehicles, which we denote by the acronym MAV.

In the development of the topics, we have in mind the software architecture shown in Figure 1.1. The block labeled *unmanned aircraft* in Figure 1.1 is the six-degree-of-freedom (DOF) physical aircraft that responds to servo command inputs (elevator, aileron, rudder, and throttle) and wind and other disturbances. The mathematical models required to understand fixed-wing flight are complicated and are covered in Chapters 2 to 5 and Chapter 9. In particular, in Chapter 2 we discuss coordinate frames and transformations between frames. A study of coordinate frames is required since most specifications for MAVs are given in the inertial frame (e.g., orbit a specific coordinate), whereas most of the sensor measurements are with respect to the body frame, and the actuators exert forces and torques in the body frame. In Chapter 3 we develop the basic rigid-body kinematics and dynamics of an aircraft. In Chapter 4 we describe the aerodynamic forces and moments that act on fixed-wing aircraft. Chapter 5 begins by combining the results of Chapters 3 and 4 to obtain a six-DOF, 12-state, nonlinear dynamic model for a MAV. While incorporating the fidelity desired for simulation purposes, the six-DOF model is fairly complicated and cumbersome to work with. The design and analysis of aircraft control approaches are more easily accomplished using

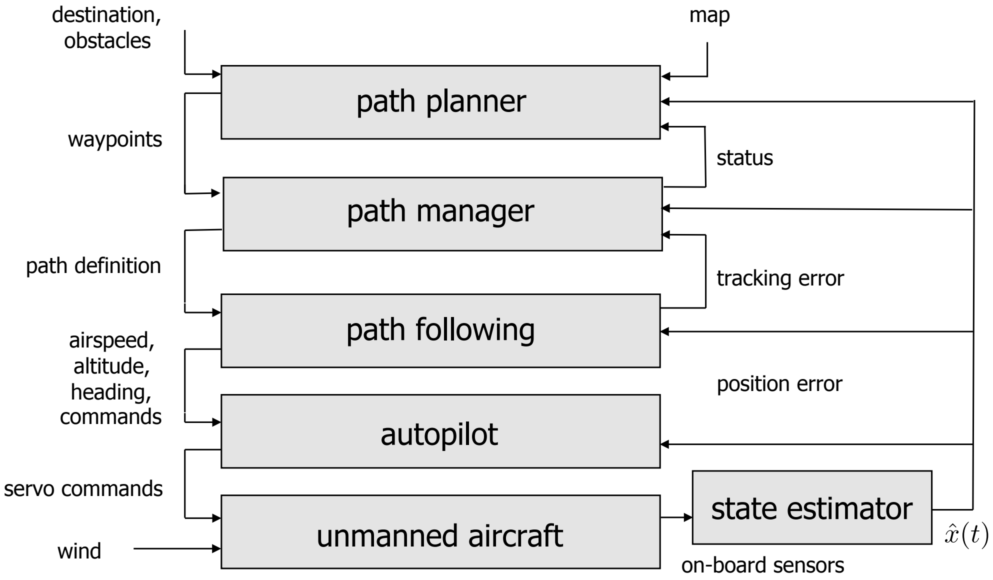


Figure 1.1: The system architecture that will be assumed throughout the book. The path planner produces straight-line or Dubins paths through obstacle fields. The path manager switches between orbit following and straight-line path following to maneuver along the waypoint paths. The path-following block produces commands to the low-level autopilot, which controls the airframe. Each of the blocks relies on estimates of the states produced by filtering the onboard sensors.

lower-order linear models. Linear models that describe small deviations from trim are derived in Chapter 5, which describes trim conditions and derives linear transfer function and state-space models for MAVs commonly used to develop aircraft control laws.

The block labeled *autopilot* in Figure 1.1 refers to the low-level control algorithms that maintain roll and pitch angles, airspeed, altitude, and course heading. Chapter 6 introduces the standard technique of successive loop closure to design the autopilot control laws. Nested control loops are closed one at a time, with inner loops maintaining roll and pitch angles and outer loops maintaining airspeed, altitude, and course.

The autopilot and the higher level blocks rely on accurate state estimates obtained by dynamically filtering the onboard sensors which include accelerometers, rate gyros, pressure sensors, magnetometers, and GPS receivers. A description of these sensors and their mathematical models is given in Chapter 7. Because it is not possible to measure all the states of small unmanned aircraft using standard sensors, state estimation plays an important role. Descriptions of several state-estimation techniques that are effective for MAVs are given in Chapter 8.

A complete model of the flight dynamics coupled with the autopilot and state estimation techniques represents a high dimensional, highly complex, nonlinear system of equations. A full model of the system is too complicated to facilitate the development of high level guidance algorithms. Therefore, Chapter 9 develops low-order nonlinear equations that model the closed-loop behavior of the system. These models are used in subsequent chapters to develop guidance algorithms.

One of the primary challenges with MAVs is flight in windy conditions. Since airspeeds in the range of 20 to 40 mph are typical for MAVs, and since wind speeds at several hundred feet above ground level (AGL) almost always exceed 10 mph, MAVs must be able to maneuver effectively in wind. Traditional trajectory tracking methods used in robotics do not work well for MAVs. The primary difficulty with these methods is the requirement to be in a particular location at a particular time, which cannot properly take into account the variations in ground speed caused by the unknown and changing effects of the wind. Alternatively, path-following methods that simply maintain the vehicle on a desired path have proven to be effective in flight tests. Chapter 10 describes the algorithms and methods used to provide the capabilities of the *path following* block in Figure 1.1. We will focus exclusively on straight-line paths and circular orbits and arcs. Other useful paths can be built up from these straight-line and circular path primitives.

The block labeled *path manager* in Figure 1.1 is a finite-state machine that converts a sequence of waypoint configurations (positions and orientations) into sequences of straight-line paths and circular arcs that can be flown by the MAV. This makes it possible to simplify the path planning problem so that the *path planner* produces either a sequence of straight-line paths that maneuver the MAV through obstacle fields, or a Dubin's path that maneuvers through the obstacle field. Chapter 11 describes the *path manager*, while Chapter 12 describes the *path planner*. For path planning we consider two classes of problems. The first class of problems is point-to-point algorithms, where the objective is to maneuver from a start position to an end position while avoiding a set of obstacles. The second class of problems is search algorithms, where the objective is to cover a region, potentially having no-go regions, with a sensor footprint.

Almost all applications involving MAVs require the use of an onboard electro-optical/infrared (EO/IR) video camera. The typical objective of the camera is to provide visual information to the end user. Since MAV payload capacities are limited, however, it makes sense to also use the video camera for navigation, guidance, and control. Effective use of camera information is currently an active research topic. In Chapter 13 we discuss several potential uses of video cameras on MAVs, including geolocation and vision-based landing. Geolocation uses a sequence of images as well as the onboard sensors to estimate the world coordinates of objects on the ground. Vision-based landing uses video images captured by the MAV to guide the MAV to a target identified in the image plane. We feel that an understanding of these problems will enable further investigations in vision-based control of MAVs.

In Chapter 13, we use the software architecture shown in Figure 1.2, where the *path planner* block has been replaced with the block *vision-based guidance*. However, the vision-based guidance laws interact with the architecture in the same manner as the path planner. The modularity of the architecture is one of its most appealing features.

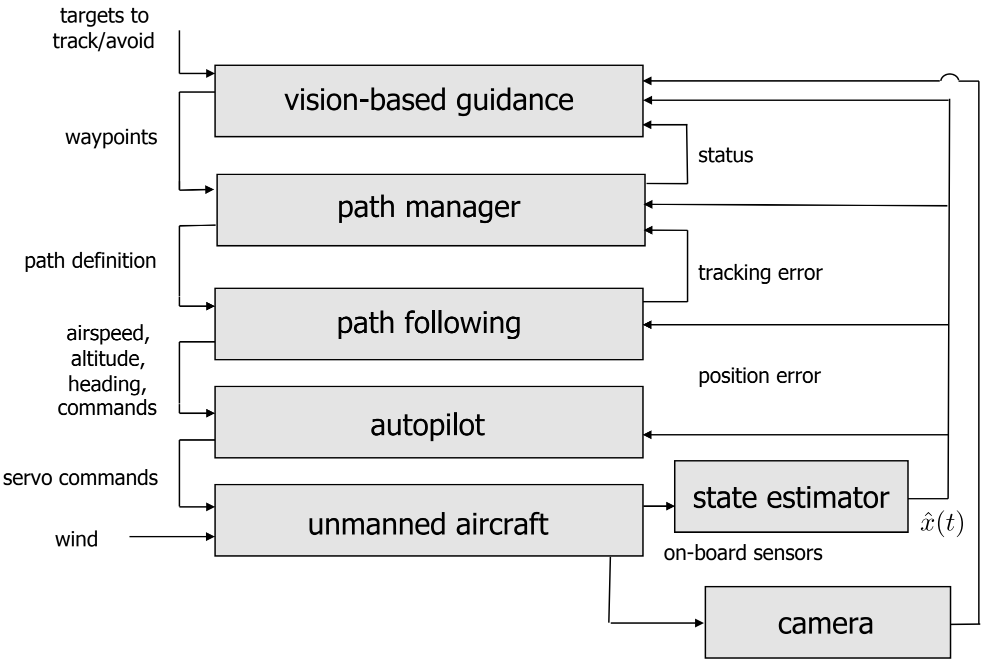


Figure 1.2: System architecture for vision-based navigation, guidance, and control. A video camera is added as an additional sensor and the path planner has been replaced with a vision-based guidance block.

