A wide array of potential applications exist for robots that have the level of mobility offered by flight. The military applications of aerial robotics have been recognized ever since the beginnings of powered flight, and they have already been realized to sometimes spectacular effect in surveillance, targeting, and even strike missions. The range of civilian applications is even greater and includes remote sensing, disaster response, image acquisition, surveillance, transportation, and delivery of goods.

This chapter first presents a brief history of aerial robotics. It then continues by describing the range of possible and actual applications of aerial robotics. The list of current challenges to aerial robotics is then described. Building from basic notions of flight, propulsion, and available sensor technology, the chapter then moves on to describe some of the current research efforts aimed at addressing the various challenges faced by aerial robots.

The challenges faced by aerial robots span several and distinct fields, including state regulations, man–machine interface design issues, navigation, safety/reliability, collision prevention, and take–off/landing techniques. The size of aerial robots can considerably influence their flight dynamics, and small aerial robots can end up looking considerably different from their larger counterparts. Similar to their manned counterparts, aerial robots may enjoy diverse propulsion systems and operate over large speed ranges.

Aerial robots must be equipped with reliable position and actuation equipment so as to be capable of controlled flight, and this constitutes a nontrivial requirement prior to doing research or development in this field. However, many universities, research centers, and industries have now met this requirement and are actively working on the challenges presented above. The largest obstacle to the commercial development of aerial robots is, however, the necessity to comply with and support a regulatory environment which is only beginning to address these rapidly developing systems.
44.1 Background

The term *aerial robotics* is often attributed to Robert Michelson [44.1], as a way to capture a new class of highly intelligent, small flying machines. However, it is clear that the range of systems and activities covered under the label *aerial robotics* could extend much further, and that its roots can be found far back in the beginning of the 20th century, together with the birth of aviation. Behind the word *aerial robotics* we can find several meanings: it could mean *robotic flying machines*, that is, a mission-independent, platform-oriented concept; however, it could also mean *robotics that use flying machines*, that is, a platform-independent, mission-oriented concept. Finally, it could mean a combination of the above, that is a description of the robotic platform, together with its robotic mission. In aerospace jargon, robotic flying machines are commonly referred to as *unmanned aerial vehicles* (UAVs), while the entire infrastructures, systems and, human components required to operate such machines for a given operational goal are often called *unmanned aerials systems* (UASs). Finally, it is worth noting that many current manned aerial systems definitely carry relevant features of some robotic systems, and so much of this discussion is relevant to manned aircraft.

44.2 History of Aerial Robotics

The history of aerial robotics is very closely tied to the history of flight itself. Indeed, the rate of fatalities associated with early manned flight tests probably convinced engineers that there was a need to operate flying machines without the presence of humans on board even before potential applications of unmanned aircraft surfaced. In 1903, heavier-than-air flight was unambiguously shown to be feasible, following the achievements of the Wright brothers. The first successful powered flight was unmanned, presumably to reduce the risk to the pilot and to allow a smaller and less expensive vehicle (reasoning that is still put forth today) by Samuel P. Langley’s *Number 5* in 1896 [44.2].

Finding a truly defining moment for aerial robotics is a challenge, with encyclopedias dating the concept back to Leonardo da Vinci, while Newcome [44.3], in his history of unmanned aviation, gives early credit to Nikola Tesla for devising a robotic vehicle remotely controlled by electromagnetic waves, and with enough onboard logic to recognize and execute remotely transmitted orders. However, the concept imagined and engineered by Tesla did not apply specifically to airborne vehicles.

Using the definition

>An aerial robot is a system capable of sustained flight with no direct human control and able to perform a specific task,

leads us almost immediately to the Hewitt–Sperry automatic airplane, developed before and during World War I [44.3]. The airplane’s purpose was to act as a flying torpedo, carrying onboard *intelligence* to sustain flight over long periods of time without human intervention. Such intelligence was provided by a complex system involving Sperry’s own gyroscopes, mechanically coupled to the airplane’s control surfaces so as to stabilize the vehicle. This made the airplane suitable for...
human remote control and, eventually, prosecution of distant targets. As discussed later, one of the key characteristics of aerial robotics is this particular necessity for the robot to sustain itself in the air with no human intervention, which requires the early adoption and understanding of the critical role played by onboard intelligence, much more so than other robotic applications. While a string of inventors in many countries came to develop ever more sophisticated machines, credit goes to the German V-1 cruise missile for making a lasting, and unfortunately deadly, impact on large segments of the population in England. This form of robotic aircraft owed its relative inefficiency (three out of four vehicles reportedly missed their target – predominantly London) to mechanical failures and lack of good navigation capabilities beyond dead-reckoning assisted by gyroscopic devices. Many of the ensuing aerial robotics developments followed the initial idea of defense applications for unmanned systems, that is, ever more accurate flying machines for the purpose of either reconnaissance or weapon delivery. One notable machine was the US Navy’s Gyrodyne QH-50 DASH, an unmanned helicopter developed in the 1950s and operated from US destroyers, which was able to perform reconnaissance missions and deliver torpedoes (Fig. 44.2). However, these machines remained relatively unintelligent, and their level of autonomy remained limited to the ability to sustain flight using complex inertial and other measurement systems.

---

**Fig. 44.1** V-1 German cruise missile (1940s)

**Fig. 44.2** QH-50 DASH unmanned helicopter on final approach (US Navy)

**Fig. 44.3** Taxonomy of unmanned aerial vehicles (after R. Weibel [44.4, 5])
Field and Service Robotics

Part F

44.3 Applications of Aerial Robotics

Listing all possible applications of aerial robotics is very challenging. However, there are fewer actual implementations, because of the necessity for the corresponding operations to comply with stringent air safety regulations. In the following, a brief description of possible and current applications is provided.

44.3.1 Possible Applications of Aerial Robots

The list of possible applications of aerial robots is long. According to [44.6, 7], such applications fall within nine categories:

- Remote sensing such as pipeline spotting, powerline monitoring, volcanic sampling, mapping, meteorology, geology, and agriculture [44.8, 9], as well as unexploded mine detection [44.10].
- Disaster response such as chemical sensing, flood monitoring, and wildfire management.
- Surveillance such as law enforcement, traffic monitoring, coastal and maritime patrol, and border patrols [44.11].
- Search and rescue in low-density or hard-to-reach areas.
- Transportation including small and large cargo transport, and possibly passenger transport.
- Communications as permanent or ad hoc communication relays for voice and data transmission, as well as broadcast units for television or radio.
- Payload delivery e.g., firefighting or crop dusting.
- Image acquisition for cinematography and real-time entertainment.