1.2 Design Models

The design philosophy that we follow throughout the book is illustrated schematically in Figure 1.3. The unmanned aircraft operating in its environment is depicted in Figure 1.3 as the "Physical System," and includes the actuators (control flaps and propeller) and the sensors (IMU, GPS, camera, etc.). The first step in the design process is to model the physical system using nonlinear differential equations. While approximations and simplifications will be necessary at this step, the hope is to capture in mathematics all of the important characteristics of the physical system. In this book, the model of the physical system includes rigid body kinematics and dynamics (Chapter 3), aerodynamic forces and moments (Chapter 4), and the onboard sensors (Chapter 7). The resulting model is called the "Simulation Model" in Figure 1.3 and will be used for the high fidelity computer simulation of the physical system. However, we should note that the simulation model

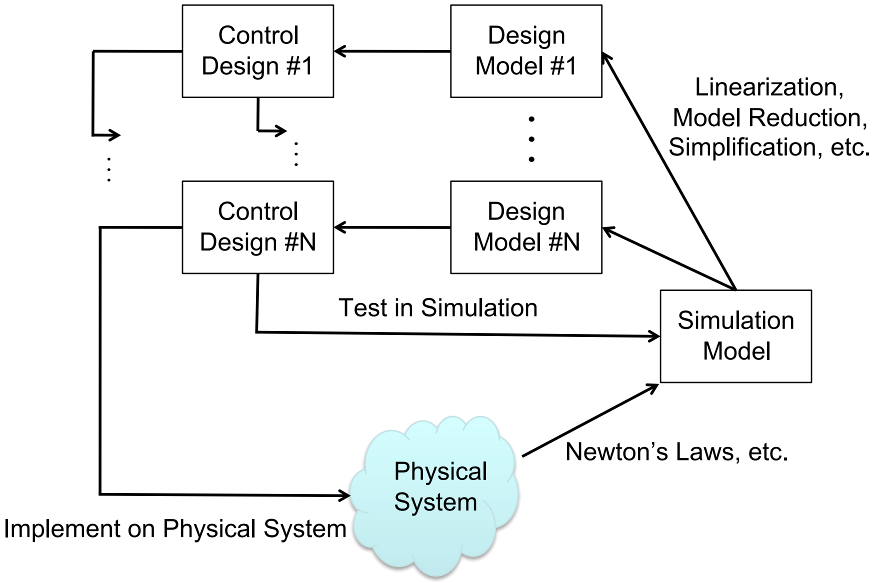


Figure 1.3: The design process. Using principles of physics, the physical system is modeled mathematically, resulting in the simulation model. The simulation model is simplified to create design models that are used for the control design. The control design is then tested and debugged in simulation and finally implemented on the physical system.

is only an approximation of the physical system, and simply because a design is effective on the simulation model, we should not assume that it will function properly on the physical system.

The simulation model is typically nonlinear and high order and is too mathematically complex to be useful for control design. Therefore, to facilitate design, the simulation model is simplified and usually linearized to create lower-order design models. For any physical system, there may be multiple design models that capture certain aspects of the design process. For MAVs, we will use a variety of different design models for both low-level control and also for high-level guidance. In Chapter 5, we will decompose the aircraft motion into longitudinal (pitching and climbing) motion and lateral (rolling and heading) motion, and we will have different design models for each type of motion. The linear design models developed in Chapter 5 will be used in Chapter 6 to develop low-level autopilot loops that control the airspeed, altitude, and course angle of the vehicle. In Chapter 8, we show how to estimate the states needed for the autopilot loops using sensors typically found on small and micro air vehicles.

The mathematical equations describing the physics of the system, the low-level autopilot, and the state estimation routines, when considered as a whole, are very complex and are not useful for designing the higher level guidance routines. Therefore, in Chapter 9 we develop nonlinear design models that model the closed-loop behavior of the system, where the input is commanded airspeed, altitude, and course angle, and the outputs are the inertial position and orientation of the aircraft. The design models developed in Chapter 9 are used in Chapters 10 through 13 to develop guidance strategies for the MAV.

As shown in Figure 1.3, the design models are used to design the guidance and control systems. The designs are then tested against the high fidelity simulation model, which sometimes requires that the design models be modified or enhanced if they have not captured the essential features of the system. After the designs have been thoroughly tested against the simulation model, they are implemented on the physical system and are again tested and debugged, sometimes requiring that the simulation model be modified to more closely match the physical system.

1.3 Design Project

In this textbook we have decided to replace traditional pencil-and-paper homework problems with a complete and rather extensive design project. The design project is an integral part of the book, and we believe that it will play a significant role in helping the reader to internalize the material that is presented.

The design project involves building a MAV flight simulator from the ground up. The flight simulator will be built using Matlab/Simulink, and we have specifically designed the assignments so that additional add-on packages are not required.[1] The website for the book contains a number of different Matlab and Simulink files that will assist you in developing the flight simulator. Our strategy is to provide you with the basic skeleton files that pass the right information between blocks, but to have you write the internal workings of each block. The project builds upon itself and requires the successful completion of each chapter before it is possible to move to the next chapter. To help you know when the design from each chapter is working, we have included graphs and pictures on the website that show the output of our simulator at each stage.

The project assignment in Chapter 2 is to develop an animation of an aircraft and to ensure that you can properly rotate the body of the aircraft on the screen. A tutorial on animating graphics in Matlab is provided in Appendix B. The assignment in Chapter 3 is to drive the animation using a mathematical model of the rigid body equations of motion. In Chapter 4 the force and moments acting on a fixed wing aircraft are added to the simulation. The assignment in Chapter 5 is to use the Simulink commands `trim` and `linmod` to find the trim conditions of the aircraft and to derive linear transfer function and state space models of the simulation. The assignment in

[1] We have also taught the course using the public domain flight simulator Aviones, which is available for download at Sourceforge.net. For those who do not have access to Matlab/Simulink and would prefer to develop the project in C/C++, we encourage the use of *Aviones.*

Chapter 6 adds an autopilot block that uses the real states to control the aircraft. In Chapter 7, a model of the sensors is added to the simulator, and in Chapter 8, state estimation schemes are added to estimate the states needed for the autopilot using the available sensors. The result of the project assignment in Chapter 8 is a closed-loop system that controls airspeed, altitude, and course angle using only available sensor information. The assignment in Chapter 9 is to approximate the closed-loop behavior using simple design models and to tune the parameters of the design model so that it essentially matches the behavior of the closed-loop high-fidelity simulation. The assignment in Chapter 10 is to develop simple guidance algorithms for following straight-lines and circular orbits in the presence of wind. In Chapter 11, straight-line and orbit following are used to synthesize more complicated paths, with emphasis on following Dubins paths. The assignment in Chapter 12 is to implement the RRT path planning scheme to plan Dubins paths through an obstacle field. The project assignment in Chapter 13 is to point a camera at a moving ground target and to estimate the inertial position of the target using camera data and onboard sensors (geolocation).

Chapter 2

Coordinate Frames

In studying unmanned aircraft systems, it is important to understand how different bodies are oriented relative to each other. Most obviously, we need to understand how the aircraft is oriented with respect to the earth. We may also want to know how a sensor (e.g., a camera) is oriented relative to the aircraft or how an antenna is oriented relative to a signal source on the ground. This chapter describes the various coordinate systems used to describe the position and orientation of the aircraft and its sensors, and the transformation between these coordinate systems. It is necessary to use several different coordinate systems for the following reasons:

- 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.

One coordinate frame is transformed into another through two basic operations: rotation and translation. Section 2.1 describes rotation matrices and their use in transforming between coordinate frames. Section 2.2 describes the specific coordinate frames used for miniature air vehicle systems. In Section 2.3 we define airspeed, ground speed, and wind speed and the relationship between these quantities. This leads to the more detailed discussion of the wind triangle in Section 2.4. In Section 2.5 we derive an expression for differentiating a vector in a rotating and translating frame.

2.1 Rotation Matrices

We begin by considering the two coordinate frames shown in Figure 2.1. The vector $\mathbf{p}$ can be expressed in both the $\mathcal{F}^0$ frame (specified by $(\mathbf{i}^0, \mathbf{j}^0, \mathbf{k}^0)$) and in the $\mathcal{F}^1$ frame (specified by $(\mathbf{i}^1, \mathbf{j}^1, \mathbf{k}^1)$). In the $\mathcal{F}^0$ frame we have

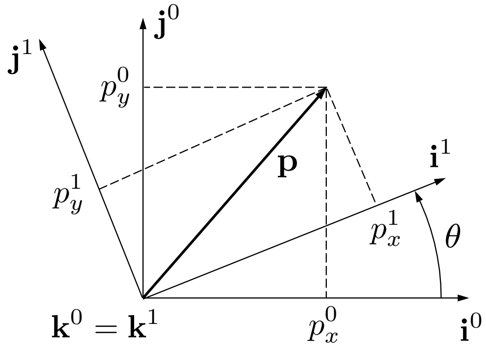


Figure 2.1: Rotation in 2D

$$\mathbf{p} = p_x^0 \mathbf{i}^0 + p_y^0 \mathbf{j}^0 + p_z^0 \mathbf{k}^0.$$

Alternatively in the $\mathcal{F}^1$ frame we have

$$\mathbf{p} = p_x^1 \mathbf{i}^1 + p_y^1 \mathbf{j}^1 + p_z^1 \mathbf{k}^1.$$

The vector sets $(\mathbf{i}^0, \mathbf{j}^0, \mathbf{k}^0)$ and $(\mathbf{i}^1, \mathbf{j}^1, \mathbf{k}^1)$ are each mutually perpendicular sets of unit basis vectors.

Setting these two expressions equal to each other gives

$$p_x^1 \mathbf{i}^1 + p_y^1 \mathbf{j}^1 + p_z^1 \mathbf{k}^1 = p_x^0 \mathbf{i}^0 + p_y^0 \mathbf{j}^0 + p_z^0 \mathbf{k}^0.$$

Taking the dot product of both sides with $\mathbf{i}^1$, $\mathbf{j}^1$, and $\mathbf{k}^1$ respectively, and stacking the result into matrix form gives

$$\mathbf{p}^1 \triangleq \begin{pmatrix} p_x^1 \\ p_y^1 \\ p_z^1 \end{pmatrix} = \begin{pmatrix} \mathbf{i}^1 \cdot \mathbf{i}^0 & \mathbf{i}^1 \cdot \mathbf{j}^0 & \mathbf{i}^1 \cdot \mathbf{k}^0 \\ \mathbf{j}^1 \cdot \mathbf{i}^0 & \mathbf{j}^1 \cdot \mathbf{j}^0 & \mathbf{j}^1 \cdot \mathbf{k}^0 \\ \mathbf{k}^1 \cdot \mathbf{i}^0 & \mathbf{k}^1 \cdot \mathbf{j}^0 & \mathbf{k}^1 \cdot \mathbf{k}^0 \end{pmatrix} \begin{pmatrix} p_x^0 \\ p_y^0 \\ p_z^0 \end{pmatrix}.$$

From the geometry of Figure 2.1 we get

$$\mathbf{p}^1 = \mathcal{R}_0^1 \mathbf{p}^0, \tag{2.1}$$

where

$$\mathcal{R}_0^1 \triangleq \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

The notation $\mathcal{R}_0^1$ is used to denote a rotation from coordinate frame $\mathcal{F}^0$ to coordinate frame $\mathcal{F}^1$.

Proceeding in a similar way, a right-handed rotation of the coordinate system about the y-axis gives

$$\mathcal{R}_0^1 \triangleq \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix},$$

and a right-handed rotation of the coordinate system about the x-axis is

$$\mathcal{R}_0^1 \triangleq \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix}.$$

As pointed out in [7], the negative sign on the sine term appears above the line with only ones and zeros.

The matrix $\mathcal{R}_0^1$ in the above equations is an example of a more general class of *orthonormal* rotation matrices that have the following properties:

P.1. $(\mathcal{R}_a^b)^{-1} = (\mathcal{R}_a^b)^T = \mathcal{R}_b^a$.

P.2. $\mathcal{R}_b^c \mathcal{R}_a^b = \mathcal{R}_a^c$.

P.3. $\det\left(\mathcal{R}_a^b\right) = 1$,

where $\det(\cdot)$ is the determinant of a matrix.

In the derivation of Equation (2.1) note that the vector $\mathbf{p}$ remains constant and the new coordinate frame $\mathcal{F}^1$ was obtained by rotating $\mathcal{F}^0$ through a *right-handed* rotation of angle θ. Alternatively, rotation matrices can be used to rotate a vector through a prescribed angle in a fixed reference frame. As an example, consider the *left-handed* rotation of a vector $\mathbf{p}$ in frame $\mathcal{F}^0$ about the $\mathbf{k}^0$-axis by the angle θ, as shown in Figure 2.2.

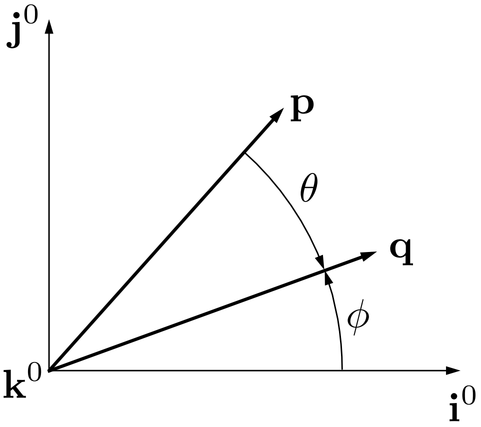


Figure 2.2: Rotation of $\mathbf{p}$ about the $\mathbf{k}^0$-axis.

Assuming $\mathbf{p}$ and $\mathbf{q}$ are confined to the $\mathbf{i}^0$-$\mathbf{j}^0$ plane, we can write the components of $\mathbf{p}$ and $\mathbf{q}$ as

$$\begin{aligned}\mathbf{p} &= \begin{pmatrix} p\cos(\theta+\phi) \\ p\sin(\theta+\phi) \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} p\cos\theta\cos\phi - p\sin\theta\sin\phi \\ p\sin\theta\cos\phi + p\cos\theta\sin\phi \\ 0 \end{pmatrix} \end{aligned} \tag{2.2}$$

and

$$\mathbf{q} = \begin{pmatrix} q\cos\phi \\ q\sin\phi \\ 0 \end{pmatrix}, \tag{2.3}$$

where $p \triangleq |\mathbf{p}| = q \triangleq |\mathbf{q}|$. Combining Equations (2.2) and (2.3) gives

$$\begin{aligned}\mathbf{p} &= \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{q} \\ &= (\mathcal{R}_0^1)^T \mathbf{q} \\ \text{and} & \\ \mathbf{q} &= \mathcal{R}_0^1 \mathbf{p}. \end{aligned}$$

In this case, the rotation matrix $\mathcal{R}_0^1$ can be interpreted as a left-handed rotation of the vector $\mathbf{p}$ through the angle θ to a new vector $\mathbf{q}$ in the same reference frame. Notice that a right-handed rotation of a vector (in this case from $\mathbf{q}$ to $\mathbf{p}$) can be obtained by using $(\mathcal{R}_0^1)^T$. This interpretation contrasts with our original use of the rotation matrix to transform a fixed vector $\mathbf{p}$ from an expression in frame $\mathcal{F}^0$ to an expression in frame $\mathcal{F}^1$ where $\mathcal{F}^1$ has been obtained from $\mathcal{F}^0$ by a right-handed rotation. In our treatment of MAVs, we will use rotation matrices primarily to rotate coordinate frames as described in the following section.