From a robotics perspective, probably the next significant technological enabler is the advent of lightweight processors and sensor systems, together with global navigation satellite systems, which allowed aerial robots to perform increasingly complex tasks. Japan, motivated by a policy of food self-sufficiency combined with a massive shortage of agricultural workforce, took a lead in aerial robotics by developing highly reliable helicopters in the 1980s, such as the Yamaha R-50 and subsequent Yamaha R-Max (Fig. 44.4), with similar systems developed by other companies such as Yanmar. These robotic helicopters are used primarily for crop dusting applications, especially over wet rice fields. These vehicles also turned out to be very popular among universities and other institutions for their unmatched ability to fly reliably and take off and land from limited areas, allowing researchers to focus their attention on developing higher levels of autonomy beyond basic vehicle navigation and control. Military applications of unmanned robotics followed the track of the German V1, with the advent of modern cruise missile technology.

From the mid 1980s on, the development of aerial robots has followed an exponential pace, with one notable trend for operational aerial robots to systematically find military applications as their most significant market.

A snapshot of available aerial vehicle platforms and systems appeared recently in [44.6], and regular updates on available platforms can be found in the aerospace literature. The chart in Fig. 44.3 illustrates the number of machines currently under development or in operation, which exceeds 200 vehicle types. From this chart, one concludes, however, that the vast majority of current, operational aerial robots are fixed-wing aircraft, and that they tend to be present at all altitudes.

We must also remark that onboard robotic intelligence has made its way not only into unmanned aircraft but also manned aircraft. Many commercial airliners now have the ability to fly automatically from right after take-off (a decision left to the pilot) to right after landing, by engaging the autopilot and letting the aircraft fly a predetermined profile. In addition, these vehicles are now able to make systems-management decisions based on sensor inputs, sometimes escaping the human pilot’s ability to understand them.
Military applications of aerial robots follow the same descriptive lines, with a particular emphasis on remote sensing of humans and critical infrastructure, surveillance of human activity, and payload delivery (bombs, missiles, and ad hoc ground infrastructures devoted to communication and surveillance).

### 44.3.2 Current Applications

Current applications of aerial robots are somewhat fewer and they are at present driven by the military context.

**Aerial Observations**

The most important application of aerial robots is aerial observations, which can then be used for terrain mapping, environmental surveys, crop monitoring, target identification etc. There is a divide, however, between the state of the art for military applications and civilian applications, detailed below.

**Military Operations.** Military and government use of aerial robots has sharply increased in recent years in war zones. As a result, dozens of vehicles are now delivered every month, and end up flying in “hot” areas around the globe, most notably in Southwest Asia. The machines being flown range from man-portable machines flying at low altitudes, such as the Pointer or Raven aircraft, to mid-sized machines such as the Aerosonde, Seascan, or Shadow unmanned vehicles, to larger-sized vehicles such as the Predator or Global Hawk. Their wings span from a meter or so for the smaller vehicles to 35 m for Global Hawk (the same as a Boeing 737). Besides their use in military areas or war zones, these machines now find applications in border surveillance, with a particular interest in oceanic borders, where vehicles operate in desert or quasi-desert areas.

**Civilian and Private Applications.** Current civilian applications of aerial robots for surveillance and observation remain sporadic and ad hoc: unlike many other robotic devices, civilian aerial robots do not operate in closed environments but in civilian airspace, which is subject to strong safety regulations that do not yet systematically accommodate aerial robots. Consequently, the current trends in civilian aerial robotics are as follows.

Small-scale, intermittent civilian aerial robotic applications tend to happen in relatively isolated environments (e.g., for film making or environmental surveys), and often follow the safety and operations rules most familiar to their operators, derived from model aircraft operations. Most often, the operated machines do in fact bear much resemblance to radio-controlled model airplanes. Other intermittent applications involve the use of unmanned vehicles for specific reconnaissance tasks, such as the detection of fish banks from trawlers [44.12]. Such a task constituted one of the original purposes for the development of machines such as the Seascan unmanned aerial vehicle.

Long-term scientific applications such as atmospheric sampling experiments [44.13] appear to benefit considerably from aerial robots. One report [44.13] reads:

> From March 6 to March 31 2006, we probed the polluted atmosphere over the North Indian Ocean with lightweight unmanned aerial vehicles (or UAVs) fully equipped with instruments. This UAV campaign launched from the Maldives laid a solid foundation for the use of UAVs to study how human beings are polluting the atmosphere and their impact on climate, including global warming.

Because such activities naturally require much planning ahead, special permits can be obtained from aviation authorities within time limits that do not significantly affect the overall experimental project. Other scientific missions led with success include [44.14], where the authors were able to survey Mount St. Helens (then active) by taking advantage of the temporary interdiction to fly in the vicinity of the volcano.

With the progressive introduction of aerial robots in the regulatory framework of many countries, we believe that intermittent applications of aerial robotics in populated areas will eventually become commonplace. However, this requires that flight authorizations be delivered within a fraction of the time duration of the event: for example, firefighting operations are often triggered within a few seconds of the fire alert. Permits for aerial robotic support should therefore be delivered about as quickly if they are ever to be embraced by firefighters.

**Payload Delivery**

Under the heading payload delivery, we find the numerous applications of aerial robots aimed at delivering solid, liquid or gaseous products in areas that are hard to reach for humans. So far, the most successful civilian application has been chemical crop spraying using small unmanned helicopters. Leveraging the high costs and prices associated with crop culture in Japan, several thousand helicopters have been purchased by farmers, resulting in a profitable
operation both for themselves and for the helicopter manufacturers, among them Yamaha and Yanmar. However, this application remains unique and involved the involvement of Japan’s government for it to be successful.

Besides this particular application, military applications form the bulk of unmanned aerial robotics for the purpose of payload delivery, beginning in its crudest form with missiles, and evolving towards cruise missiles, able to navigate for thousands of miles and reach their targets with high precision. One of the most talked-about recent military application of aerial robots for payload delivery involves the Predator aircraft equipped with Hellfire missiles.

### 44.4 Current Challenges

In the following, we introduce six major challenges for aerial robotics. This list is not meant to be exhaustive, but it reflects the current focus of researchers. How these challenges are addressed will be discussed later.

#### 44.4.1 Regulations and Certification

A big challenge to the success of aerial robots is doubtlessly their acceptance by certification authorities. Indeed, the operation of aerial robots is currently significantly limited by regulatory constraints. This is due to the complex set of regulations put in place by national agencies (e.g., the Federal Aviation Administration in the US, the National Air Traffic Services in the UK, or the Direction Générale de l’Aviation Civile in France), whose aim is to maintain very high levels of safety for air traffic. The downside of the excellent safety record reached by regulatory agencies is their (justified) risk adversity, and therefore a slow acceptance of disruptive technologies such as aerial robots. This is compounded by the current rapid pace of change and related lack of standards among aerial robotic systems and how they are used. However, with the help of other organizations, such as the Radio Technical Commission for Aeronautics (RTCA), regulatory agencies have moved forward towards establishing rules for the routine operation of aerial robots. Such rules include the ability for aerial robots to see and avoid or sense and avoid other traffic at least as well as a human pilot.