2.2 MAV Coordinate Frames

To derive and understand the dynamic behavior of MAVs, several coordinate systems are of interest. In this section, we will define and describe the following coordinate frames: the inertial frame, the vehicle frame, the vehicle-1 frame, the vehicle-2 frame, the body frame, the stability frame, and the wind frame. The inertial and vehicle frames are related by a translation, while the remaining frames are related by rotations. The angles defining the relative orientations of the vehicle, vehicle-1, vehicle-2, and body frames are the roll, pitch, and yaw angles that describe the attitude of the aircraft. These angles are commonly known as Euler angles . The rotation angles that define the relative orientation of the body, stability, and wind coordinate frames are the angle of attack and sideslip angles. Throughout the book we assume a flat, non-rotating earth — a valid assumption for MAVs.

2.2.1 The inertial frame $\mathcal{F}^i$

The inertial coordinate system is an earth-fixed coordinate system with its origin at the defined home location. As shown in Figure 2.3, the unit vector $\mathbf{i}^i$ is directed North, $\mathbf{j}^i$ is directed East, and $\mathbf{k}^i$ is directed toward the center of the earth, or down. This coordinate system is sometimes referred to as a North-East-Down reference frame. It is common for North to be referred to as the inertial x direction, East to be referred to as the inertial y direction, and down to be referred to as the inertial z direction.

2.2.2 The vehicle frame $\mathcal{F}^v$

The origin of the vehicle frame is at the center of mass of the MAV. However, the axes of $\mathcal{F}^v$ are aligned with the axis of the inertial frame $\mathcal{F}^i$. In other words, the unit vector $\mathbf{i}^v$ points North, $\mathbf{j}^v$ points East, and $\mathbf{k}^v$ points toward the center of the earth, as shown in Figure 2.4.

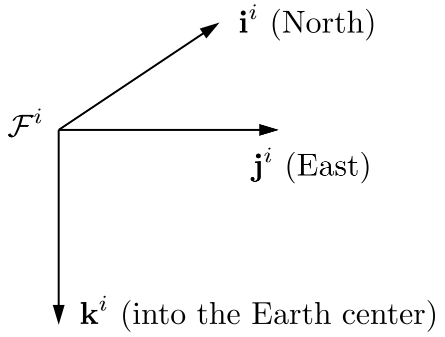


Figure 2.3: The inertial coordinate frame. The $\mathbf{i}^i$-axis points North, the $\mathbf{j}^i$-axis points East, and the $\mathbf{k}^i$-axis points into the earth.

2.2.3 The vehicle-1 frame $\mathcal{F}^{v1}$

The origin of the vehicle-1 frame is identical to the vehicle frame: the center of mass of the aircraft. However, $\mathcal{F}^{v1}$ is rotated in the positive right-handed direction about $\mathbf{k}^v$ by the heading (or yaw) angle ψ. In the absence of additional rotations, $\mathbf{i}^{v1}$ points out the nose of the airframe, $\mathbf{j}^{v1}$ points out the right wing, and $\mathbf{k}^{v1}$ is aligned with $\mathbf{k}^v$ and points into the earth. The vehicle-1 frame is shown in Figure 2.5.

The transformation from $\mathcal{F}^v$ to $\mathcal{F}^{v1}$ is given by

$$\mathbf{p}^{v1} = \mathcal{R}_v^{v1}(\psi)\mathbf{p}^v,$$

where

$$\mathcal{R}_v^{v1}(\psi) = \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

2.2.4 The vehicle-2 frame $\mathcal{F}^{v2}$

The origin of the vehicle-2 frame is again the center of mass of the aircraft and is obtained by rotating the vehicle-1 frame in a right-handed rotation about the $\mathbf{j}^{v1}$ axis by the pitch angle θ. The unit vector $\mathbf{i}^{v2}$ points out the nose of the aircraft, $\mathbf{j}^{v2}$ points out the right wing, and $\mathbf{k}^{v2}$ points out the belly, as shown in Figure 2.6.

The transformation from $\mathcal{F}^{v1}$ to $\mathcal{F}^{v2}$ is given by

$$\mathbf{p}^{v2} = \mathcal{R}_{v1}^{v2}(\theta)\mathbf{p}^{v1},$$

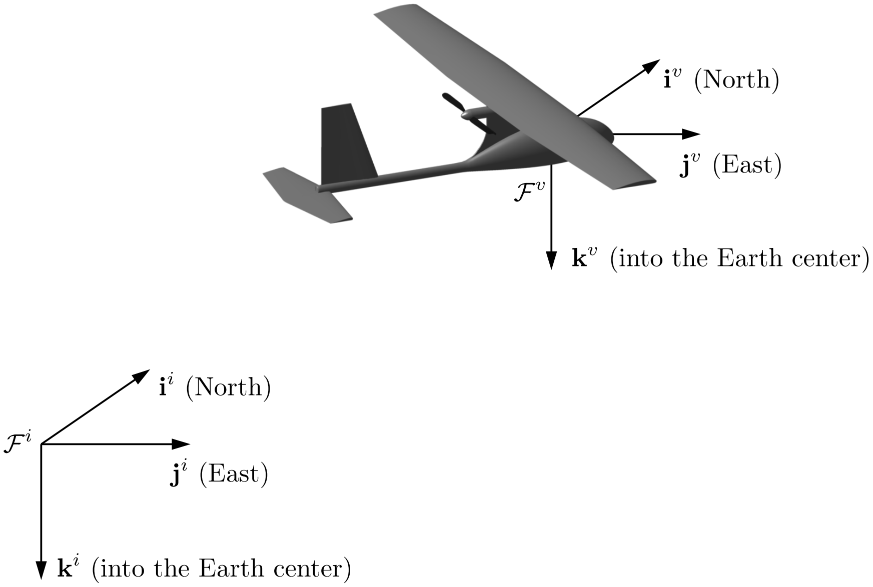


Figure 2.4: The vehicle coordinate frame. The $\mathbf{i}^v$-axis points North, the $\mathbf{j}^v$-axis points East, and the $\mathbf{k}^v$-axis points into the earth.

where

$$\mathcal{R}_{v1}^{v2}(\theta) = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix}.$$

2.2.5 The body frame $\mathcal{F}^b$

The body frame is obtained by rotating the vehicle-2 frame in a right-handed rotation about $\mathbf{i}^{v2}$ by the roll angle ϕ. Therefore, the origin is the center of mass, $\mathbf{i}^b$ points out the nose of the airframe, $\mathbf{j}^b$ points out the right wing, and $\mathbf{k}^b$ points out the belly. The body frame is shown in Figure 2.7. The directions indicated by the $\mathbf{i}^b$, $\mathbf{j}^b$, and $\mathbf{k}^b$ unit vectors are sometimes referred to as the body x, the body y, and the body z directions, respectively.

The transformation from $\mathcal{F}^{v2}$ to $\mathcal{F}^b$ is given by

$$\mathbf{p}^b = \mathcal{R}_{v2}^b(\phi)\mathbf{p}^{v2},$$

where

$$\mathcal{R}_{v2}^b(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix}.$$

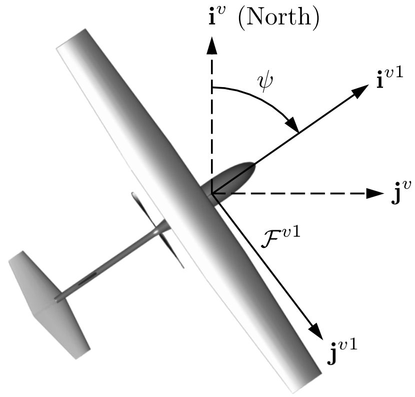


Figure 2.5: The vehicle-1 frame. The $\mathbf{i}^{v1}$-axis points out the nose of the aircraft, the $\mathbf{j}^{v1}$-axis points out the right wing, and the $\mathbf{k}^{v1}$-axis points into the earth.

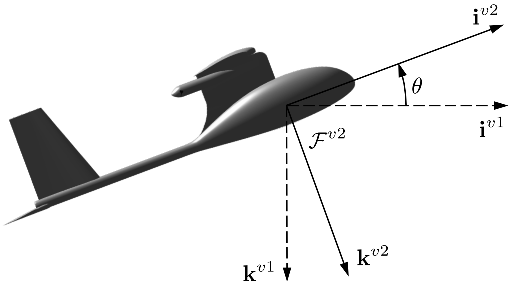


Figure 2.6: The vehicle-2 frame. The $\mathbf{i}^{v2}$-axis points out the nose of the aircraft, the $\mathbf{j}^{v2}$-axis points out the right wing, and the $\mathbf{k}^{v2}$-axis points out the belly.

The transformation from the vehicle frame to the body frame is given by

$$\begin{aligned}\mathcal{R}_v^b(\phi,\theta,\psi) &= \mathcal{R}_{v2}^b(\phi)\mathcal{R}_{v1}^{v2}(\theta)\mathcal{R}_v^{v1}(\psi) \qquad (2.4)\\ &= \begin{pmatrix}1 & 0 & 0\\ 0 & \cos\phi & \sin\phi\\ 0 & -\sin\phi & \cos\phi\end{pmatrix}\begin{pmatrix}\cos\theta & 0 & -\sin\theta\\ 0 & 1 & 0\\ \sin\theta & 0 & \cos\theta\end{pmatrix}\begin{pmatrix}\cos\psi & \sin\psi & 0\\ -\sin\psi & \cos\psi & 0\\ 0 & 0 & 1\end{pmatrix}\\ &= \begin{pmatrix}\mathrm{c}_\theta\mathrm{c}_\psi & \mathrm{c}_\theta\mathrm{s}_\psi & -\mathrm{s}_\theta\\ \mathrm{s}_\phi\mathrm{s}_\theta\mathrm{c}_\psi - \mathrm{c}_\phi\mathrm{s}_\psi & \mathrm{s}_\phi\mathrm{s}_\theta\mathrm{s}_\psi + \mathrm{c}_\phi\mathrm{c}_\psi & \mathrm{s}_\phi\mathrm{c}_\theta\\ \mathrm{c}_\phi\mathrm{s}_\theta\mathrm{c}_\psi + \mathrm{s}_\phi\mathrm{s}_\psi & \mathrm{c}_\phi\mathrm{s}_\theta\mathrm{s}_\psi - \mathrm{s}_\phi\mathrm{c}_\psi & \mathrm{c}_\phi\mathrm{c}_\theta\end{pmatrix}, \qquad (2.5)\end{aligned}$$

where $\mathrm{c}_\phi \stackrel{\triangle}{=} \cos\phi$ and $\mathrm{s}_\phi \stackrel{\triangle}{=} \sin\phi$. The angles ϕ, θ, and ψ are commonly referred to as Euler angles. The use of Euler angles is one approach for representing the orientation of a body in three-

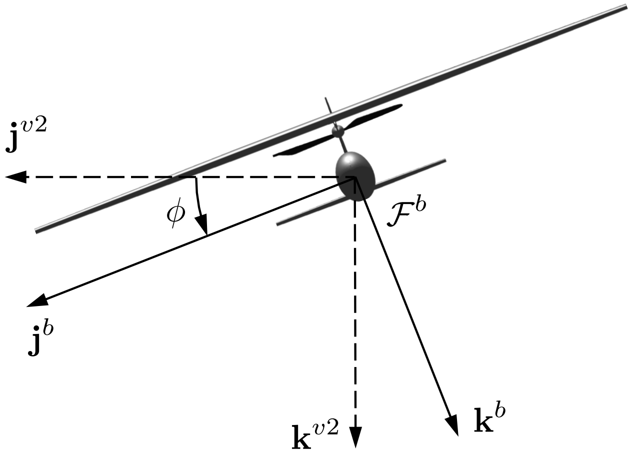


Figure 2.7: The body frame. The $\mathbf{i}^b$-axis points out the nose of the airframe, the $\mathbf{j}^b$-axis points out the right wing, and the $\mathbf{k}^b$-axis points out the belly.

dimensional space. The rotation sequence ψ-θ-ϕ is commonly used for aircraft and is just one of several Euler angle systems in use [8].

2.2.6 The stability frame $\mathcal{F}^s$

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 relative wind or the relative wind vector, denoted $\mathbf{V}_a$. The magnitude of the relative wind vector is typically 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 wind vector. This angle is called the angle of attack and is denoted by α. As shown in Figure 2.8, the angle of attack is defined as a *left*-handed rotation about $\mathbf{j}^b$ and is such that $\mathbf{i}^s$ aligns with the projection of $\mathbf{V}_a$ onto the plane spanned by $\mathbf{i}^b$ and $\mathbf{k}^b$. 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 $\mathbf{i}^s$ axis to the body frame $\mathbf{i}^b$ axis.

Since α is given by a left-handed rotation, the transformation from $\mathcal{F}^b$ to $\mathcal{F}^s$ is given by

$$\mathbf{p}^s = \mathcal{R}_b^s(\alpha)\mathbf{p}^b,$$

where

$$\mathcal{R}_b^s(\alpha) = \begin{pmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{pmatrix}.$$

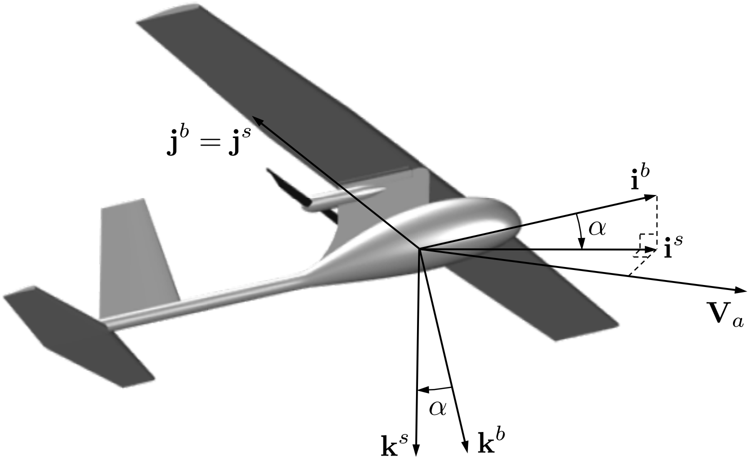


Figure 2.8: The stability frame. The $\mathbf{i}^s$-axis points along the projection of the relative wind vector onto the $\mathbf{i}^b$-$\mathbf{k}^b$ plane of the body frame, the $\mathbf{j}^s$-axis is identical to the $\mathbf{j}^b$-axis of the body frame, and the $\mathbf{k}^s$-axis is constructed to make a right-handed coordinate system. Note that the angle of attack is defined as a *left*-handed rotation about the body $\mathbf{j}^b$-axis.

2.2.7 The wind frame $\mathcal{F}^w$

The angle between the velocity vector and the $\mathbf{i}^b$-$\mathbf{k}^b$ plane is called the side-slip angle and is denoted by β. As shown in Figure 2.9, the wind frame is obtained by rotating the stability frame by a right-handed rotation of β about $\mathbf{k}^s$. The unit vector $\mathbf{i}^w$ is aligned with the relative wind vector $\mathbf{V}_a$.

The transformation from $\mathcal{F}^s$ to $\mathcal{F}^w$ is given by

$$\mathbf{p}^w = \mathcal{R}_s^w(\beta)\mathbf{p}^s,$$

where

$$\mathcal{R}_s^w(\beta) = \begin{pmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

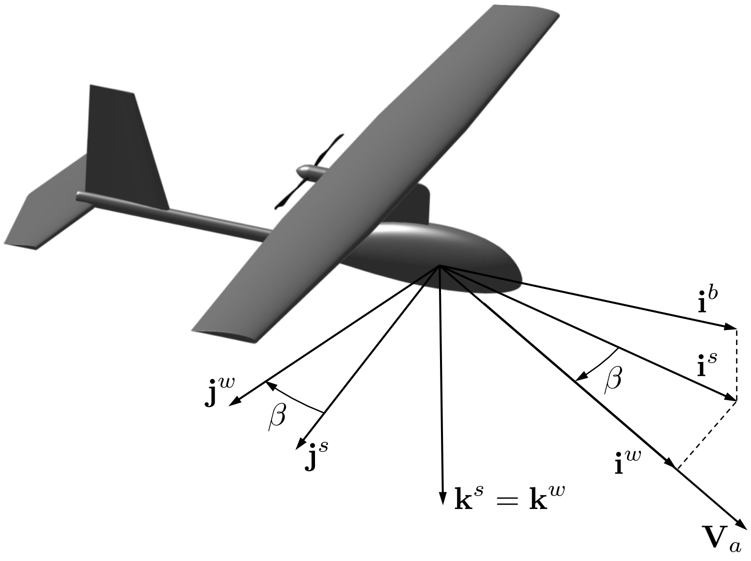


Figure 2.9: The wind frame. The $\mathbf{i}^w$-axis points along the relative wind vector.