In the recent past, many aerial robotics research activities and corresponding flight tests have occurred at very low altitude. In the absence of clearly defined rules by regulatory bodies until recently (2000), many researchers have operated under the rules of local radio-controlled aircraft associations (e.g., the Academy of Model Aeronautics). However, there is a rapidly growing trend for radio-controlled vehicles to incorporate more onboard electronics, including radio transmitters and sometimes guidance systems. In this environment, one can expect regulatory bodies such as the FAA to continue to evolve their policies.

The ensuing challenge is for the research community to develop the requirements and subsequent technology that meets the constraints set by the regulatory agencies, or to propose and justify alternate constraints. In particular, the maturation of aerial robots leading to their everyday use in populated areas will require the development of more reliable components, defined maintenance procedures, formal training programs, and the automation of emergency procedures (such as the forced landing process). The core technology for UAS already exists to demonstrate safety concepts. However, developing highly dependable systems – and making such dependability guarantees acceptable to the regulatory authorities – is a current and urgent challenge.

#### 44.4.2 Human–Machine Interfaces

The pilot interfaces used for manned aircraft have evolved continuously since the first manned aircraft. The standards that exist today directly benefit safety and operator costs by minimizing operational errors and training time when transitioning between aircraft types. Despite ongoing development efforts [44.15], this can-
not be said of aerial robotic systems, which run cover a much wider range of autonomy, mission capability, and operator skill. Add to this the desire to have single operators control multiple aircraft, and it clear this area presents an ongoing challenge for researchers.

44.4.3 Navigation

Figuring out absolute and relative position is a central issue for aerial robots, as it is for other robotics activities. The existence of a significant manmade infrastructure (the global navigation satellite system – GNSS) makes basic navigation easy but remains the subject of an intense debate; indeed, systems that overly depend on such infrastructure lack resilience and tolerance to positioning services shortage, whether such a shortage originates from the system itself or from the particular robot configuration (in cluttered environments such as cities). This situation will improve with the development of highly reliable multimode navigation systems with built-in integrity monitors, and with three independent satellite navigation constellations (Glonass, Galileo, and GPS) currently deployed or under deployment.

The challenge for researchers is to develop navigation technologies that allow aerial robots to live without manmade external navigation infrastructure, to handle the situations when it is not available.

44.4.4 Agile Flight and Fault Tolerance

Nearly every aircraft in operational use today has been challenged to fly far beyond its flight envelope during flight tests, including famous maneuvers such as that of the Boeing 707 that, during early flight demonstrations to customers, performed a full barrel roll. The purpose of these demonstrations is not only to show the full capabilities of the vehicle, but also to bring a sense of safety to the pilots, that the aircraft is still able to perform well after its goes into some upset condition. What applies to large, manned aircraft also applies to aerial robots, which must be able to keep operating well at unusual attitudes and under partial system failures such as loss of actuation [44.16]. Researchers must develop automation systems that meet this need.

44.4.5 Obstacle Avoidance

The ability for a vehicle to manage its position away from obstacles represents a significant issue and a necessity for low-altitude operations in crowded environments. One of the key features of aerial robots is their possibly high speeds, which challenges many existing sensor management and data processing algorithms, especially their ability to detect hard-to-see obstacles such as suspended cables quickly.

44.4.6 Aerial Robot Landing and Interaction with Other Vehicles

Owing to the finite endurance of aerial robots, landing and docking are particularly important to them. While landing constitutes an important element, docking with other vehicles, such as during aerial refueling, is also very important. All operations involving close coordination and physical interaction between vehicles or between a vehicle and the ground require further research.

44.4.7 Multivehicle Coordination

Several tasks require aerial robots to operate as a group, rather than as individual systems. This happens, for example, in order to create phased array antennas, or to perform object geolocation, or to improve the quality of a surveillance service (e.g., fire monitoring). Other tasks requiring multivehicle coordination include the requirement for collision avoidance. More recently, multivehicle coordination has been seen as a valuable way to design aerial robotic systems that remain functional despite individual vehicle failures.

44.5 Basic Aerial Robot Flight Concepts

44.5.1 Aerial Robot Flight and the Importance of Scales

Like all flying machines, the performance of aerial robots depends extensively on: (1) their size and (2) the characteristics of their lifting mechanisms (wings, rotors). A detailed description of vehicle flight mechanics is outside of the scope of this chapter; we can, nevertheless, recall a few fundamental and useful notions critical to successful flight. The reference [44.17] is an excellent and entertaining introduction to the subject, while [44.18, 19] offer a more academic perspective on
the matter. One important quantity is the mass of a flying machine. Roughly speaking, the mass of a flying machine is proportional to its volume, and therefore grows like the cubic power of its size. Another quantity is the lifting forces that keep a vehicle up in the air; these are proportional to the pressure exercised on the lifting surface (rotor or wing), times the area of the lifting surface, that is, roughly the second power of the vehicle size. The pressure itself is proportional to the density of the surrounding atmosphere (it need not be air only, think of Mars), multiplied by the square of the average velocity of the gas molecules relative to the lifting surface.

For illustrative purposes, consider the flying wing shown in Fig. 44.5 and a notional scaled-down version of it flying together. To make matters simpler, we assume that the scaled-down wing is about half the size of the full-sized wing. We now examine the impact of scales on the way these wings must fly.

Consider for example the lift created by the full-scale flying wing depicted in Fig. 44.5: it is proportional to $S \rho V^2 \alpha$, where $S$ is its total surface, $\rho$ is the air density, $V$ is the wing speed relative to the surrounding air, and $\alpha$ is the angle of attack (roughly speaking the angle between the wing chord and the flow of air).

To get an idea of the importance of scales, and following arguments developed in much greater detail in [44.17], we now examine the requirements for the scaled-down wing to fly at the same speed as the large wing, assuming all its components are shrunk by a factor two in size as shown in the picture, and examine the consequences of having to meet such requirements.

First, the mass of the wing roughly gets divided by a factor 8 ($2^3$). However, its lifting surface has shrunk by a factor of 4 only ($2^2$). So, if we were to fly this smaller wing at the same speed, same altitude, and same angle of attack as its big sister, the total generated lift must be $S/2\rho V^2$, that is, twice as much as necessary to balance out the effect of gravity.

Several solutions to this issue are possible: to reduce the actual wing dimensions at constant mass, slow it down, or reduce its angle of attack.