The total transformation from the body frame to the wind frame is given by

$$\begin{aligned}
\mathcal{R}_b^w(\alpha,\beta) &= \mathcal{R}_s^w(\beta)\mathcal{R}_b^s(\alpha) \\
&= \begin{pmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{pmatrix} \\
&= \begin{pmatrix} \cos\beta\cos\alpha & \sin\beta & \cos\beta\sin\alpha \\ -\sin\beta\cos\alpha & \cos\beta & -\sin\beta\sin\alpha \\ -\sin\alpha & 0 & \cos\alpha \end{pmatrix}.
\end{aligned}$$

Alternatively, the transformation from the wind frame to the body frame is

$$\mathcal{R}_w^b(\alpha,\beta) = (\mathcal{R}_b^w)^T(\alpha,\beta) = \begin{pmatrix} \cos\beta\cos\alpha & -\sin\beta\cos\alpha & -\sin\alpha \\ \sin\beta & \cos\beta & 0 \\ \cos\beta\sin\alpha & -\sin\beta\sin\alpha & \cos\alpha \end{pmatrix}.$$

2.3 Airspeed, Wind Speed, and Ground Speed

When developing the dynamic equations of motion for a MAV, it is important to remember that the inertial forces experienced by the MAV 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 MAVs and we must carefully distinguish between airspeed, represented by the relative wind vector $\mathbf{V}_a$, and the ground speed, represented by the velocity with respect to the inertial frame $\mathbf{V}_g$. These velocities are related by the expression

$$\mathbf{V}_a = \mathbf{V}_g - \mathbf{V}_w, \tag{2.6}$$

where $\mathbf{V}_w$ is the wind velocity relative to the inertial frame.

The MAV velocity $\mathbf{V}_g$ can be expressed in the body frame in terms of components along the $\mathbf{i}^b$, $\mathbf{j}^b$, and $\mathbf{k}^b$ axes

$$\mathbf{V}_g^b = \begin{pmatrix} u \\ v \\ w \end{pmatrix},$$

where $\mathbf{V}_g^b$ is the velocity of the MAV *with respect to the inertial frame,* as expressed in the body frame. Similarly, we can write an expression for the wind velocity in the body frame as

$$\mathbf{V}_w^b = \begin{pmatrix} u_w \\ v_w \\ w_w \end{pmatrix}.$$

Keeping in mind that the relative wind vector $\mathbf{V}_a$ is the velocity of the MAV relative to the wind, it can be expressed in the wind frame as

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

Defining u_r, v_r, and w_r as the body-frame components of the relative wind vector, it can be written in the body frame as

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

When developing a MAV simulation, u_r, v_r, and w_r are used to calculate the aerodynamic forces and moments acting on the MAV. The body-frame velocity components u, v, and w are states of the MAV system and are readily available from the solution of the equations of motion. The wind velocity components u_w, v_w, and w_w typically come from a wind model as inputs to the equations of motion. Combining expressions, we can express the relative wind body-frame components in terms of the airspeed, angle of attack, and sideslip angle as

$$\begin{aligned}\mathbf{V}_a^b = \begin{pmatrix} u_r \\ v_r \\ w_r \end{pmatrix} &= \mathcal{R}_w^b \begin{pmatrix} V_a \\ 0 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} \cos\beta\cos\alpha & -\sin\beta\cos\alpha & -\sin\alpha \\ \sin\beta & \cos\beta & -\sin\beta\sin\alpha \\ \cos\beta\sin\alpha & 0 & \cos\alpha \end{pmatrix} \begin{pmatrix} V_a \\ 0 \\ 0 \end{pmatrix},\end{aligned}$$

which implies that

$$\begin{pmatrix} u_r \\ v_r \\ w_r \end{pmatrix} = V_a \begin{pmatrix} \cos\alpha\cos\beta \\ \sin\beta \\ \sin\alpha\cos\beta \end{pmatrix}. \tag{2.7}$$

Inverting this relationship gives

$$\begin{aligned} V_a &= \sqrt{u_r^2 + v_r^2 + w_r^2} \\ \alpha &= \tan^{-1}\left(\frac{w_r}{u_r}\right) \\ \beta &= \sin^{-1}\left(\frac{v_r}{\sqrt{u_r^2 + v_r^2 + w_r^2}}\right). \end{aligned} \tag{2.8}$$

Given that aerodynamic forces and moments are commonly expressed in terms of V_a, α, and β, these expressions are essential in formulating the equations of motion for a MAV.

2.4 The Wind Triangle

For MAVs, the wind speed is often in the range of 20 to 50 percent of the airspeed. The significant effect of wind on MAVs is important to understand, more so than for larger conventional aircraft, where the airspeed is typically much greater than the wind speed. Having introduced the concepts of reference frames, airframe velocity, wind velocity, and the relative wind vector, we can discuss some important definitions relating to the navigation of MAVs.

The direction of the ground speed vector relative to an inertial frame is specified using two angles. These angles are the course angle χ and the (inertial referenced) flight path angle γ. Figure 2.10 shows how these two angles are defined. The flight path angle γ is defined as the angle between the horizontal plane and the ground velocity vector $\mathbf{V}_g$, while the course χ is the angle between the projection of the ground velocity vector onto the horizontal plane and true North.

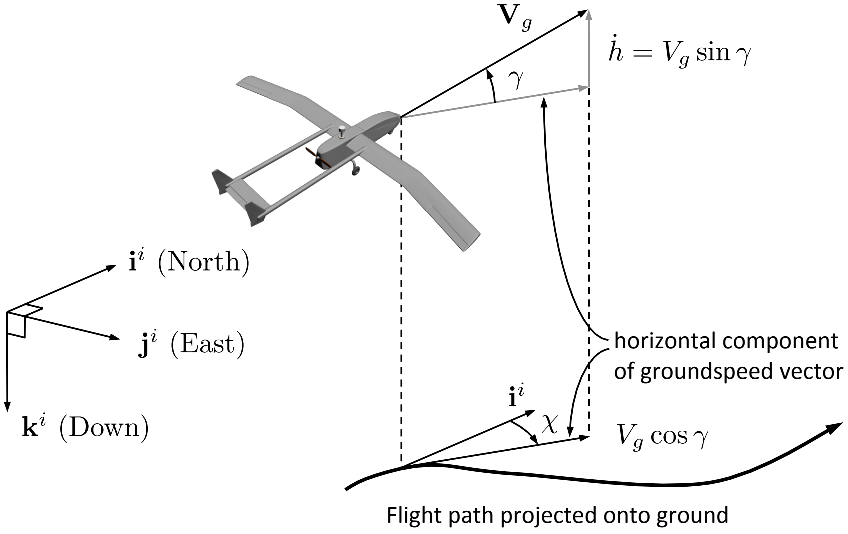


Figure 2.10: The flight path angle γ and the course angle χ.

The relationship between the groundspeed vector, the airspeed vector, and the wind vector, which is given by Equation (2.6) is called the wind triangle. A more detailed depiction of the wind triangle is given in the horizontal plane in Figure 2.11 and in the vertical plane in Figure 2.12. Figure 2.11 shows an air vehicle following a ground track represented by the dashed line. The North direction is indicated by the $\mathbf{i}^i$ vector, and the direction that the vehicle is pointed is shown by the $\mathbf{i}^b$ vector, which is fixed in the direction of the body x-axis. For level flight, the heading (yaw) angle ψ, is the angle between $\mathbf{i}^i$ and $\mathbf{i}^b$ and defines the direction the vehicle is pointed. The direction the vehicle is traveling with respect to the surrounding air mass is given by the relative wind or airspeed vector $\mathbf{V}_a$. In steady, level flight, $\mathbf{V}_a$ is commonly aligned with $\mathbf{i}^b$, meaning the sideslip angle β is zero.