**Shrink the Wing**

To obtain the proper lift (while keeping the speed and angle of attack constant), we must shrink the wing area by another factor of two, or the wing dimensions by a factor $\sqrt{2}$. Thus we already see one important conclusion, which is that, at equal speed and angle of attack, the relative size of the wings with respect to the overall vehicle must shrink as the overall vehicle size goes down. Borrowing again from [44.17], this explains much of why a Boeing 747 looks, with its large deployed wings and relatively narrow fuselage, like a condor while a B737 feels more like a puffin, and the smaller Embraer 145 is like a dart, as shown in Fig. 44.6. All
these aircraft can fly about the same speeds and altitude ranges.

**Reduce the Speed**

If this option is chosen, then speed must be divided by a factor $\sqrt{2}$ for our scaled model to balance lift and weight. The consequences of reducing speed are many: the time required for mission completion of course increases. On the other hand, the drag generated by the flying machine (and which must be paid for by the propulsion system) goes down.

Pushed to their limits, the consequences of slowing down the vehicle as it shrinks can be quite dramatic: consider a dragonfly (one of the role models for micro-aerial robots) trying to land next to a Boeing 747 at the same airport. The figure below shows that the two share (very roughly) the same proportions.

For the sake of simplicity, assume that the dragonfly is the $1/1000$ scaled-down version of the Boeing 747. In order for both to fly level, and according to our rule, the dragonfly must fly a factor $\sqrt{1000} = 32$ slower than the B747. Assume the B747 flies at 500 km/h; that makes the dragonfly fly at about 17 km/h. Imagine now that the weather is gusty, with winds topping 30 km/h. The 747 (and its passengers) will see little variation in airspeed (from 470 to 530 km/h), and the variation in produced lift will be 33%, enough to shake the aircraft a bit, but not unusually bad. As for the dragonfly, the same gusts will create airspeed variations of well over 100%, and the produced lift will vary from zero to five or six times the nominal lift. A rough ride naturally follows, and indeed, the flight of smaller vehicles often looks much less smooth than that of large ones.

**Reduce the Angle of Attack**

The latter option, reducing the angle of attack, rests upon the fact that, roughly, the lift created by a wing (or a rotor), is a linear function of the angle of attack. This makes it possible to fly about the same speed with a scaled-down model of a flying machine. However, this option comes with significant drawbacks, especially for fixed-wing aircraft. In particular, the sensitivity of the lift created by the wing to external perturbations (e.g., air turbulence and wind gusts) would again be higher, creating another recipe for bumpy rides.

The previous considerations about the forces acting on aerial vehicles also apply to moments: consider the flying wings shown in Fig. 44.5, and assume that their density (mass per unit volume) is constant throughout. Their angular inertia about any axis are proportional to the fifth power of their size. On the other hand, the forces that apply to the wings are proportional to their area; thus when moments are computed, forces are multiplied by distances, and the resulting moments become proportional to volume, that is, the third power of vehicle size. Consider then the angular momentum equation

$$J \ddot{\theta} = M, \quad (44.1)$$

where $J$ is the moment of inertia of the vehicle and $M$ is the applied torque. The term to the left of (44.1) decreases much faster with vehicle size than that to the right of the equation. As a consequence, we might immediately conclude that the scaled-down flying wing is inherently much more maneuverable than the larger one, in the sense that it can change orientation much faster.

This opens up a wealth of possibilities for robotics: venturing into the world of small flying robots, enabled by improvements of battery power and computation densities opens new possibilities in terms of defining the way these vehicles fly and interact with their environment.

### 44.5.2 Propulsion Systems

Several propulsion systems exist for aerial robots, including: jet, internal combustion, rocket, and electric. Older but recurrent options also include pulse engines such as those used on the German V1.

Owing to established aircraft and helicopter propulsion technologies, internal combustion engines and jet engines form the bulk of the propulsion means for medium to large-sized operational vehicles (50 kg or more), allowing many of them to fly reliably over periods of several hours to several tens of hours. When considering operational robots, the kind of fuel used matters: preference is given to fuels al-
ready used in other devices, and preference goes to heavy fuels, which are less prone to sudden and dangerous combustion or explosion, for example after a crash.

Electric propulsion systems, once unthinkable, have become a reality for several small-sized aerial robots, thanks to the development of affordable brushless electric engines and lightweight batteries. Initially developed for computer and communication applications, these batteries have been very quickly adopted by small-sized (a few kg) aerial robots such as Aerovironment’s Pointer and Raven aircraft, which are able to fly over periods exceeding one hour. The National Aeronautics and Space Administration’s (NASA) Pathfinder unmanned aircraft combines lightweight electric engines with wing-mounted solar panels to yield the aircraft shown in Fig. 44.8.

A notable departure from these propulsion systems is the Mars airplane’s propulsion system [44.20]: with an inert, low-density atmosphere on Mars, such a vehicle relies on a rocket engine for propulsion.

### 44.5.3 Flight Vehicle Types and Flight Regimes

Several vehicle types form the bulk of aerial robots, including fixed-wing machines, helicopters, flapping wing systems, and combinations thereof. The boundaries between these vehicle types are, however, mostly inherited from historical developments and intellectual stove-piping, rather than any fundamental guidelines dictated by the laws of mechanics and thermodynamics. For that reason, it is easier and more logical to introduce different *flight regimes* than flight vehicles, although to every regime there naturally corresponds one particular vehicle.

There are essentially two flight regimes. In the first regime, called *hover*, the speed of the vehicle relative to the surrounding air is small, such that few or no forces act on the vehicle except those resulting from the propulsion system itself. In the second regime, which we may call *cruising flight*, there is a significant relative speed between the vehicle and its surrounding environment, and significant aerodynamic forces act on the vehicle; these aerodynamic forces then largely dominate those generated by the power system.

**Hover**

Hover is the condition when the vehicle body does not move significantly with respect to the air mass surrounding it. Under these conditions, only propulsion systems are available to keep the vehicle up in the air (for heavier-than-air systems). Helicopters epitomize these situations, and they are especially designed to sustain such hover conditions over long periods of time. Helicopters come in all sizes and shapes. Robotic helicopters are best represented by Yamaha’s R-50, and now RMAX models (see, for example, Fig. 44.4), and both have been a staple of airborne robotics research for years at several academic institutions because of their reliability and available payload, which allows them to carry many instruments of interest to robotics research (including navigation sensors such as video cameras, laser range finders, and radars). With the evolution of the economic and political context, one is bound to see other such machines abound in the future. Indeed, the ability to hover is extremely useful for delivery/pickup of materials, rescue missions, and, in general, any operations that require close proximity to rugged terrain.