The direction the vehicle is traveling with respect to the ground is shown by the velocity vector $\mathbf{V}_g$. The angle between the inertial North and the inertial velocity vector projected onto the local North-East plane is called the course angle χ. If there is a constant ambient wind, the aircraft will need to crab into the wind in order to follow a ground track that is not aligned with the wind. The *crab angle* χ_c is defined as the difference between the course and the heading angles as follows:

$$\chi_c \overset{\triangle}{=} \chi - \psi.$$

Figure 2.12 depicts the vertical component of the wind triangle. When there is a down component of wind, we define the angle from the inertial North-East plane to $\mathbf{V}_a$ as the *air-mass-referenced flight-path angle* and denote it by γ_a. The relationship between the air mass referenced

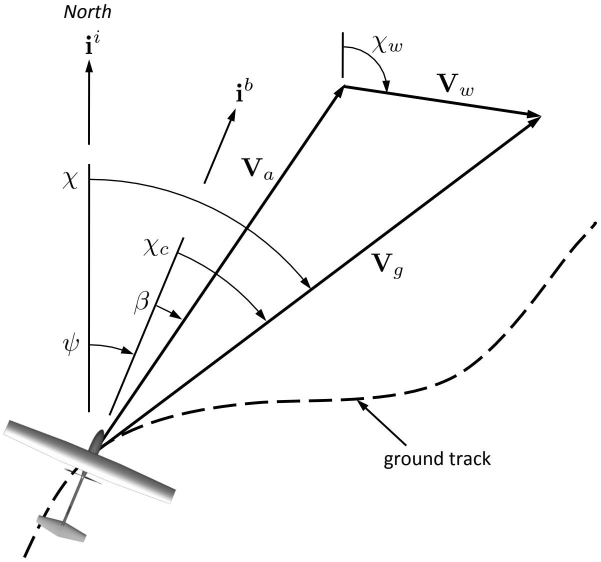


Figure 2.11: Heading is the direction that the MAV is pointed. Course is the direction of travel relative to the earth's surface. The crab angle is the difference between course and heading. In the absence of wind, the crab angle is zero.

flight path angle, the angle of attack, and the pitch angle is given by

$$\gamma_a = \theta - \alpha.$$

In the absence of wind $\gamma_a = \gamma$.

The ground speed vector in the inertial frame can be expressed as

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

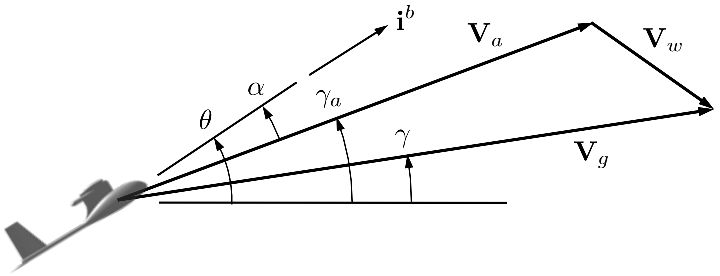


Figure 2.12: The wind triangle viewed from the vertical perspective.

where $V_g = \|\mathbf{V}_g\|$. Similarly, the airspeed vector in the inertial frame can be expressed as

$$\mathbf{V}_a^i = V_a \begin{pmatrix} \cos\psi\cos\gamma_a \\ \sin\psi\cos\gamma_a \\ -\sin\gamma_a \end{pmatrix},$$

where $V_a = \|\mathbf{V}_a\|$. Therefore, the wind triangle can be expressed in inertial coordinates as

$$V_g \begin{pmatrix} \cos\chi\cos\gamma \\ \sin\chi\cos\gamma \\ -\sin\gamma \end{pmatrix} = V_a \begin{pmatrix} \cos\psi\cos\gamma_a \\ \sin\psi\cos\gamma_a \\ -\sin\gamma_a \end{pmatrix} + \begin{pmatrix} w_n \\ w_e \\ w_d \end{pmatrix}. \tag{2.9}$$

Equation (2.9) allows us to derive relationships between V_g, V_a, χ, ψ, γ and γ_a. To find an expression for γ_a, multiply both sides of Equation (2.9) by $(\cos\chi\sin\gamma, \sin\chi\sin\gamma, \cos\gamma)$ to eliminate V_g and obtain

$$V_a\left(\sin\gamma_a\cos\gamma - \cos\gamma_a\sin\gamma\cos(\chi-\psi)\right) = \begin{pmatrix} w_n \\ w_e \\ w_d \end{pmatrix}^T \begin{pmatrix} \cos\chi\sin\gamma \\ \sin\chi\sin\gamma \\ \cos\gamma \end{pmatrix}.$$

If we assume that the crab angle $\chi_c = \chi - \psi$ is less than 30 degrees, then $\cos(\chi-\psi) \approx 1$ and we can solve for γ_a to obtain

$$\gamma_a \approx \gamma + \sin^{-1}\left(\frac{1}{V_a}\begin{pmatrix} w_n \\ w_e \\ w_d \end{pmatrix}^T \begin{pmatrix} \cos\chi\sin\gamma \\ \sin\chi\sin\gamma \\ \cos\gamma \end{pmatrix}\right). \tag{2.10}$$

To derive an expression for ψ, multiply both sides of Equation (2.9) by $(-\sin\chi, \cos\chi, 0)$ to get the expression

$$0 = V_a\cos\gamma_a\left(-\sin\chi\cos\psi + \cos\chi\sin\psi\right) + \begin{pmatrix} w_n \\ w_e \end{pmatrix}^T \begin{pmatrix} -\sin\chi \\ \cos\chi \end{pmatrix}.$$

Solving for ψ gives

$$\psi = \chi - \sin^{-1}\left(\frac{1}{V_a\cos\gamma_a}\begin{pmatrix} w_n \\ w_e \end{pmatrix}^T \begin{pmatrix} -\sin\chi \\ \cos\chi \end{pmatrix}\right). \tag{2.11}$$

An expression for groundspeed can be obtained by taking the squared norm of each side of Equation (2.9) to get

$$V_g = \sqrt{V_a^2 + V_w^2 + 2V_a \begin{pmatrix} \cos\psi\cos\gamma_a \\ \sin\psi\cos\gamma_a \\ -\sin\gamma_a \end{pmatrix}^T \begin{pmatrix} w_n \\ w_e \\ w_d \end{pmatrix}}, \tag{2.12}$$

where $V_w = \|\mathbf{V}_w\| = \sqrt{w_n^2 + w_e^2 + w_d^2}$ is the wind speed.

Because wind typically has a significant impact on the flight behavior of small unmanned aircraft, we have tried to carefully account for it throughout the text. If wind effects are negligible, however, some important simplifications result. For example, when $V_w = 0$, we also have that $V_a = V_g$, $u = u_r$, $v = v_r$, $w = w_r$, $\psi = \chi$ (assuming also that $\beta = 0$), and $\gamma = \gamma_a$.

2.5 Differentiation of a Vector

In the process of deriving equations of motion for a MAV, it is necessary to compute derivatives of vectors in reference frames that are moving with respect to one another. Suppose that we are given two coordinate frames, $\mathcal{F}^i$ and $\mathcal{F}^b$, as shown in Figure 2.13. For example, $\mathcal{F}^i$ might represent the inertial frame and $\mathcal{F}^b$ might represent the body frame of a MAV. Suppose that the vector $\mathbf{p}$ is moving in $\mathcal{F}^b$ and that $\mathcal{F}^b$ is rotating (but not translating) with respect to $\mathcal{F}^i$. Our objective is to find the time derivative of $\mathbf{p}$ as seen from frame $\mathcal{F}^i$. To do this, denote the angular velocity of frame $\mathcal{F}^b$ in $\mathcal{F}^i$ as $\boldsymbol{\omega}_{b/i}$ and express the vector $\mathbf{p}$ in terms of its vector components as

$$\mathbf{p} = p_x\mathbf{i}^b + p_y\mathbf{j}^b + p_z\mathbf{k}^b. \tag{2.13}$$

The time derivative of $\mathbf{p}$ with respect to frame $\mathcal{F}^i$ can be found by differentiating Equation (2.13) as

$$\frac{d}{dt_i}\mathbf{p} = \dot{p}_x\mathbf{i}^b + \dot{p}_y\mathbf{j}^b + \dot{p}_z\mathbf{k}^b + p_x\frac{d}{dt_i}\mathbf{i}^b + p_y\frac{d}{dt_i}\mathbf{j}^b + p_z\frac{d}{dt_i}\mathbf{k}^b, \tag{2.14}$$

where d/dt_i represents time differentiation with respect to the inertial frame. The first three terms on the right-hand side of Equation (2.14) represent the change in $\mathbf{p}$ as viewed by an observer in the rotating $\mathcal{F}^b$ frame. Thus, the differentiation is carried out in the moving frame. We denote this local derivative term by

$$\frac{d}{dt_b}\mathbf{p} = \dot{p}_x\mathbf{i}^b + \dot{p}_y\mathbf{j}^b + \dot{p}_z\mathbf{k}^b. \tag{2.15}$$

The next three terms on the right-hand side of Equation (2.14) represent the change in $\mathbf{p}$ due to the rotation of frame $\mathcal{F}^b$ relative to $\mathcal{F}^i$. Given that $\mathbf{i}^b$, $\mathbf{j}^b$, and $\mathbf{k}^b$ are fixed in the $\mathcal{F}^b$ frame, their