Helicopters are not the only vehicles capable of hover, see for example the hover-capable fixed-wing aircraft in Fig. 44.9. Hovering aircraft, such as tailsitters,
have been tested successfully since the 1950s at the very least, and it is a classic trick for experienced remote control pilots to hover airplanes. Transitions from hover to forward flight and back have been automated [44.21]. As radio-control (R/C) equipment shrinks in size and mass, new generations of hovering vehicles will become available. Some of these vehicles include micro air and flapping wing vehicles.

Hovering flight typically not very fuel inefficient: the fuel consumption of a hovering vehicle can exceed that of a fixed-wing vehicle by an order of magnitude of more. This kind of consideration has led manufacturers to seek some of the mixed configurations shown in Fig. 44.9.

**Cruising Flight**

During cruising flight the aerial robot mostly uses its available surfaces and its speed relative to the surrounding atmosphere to generate lift and maintain altitude. Unlike hovering flight, cruising flight usually results in the aerial robot constantly meeting fresh air, which makes the range of adverse events to flight quite narrower. This, of course, is not true in the case when aerial robots fly in formation, in which case turbulence created by one robot may affect its neighbor(s), sometime adversely, and sometimes positively [44.22].

Robotic airplanes such as those shown in Fig. 44.8 epitomize fuel-efficient cruising flight, with large and highly optimized wings. Both aircraft are part of current NASA programs. While optimized for flight, the wings of the Mars aircraft must also be optimized for tight packaging and deployment constraints at the end of its long trip from Earth to Mars.

While many fixed-wing systems are optimized for cruising flight, any system in forward flight operates according to the same principles; for example, a helicopter in forward flight operates like an airplane whose wing is a flat disc spanning the area covered by its rotor. Constant-velocity cruising flight that generates lift from the available aerodynamic surfaces is more fuel efficient than hovering.

![Non-helicopter, hover-capable vehicles: (a) Joint Strike Fighter, (b) 1950s tailsitter aircraft and (c) Aurora Flight Sciences’ Golden Eye 100](image)

**Stalled and High-Angle-of-Attack Flight**

This flight condition can be seen as a transitional flight condition, where the characteristics of both forward flight and hover are present. Typically, an aircraft stalls when its tries to maintain altitude at low speeds: flying level at lower speeds forces the aircraft’s angle-of-attack to increase for the wings to produce more lift. However, past a critical angle of attack, the trend reverses and the lift produced by the wing decreases as the angle of attack keeps increasing. This reduced aerodynamic lift must then be compensated by increased throttle, resulting in a situation where the aircraft propeller not only acts as a means to move the aircraft forward, but also directly participates in maintaining aircraft altitude.

This flight condition, whether experienced on a helicopter or airplane, often results in important changes of the effect of control mechanisms, for example, a stalled Piper Tomahawk trainer aircraft at low throttle setting will experience ineffective ailerons, while its rudder efficiency will shift from yaw axis to roll axis control [44.23].

**44.5.4 Lighter-Than-Air Systems**

One way to deflect some of the concerns associated with high fuel consumption of heavier-than-air aircraft is to rely on lighter-than-air vehicles. While these vehicles are often associated with spectacular accidents and slow motion, they also offer an unmatched capability to fly for long periods of time (more than 48 h) and to do so silently [44.24, 25]. Establishing control over lighter-than-air vehicles can be, however, somewhat challenging. In particular these vehicles are quite sensitive to winds and often tend to go where the wind takes them. Smaller platforms used for research must therefore evolve in closed environments [44.26]. The use of such vehicles for outdoor research can be quite daunting, because their large size requires considerable infrastructure to store them.
Aerial robots exhibit the complex flight dynamics associated with flight vehicles. As a consequence, precise motion control rapidly becomes a necessity for any aerial robotics activity to occur successfully. This means that effort must go into reliable basic flight control before more advanced, intelligent mission management can be attempted. Inner-loop control is achieved by the right sensing equipment, and by adequate control algorithms. Efficient hovering vehicles tend to be unstable, which makes their stabilization more difficult than that of purely cruising vehicles, for which many commercial control packages are now commonly available.

44.6.1 Sensing and Estimation

Airborne robots come with a variety of sensing options, which include

- inertial navigation systems (gyroscopes, accelerometers)
- global navigation satellite systems (GLONASS, GPS, Galileo)
- terrestrial radio navigation systems (VHF omnidirectional range (VOR), distance measuring equipment (DME), instrument landing system (ILS))
- air data probes and altimeters
- radar and passive vision sensors
- magnetic compasses
- distance measuring (altitude radars, ultrasonic sensors, and laser range finders)

The choice of sensors is critical to obtaining a properly flying robot. Usually, the same suite of sensors may not apply to all phases of flight. We will concentrate our discussion on the first four sensor types.

Inertial Navigation Systems

Inertial measurement systems consist of a combination of usually three orthogonally mounted accelerometers and three orthogonally mounted gyroscopes (more may be used for the purpose of achieving redundancy). The accelerometer suite measures, up to sensor error, the accelerations experienced by the vehicle at the location of the inertial sensor minus gravity. Gyroscopes measure vehicle angular velocities. Modern inertial sensors are usually rigidly linked to the vehicle to form strapdown inertial measurements systems. Such systems have become very cheap and very popular. Unlike many nonflying applications, unaided inertial measurement packages are not sufficient for estimating the attitude of an airborne vehicle. Indeed, consider Fig. 44.10, showing two helicopters equipped with inertial measurement systems. One, straight up, is hovering. The second, upside down, races towards the ground with an acceleration of $2g$. The accelerations and rotation rates recorded by the onboard inertial measurement unit will be strictly the same in both cases. While attitude can be estimated by integrating angular rates over time, the approach will eventually fail without correcting for accumulated error from another source. It remains that inertial measurement units are extremely useful to measure variations in acceleration and angular velocities, and constitute a staple of inner-loop control systems. Small radio-controlled helicopters now come with built-in gyroscopic yaw dampers that make their manual operation much more manageable. A key progress was made when Analog Devices introduced a low-cost micromechanical gyroscope [44.27]. Since then, this technology appears to have made its way into most commercially available inertial measurement units (IMUs), thereby greatly reducing their cost and weight. Practical inertial navigation systems on aircraft typically receive at least position updates from other sensors (discussed below), called inertial aiding.

Global Navigation Satellite Systems

The Global Positioning System (GPS) and its Russian equivalent GLONASS and future European Galileo space-based systems offer real-time absolute position information, using a constellation of satellites circumnavigating the earth. Ever since the beginning of their operation, global navigation systems (GNS) have been the object of a debate concerning their use in aerial robotics, with many researchers recommending against using such a large manmade navigation infrastructure to achieve true autonomy. Their arguments tend to become justified by the occurrence of recent needs in such applications as Mars exploration and low-altitude flight.
in obstacle-laden environments (such as cities) where satellite-based navigation is often unavailable. Whenever they are available, however, satellite navigation systems are a convenient and cheap means for a vehicle to locate itself. This modest investment has often been the enabler of automatic flight for many researchers and is currently used by virtually all existing industrial systems. Pushed to their limits, satellite navigation systems have been shown to achieve the entire range of desired navigation and sensing functions, which include vehicle position and attitude: GPS-only flight for a small helicopter robot was achieved in 1995 at Stanford University [44.28].

Altimeter and Air Data Probes
Pressure-measuring devices are immensely useful sensors in aerial robotics. With ingenious arrangements of pressure sensors (such as pitot tubes) it is possible to measure (1) the atmospheric pressure at the location of the robot and (2) the so-called dynamic pressure, $\rho v^2/2$, along all vehicle axes. These data can themselves be transformed into precious information about the aerial robot’s altitude and direction of motion relative to the air it is flying in. Depending on the vehicle used, air data probes may be challenging to build and constitute an interesting field of investigation. Indeed, pressure probes are very sensitive to flow perturbations generated by fuselage, wings, and most importantly rotors and propellers. Air data probes are therefore positioned as far away from the main elements of the vehicle as possible (for example, along a boom extending forward of the vehicle fuselage). Figure 44.11 shows one such air data probe configuration. Mounted together with inertial measurement systems, air data probes allow aircraft to maintain stable flight at a prescribed altitude. With the current state of technology, they remain somewhat insufficient to achieve, alone, stable hovering flight for helicopters.

Passive Vision
Passive vision has become a very popular sensor for inner-loop control. Even unsophisticated light sensors able to differentiate between the intensity of infrared activity from the ground versus that emitted by the sky has made its way into small commercial products, mostly aimed at assisting remote-controlled vehicle flight. As will be discussed later, passive vision devices have also found applications for vehicle–obstacle and vehicle–vehicle proximity management, and for landing applications. Recent research aimed at using vision for inner-loop control applications includes work aiming at tracking relatively invariant features such as the horizon [44.29].

44.6.2 Estimator Design
The individual inputs collected from each sensor are usually not sufficient to estimate the state of the vehicle. Different sensors may be efficient over different flight regimes. The proper way to leverage individual information provided by each sensor is through an appropriate filtering process that can yield rather comprehensive information about the entire system’s state. Unlike ground robots, the necessity for good robot state estimates arises early in the robot development process since closed-loop flight would be impossible otherwise. However, the structure of the filters is usually a great deal simpler than their ground-based equivalent, since the difficulty of flight is compensated by a rather simple and uniform environment structure. As a consequence, simple filters such as (extended) Kalman filters are usually enough for a large number of applications and quickly enable flight [44.30].

44.6.3 Inner-Loop Control
Inner-loop control of aerial vehicles naturally builds upon the previously discussed state estimator, and is relatively easy for routine flight operations. By this we mean that the process by which a good vehicle controller is obtained only requires following standard textbook techniques such as proportional, integral, and derivative control or linear quadratic control [44.31–33], appropriately scheduled against essential parameters such as vehicle speed and altitude. As a result, several research groups, and now several companies, have built basic
guidance and control packages suitable for aerial robots, both fixed-wing and helicopter.

Among the notable recent advances for the inner-loop control of aerial robots, we find the successful experimental application of adaptive and learning control techniques [44.21, 34, 35], which offer stable controlled helicopter flight from very coarse initial vehicle dynamics knowledge.

### 44.7 Active Research Areas

This section presents some of the active research in aerial robotics. Such research efforts aim at answering the challenges outlined in Sect. 44.4.

#### 44.7.1 Interfacing with the Human Infrastructure: Meeting the Regulations

While manned flight operations indeed have an excellent safety record, the price paid for this safety is a strong specialization of those regulations to human-operated systems, and a slow evolution of these regulations towards accepting aerial robots of all sizes. Currently none of the existing aerial robots is able to meet these regulations in the absence of a human pilot. This includes the ability to see (or sense) and avoid other aircraft, comply with air-traffic rules, and operate harmoniously with the current ground-based, manned air-traffic control system [44.4, 5, 36]. In the case of the US however, the FAA has acknowledged the economic potential of aerial robots by opening an office specifically devoted to such systems (the Unmanned Aircraft System Group), and by delivering permits to fly over certain areas (especially disaster areas) within a couple of hours of a request. However, the FAA currently emphasizes access for remotely piloted machines (as opposed to fully autonomous machines), where a ground-based pilot must have the means to communicate by voice with the FAA control center in charge of the geographical area where the vehicle is operated.

Efforts to help aerial robots improve their interface with other vehicles include the adaptation of existing systems to prevent mid-air collisions between aerial robots and other traffic, the development of see-and-avoid procedures, and means to interact with an aerial robot as one would interact with a human pilot (natural language interfaces). We now detail these efforts.

**Collision Avoidance for Remotely Piloted Vehicles**

The most immediate efforts aimed at inserting aerial robots in the civilian airspace consists of adapting existing airborne collision avoidance systems (ACAS), originally designed for manned systems, to aerial robots. Such systems are based on cooperative position information sharing between aircraft extracted from radar-based navigation and surveillance systems. The reason for emphasizing such systems over other, newer technologies is that they have already undergone extensive, and expensive, development, validation and testing. As such, several unmanned vehicles, such as the Global Hawk unmanned aircraft, are now fitted with ACAS systems [44.37]. However, such systems are not automated, meaning that the remote human pilot ultimately decides whether to execute the maneuvers recommended by the system. The possibility of completely automating such collision avoidance systems is the object of recent studies [44.38], with the clear intent of fitting them on aerial robots. However, the weight of ACAS system hardware, as well as the power required to operate them, makes such systems suitable only for large vehicles. In the context of rapidly evolving technology (such as the generalization of positioning information using global navigation satellite systems), current ACAS technology may also become rapidly obsolete.

**Sense and Avoid**

The idea, encouraged by institutional service providers, is to ensure that aerial robots are able to detect the presence of other traffic and avoid it as necessary and at least as well as a human pilot. Several candidate sensing technologies are currently in development, including passive vision systems [44.39–42], in an effort aimed at making aerial robots as able as humans to avoid other traffic when the sky is clear (visual flight rules). While much can be done in the visible spectrum, concerns over vehicle flight in clouds raises the necessity to consider other frequency bands, such as the near infrared, if aerial robots will need to be able to detect other traffic better than a human pilot [44.43].