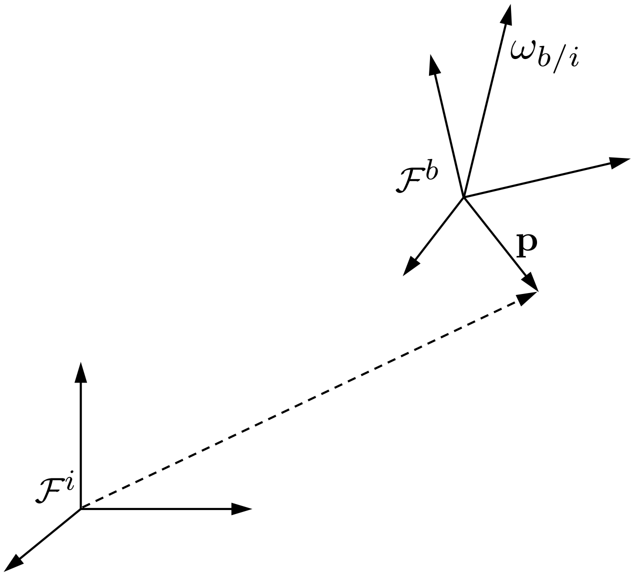


Figure 2.13: A vector in a rotating reference frame.

derivatives can be calculated as shown in [9] as

$$\begin{aligned}\dot{\mathbf{i}}^b &= \boldsymbol{\omega}_{b/i} \times \mathbf{i}^b \\ \dot{\mathbf{j}}^b &= \boldsymbol{\omega}_{b/i} \times \mathbf{j}^b \\ \dot{\mathbf{k}}^b &= \boldsymbol{\omega}_{b/i} \times \mathbf{k}^b.\end{aligned}$$

We can rewrite the last three terms of Equation (2.14) as

$$\begin{aligned}p_x\dot{\mathbf{i}}^b + p_y\dot{\mathbf{j}}^b + p_z\dot{\mathbf{k}}^b &= p_x(\boldsymbol{\omega}_{b/i} \times \mathbf{i}^b) + p_y(\boldsymbol{\omega}_{b/i} \times \mathbf{j}^b) + p_z(\boldsymbol{\omega}_{b/i} \times \mathbf{k}^b) \\ &= \boldsymbol{\omega}_{b/i} \times \mathbf{p}. \end{aligned} \tag{2.16}$$

Combining results from Equations (2.14), (2.15), and (2.16), we obtain the desired relation

$$\frac{d}{dt_i}\mathbf{p} = \frac{d}{dt_b}\mathbf{p} + \boldsymbol{\omega}_{b/i} \times \mathbf{p}, \tag{2.17}$$

which expresses the derivative of the vector $\mathbf{p}$ in frame $\mathcal{F}^i$ in terms of its change as observed in frame $\mathcal{F}^b$ and the relative rotation of the two frames. We will use this relation as we derive equations of motion for the MAV in Chapter 3.

2.6 Chapter Summary

In this chapter, we have introduced the coordinate frames important to describing the orientation of MAVs. We have described how rotation matrices can be used to transform coordinates in one frame of reference to coordinates in another frame of reference. We have introduced the 3-2-1 Euler angles (ψ, θ, and ϕ) as a means to rotate from the inertial coordinate frame to the body frame fixed in the orientation of the MAV. We have also introduced the angle of attack α and the sideslip angle β to describe the relative orientation of the body frame, the stability frame, and the wind frame. An understanding of these orientations is essential to the derivation of equations of motion and the modeling of aerodynamic forces involved in MAV flight. We have introduced the wind triangle and have made the relationships between airspeed, ground speed, wind speed, heading, course, flight path angle, and air mass relative flight path angle explicit. We have also derived the Coriolis formula for the differentiation of a vector.

Notes and References

There are many references on coordinate frames and rotations matrices. A particularly good overview of rotation matrices is [10]. Overviews of attitude representations are included in [8, 11]. The definition of the different aircraft frames can be found in [4, 1, 7, 12]. A particularly good explanation is given in [13]. Vector differentiation is discussed in most textbooks on mechanics, including [14, 15, 16, 9].

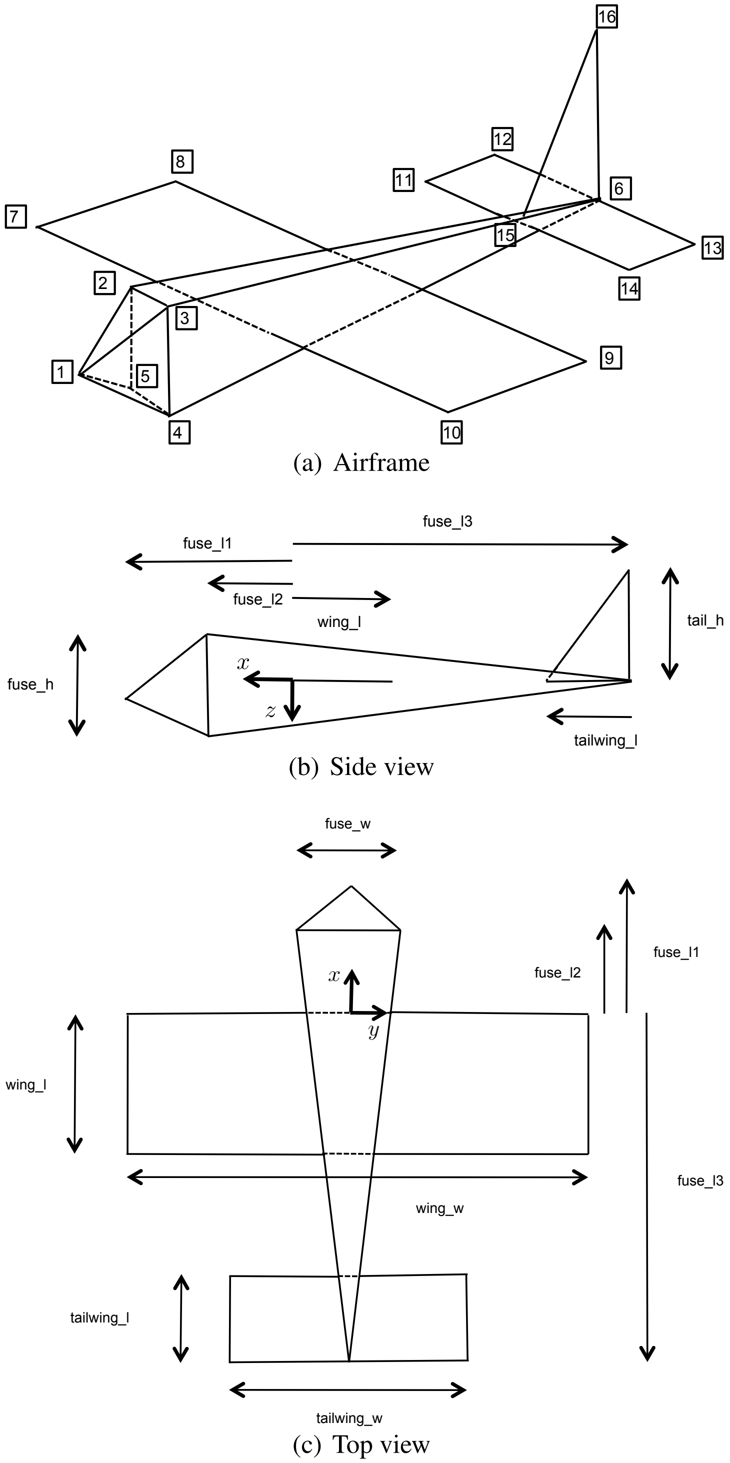
2.7 Design Project

The objective of this assignment is to create a 3D graphic of a MAV that is correctly rotated and translated to the desired configuration. Creating animations in Simulink is described in Appendix B and example files are contained at the textbook website.

2.1 Read Appendix B and study carefully the spacecraft animation using vertices and faces given at the textbook website.

2.2 Create an animation drawing of the aircraft shown in Figure 2.14.

2.3 Using a Simulink model like the one given on the website, verify that the aircraft is correctly rotated and translated in the animation.



(a) Airframe

(b) Side view

(c) Top view

Figure 2.14: Specifications for animation of aircraft for the project.

2.4 In the animation file, switch the order of rotation and translation so that the aircraft is first translated and then rotated, and observe the effect.