**Human Interfacing**

Another active research venue is to facilitate the interfacing of robots with humans (e.g., an aerial robot interacting with a human air-traffic controller). Recent research in human–aerial robot interaction has
shown that aerial robots can interact productively with humans, by combining natural language processing interfaces with advanced vehicle path and task planning capabilities [44.44–46]. A natural interface might capture standard, unambiguous phraseology such as North Atlantic Treaty Organization (NATO) phraseology or air-traffic control phraseology. The impact of such technology on aerial robots would be profound since they would then be able to enter airspace with little or no visibility and be able to interact with the predominantly ground-based, human-intensive air-traffic control structure.

### 44.7.2 High-Agility Flight

One of the important characteristics of aerial robots is the ability to operate at the limit of its structural strength, unimpeded by the presence and physiological limitations of a human pilot. This allows aerial robots, especially small ones, to operate very aggressively. As a result, several research groups have explored the possibility of achieving aggressive flight with either fixed-wing or rotary-wing vehicles. The key factors that have enabled the onset of highly aggressive flight has been the emergence of lightweight computing environments and sensors, notably GPS and inertial systems. Indeed, aggressive flight (where aggressive flight refers to any abrupt change in vehicle attitude) is closely related to available vehicle mass and size, as discussed earlier.

Figure 44.12 shows the evolution of three helicopter configurations over time. While the vehicle platform has evolved little or not at all, the onboard avionics has progressively shrunk. In the mid-1990s, the onboard avionics typically would weigh the same or more than the helicopter mass. By the early 2000s, the onboard avionics would be about half of the vehicle mass, while by the mid-2000s the onboard avionics represent only a small fraction of the helicopter mass.

The corresponding levels of achievable agility have evolved correspondingly. By the early 2000s, basic aerobatic maneuvers became feasible [44.47], and by 2007 fully fledged aerobatics had been reported [44.48]. Other efforts involving unusual flight attitudes and fault recovery include those of Chiba University (Japan), who demonstrated autorotation landings for autonomous helicopters [44.49].

Parallel to rotorcraft agile flight, several efforts have also successfully enabled aerial agility for fixed-wing robots [44.21].

### 44.7.3 Take-Off, Landing, and Interaction with Other Vehicles

One of the richest current areas of investigation for aerial robots involves vehicle operation next to other vehicles or infrastructures. These operations include take-off, landing, docking, and separation.

**Take-Off and Landing**

Take-off and landing experimentation and research is proving particularly interesting for small-sized aerial robots. Indeed, the dynamics of smaller vehicles enable strong departures from conventional, manned-vehicle take-off and landing operations, for example, most fixed-wing unmanned aerial vehicles under 5 kg are better off simply flying into the ground than attempting to land in a smooth fashion. One of the best illustrations of how vehicle landing procedures may dramatically change for smaller-sized aerial robots is Insitu’s and Hood Technology’s Skyhook concept: small, fixed-wing aerial robots are recovered by allowing them to catch a vertical cable with the tip of one of their wings [44.12]. The cable itself is held by means of a crane, itself mounted on a surface vehicle (e.g., truck or ship). At take-off, similar scaling considerations apply, with many fixed-wing vehicles being launched by hand or by means of a catapult. One of the consequences of the increased tolerance of small ve-
vehicles to crash landings is also their reduced need for high-resolution navigation information, for example, it has been shown possible to land small-sized vehicles on a designated target with monocular vision only [44.50].

Helicopter robots have, so far, not benefitted from the same kind of developments, and much of their take-off and landing procedures are similar to their larger counterparts. The main reason may be attributed to the presence of a fragile rotor that spins at high speed, and that must avoid contact with other vehicles or the ground. Many of the current robotic helicopter landing procedures simply consist of hovering above the landing area, then commanding a limited descent rate until the vehicle records it has touched the ground. More challenging situations (e.g., sloped terrain or moving platforms), traditionally handled by humans in large platforms, remain difficult for aerial robots. For this reason, the helicopter landing problem has attracted the attention of many research teams. On the one hand, there have been many efforts combining advanced sensing environments [44.51–55] with advanced control algorithms to enable affordable landing in structured environments which are not simply horizontal landing pads. On the other hand, identifying suitable landing places in unprepared environments by means of remote sensing and signal processing is also an area of active research [44.56–58].

Operations in the Vicinity of Other Vehicles: Docking and Undocking

Docking operations for unmanned aerial vehicles are necessary to improve their range and autonomy. Indeed, it is conceivable that some optimal aerial robot configuration consist of a parent–child system, whereby a larger machine provides a primary deployment and retrieval mechanism for several smaller vehicles. Such a concept has existed for a long time, with airships acting as carriers for smaller aircraft [44.59]. More recently however, it is in-flight aerial refueling that has motivated recent research on docking aerial robots. Indeed, the possibility for such vehicles to refuel considerably increases their operational range [44.60–62]. The NASA Dryden Flight Research Center has recently reported the completion of the first vision-aided fully automated aerial refueling operation, using computer vision for the purpose of recognizing and tracking the fuel hose...
that must be captured by the aerial robot, as shown in Fig. 44.14.

Undocking operations are comparatively easier to perform. They remain, however, spectacular since the dynamics of the aerial robot dramatically change as it is dropped from its mother ship. An extreme example of such a situation is illustrated by Georgia Tech’s successful dropping of a small ducted fan aerial robot from a larger autonomous helicopter. The small ducted fan then successfully stabilized itself. Pictures of this experiment are shown in Fig. 44.15.

### 44.7.4 Reactive Flight in Cluttered Environments and Obstacle Avoidance

Flight in cluttered environments includes any phase of the flight where vehicles are in close proximity to obstacles. This flight mode is particularly important for low-altitude applications. Several achievements have been reported in this area in the recent past, using a variety of sensing techniques.

Among the first significant works relying on passive vision techniques, Beard and McLain’s certainly stands out as one of the most entertaining and spectacular [44.63], using fixed-wing vehicles performing autonomous flight within a canyon using low-cost, optical flow computation techniques.

Other institutions involved with active as well as passive sensing techniques for vehicle navigation in cluttered environments and obstacle avoidance include Carnegie-Mellon University [44.64], where the authors report fast vehicle flight in highly cluttered environments, including obstacles as difficult to deal with as suspended cables. The NASA Ames research center also recently reported successes along similar lines as part of their work on adaptive landing in unprepared environments [44.56–58].

### 44.7.5 Path Planning and Higher-Level Planning Capabilities

#### Single-Robot Path Planning

Path planning for aerial robots resembles path planning for any robot, with the following distinctive characteristics: aerial robots are able to fly very fast (or may have to fly fast). Thus there is the distinct possibility of significant discrepancies between intended and actual trajectories. The vehicle dynamics must be fully accounted for when designing trajectories. Several path planning concepts have been proposed to handle this problem, including [44.65–67] and many others. Another key issue in aerial robot trajectory planning arises when there is a discrepancy between the complexity of the environment and the maneuvering space needed for the vehicle. When planned for finite time or geographical horizons, it becomes important that a planner constantly keep a feasible loitering solution within the known environment [44.46].

#### Multirobot Path Planning and Coordination

There has recently been a surge in research activities for multivehicle path planning and coordination. Such research activities have been motivated by problems as diverse as the generation of noncolliding paths, the generation of swarming behaviors for applications such as phased-array, robot-borne antenna systems, collaborative target detection and prosecution, and collaborative search for thermal currents.

This rich literature, of which only a few references have been cited, stems from the conjunction of several constraints in the problem under study, including

- highly constrained dynamical systems (with restricted radius of curvatures and minimum speed requirements, for example)
- a variety of information management possibilities (including centralized, decentralized, distributed information)
- catastrophic consequences in case of failures

Initial work aimed at studying aerial-robot coordination from the perspective of mission execution include [44.68–71]. Swarming behaviors, or the ability for a vehicle group to generate a coherent, consensual behavior using only local information, has become the focus of much attention in the research community since the recent paper [44.72].

Collision avoidance has also formed the motivation for much research in multirobot coordination, see for example [44.73–75].

### 44.7.6 Integrated Aerial Robotic Operations: Aerial Robotics Contests

Research on the ability of aerial robots to perform completely autonomous missions, especially at low altitudes, is clearly well represented in contests such as the International Aerial Robotics Competition, initiated in 1991 by Michelson [44.1]. In this contest, universities, possibly supported by industry and government, compete against each other by demonstrating how their vehicles,
or vehicle systems, meet the requirements of the competition. A basic tenet of the competition is that the small aerial robotic systems entrants must be capable of complete autonomy (no human interaction) during the mission.

The rules of the competition have evolved from the inception of this effort to reflect advances in the capabilities of the proposed systems. One of the key characteristics of the competition is that it has always emphasized the simultaneous demonstration of several robotic functionalities, including basic mission execution, object reconnaissance and detection, and object manipulation. During the early days of the competition, the task asked for an aerial robot to recognize and pick up an object in a designated area, and carry it to another designated area. As universities were able to meet the initial challenges posed by the competition rules, the rules have evolved to a higher level of sophistication. As of today, the competition rules require complete autonomous operation of the vehicles over longer distances. The vehicles must now find and reach a village from a distance of three kilometers. They must also evolve towards higher reasoning capabilities about the objects and events being encountered. Moreover, emphasis has been placed on multimodal robotics, since the robotic system must be able to enter a building and explore it, a task currently best performed by ground robots.

Recognizing the growing gap between experienced participants and new entrants, several different competition levels have been established. While US participation in the competition is predominant, several non-US participants are also present, including Germany, England, Switzerland, Canada, and India. In 2000, the Technische Universitaet Berlin won the contest. Other contest winners include Carnegie-Mellon University, the Georgia Institute of Technology, MIT/Draper Laboratory, and Stanford University.

Other aerial robotic competitions have since been established. For example, the French governmental organization Delegation Generale pour L’Armement (DGA), together with the Supaero and Ecole Nationale Superieure Des Constructions Aeronautiques (ENSICA) engineering schools have proposed a contest involving very small-sized aerial robots in 2004, with a focus on their flight mechanics at various flight regimes. The government of Queensland, Australia together with the local research organizations Commonwealth Scientific and Industrial Research Organization (CSIRO) and Queensland University of Technology launched a new contest focusing on search-and-rescue missions in 2007.

The distribution of vehicle types involved in these contests is very different from the distribution of operational aerial robots. While operational aerial robotic systems are overwhelmingly of fixed-wing type, the machines used by universities during these contests offer a much more balanced distribution of aircraft and rotorcraft. Several reasons contribute to these differences and they have been outlined earlier. The operation of fixed-wing aircraft at relatively high altitude, for reconnaissance and surveillance missions offers a large and technologically easy market to reach, although it faces significant regulatory constraints. In comparison, the operation of small vehicles in cluttered environments definitely favors hovering-like vehicles. However, these vehicles, like other robots, face significantly more constraints in terms of environment sensing, obstacle avoidance, and task planning and execution complexity. As such, they are closer to the realm of basic research typical of universities.

### 44.8 Conclusions and Further Reading

Aerial robots represent a very interesting and exciting area of robotics, involving very dynamic platforms whose size ranges from a few centimeters to several tens of meters. It seems highly probably they will continue to see new applications, beginning with those that happen in relatively unpopulated areas and relatively high altitudes. The current applications of aerial robots are focused primarily on military operations. However, an ambitious civilian market led by Japan is currently burgeoning.

Aerial robots currently pose a challenge to all regulatory agencies, which must find modalities and rules to insert them into airspace occupied by other traffic such as manned systems. The resulting technical research challenges include the development of a proper and affordable sense-and-avoid technology, and the ability for aerial robots to be conversant with other traffic and the ground control infrastructure.

Lower-altitude aerial robotics, often operating in cluttered environments, offers the opportunity to explore many generic robotics topics, including vision, path planning, mapping, and other algorithms in a progressive manner, while offering potential benefit of still more important applications.
A very dynamic research and development field, aerial robotics can be seen from a historical perspective by reading [44.3]. A snapshot of current UAV technology can be obtained, for example, from [44.76]. The lack of a known comprehensive, book-like presentation of aerial robotics and its applications clearly indicates that the field is still very young, that operational experience is slowly building up, and that many challenges, most notably regulatory and safety challenges, must still be overcome.

References

44.1 R.C. Michelson: International aerial robotics competition – The world’s smallest intelligent flying machines, 13th RPVs/UAVs International Conference, UK (1998) pp. 31.1–31.9
44.2 J.D. Anderson: Samuel Pierpont Langley – America's first aeronautical engineer, 38th AIAA Aerospace Sciences Meeting and Exhibit, Reno (2000) paper 0164
44.3 L.R. Newcome: Unmanned Aviation, a Brief History of Unmanned Aerial Vehicles (American Institute of Aeronautics and Astronautics, Reston 2004)
44.4 R.E. Weibel, R.J. Hansman: Safety considerations for operation of different classes of UAVs in the national airspace system, AIAA 3rd Unmanned Unlimited Technical Conference, Chicago (2004) paper 6244
44.6 D. Hughes: UAVs face hurdles in gaining access to civil airspace, Aviation week (2007)
44.11 S.E. Wright: UAVs in Community Police Work, AIAA Infotech., Aerospace, Arlington (2005) paper 6955
44.12 T. McGeer, S. Sliwa, D. Sliwa: Commercial applications for UAVs within the fishing industries, AIAA’s 1st Technical Conference and Workshop on Unmanned Aerospace Vehicles, Portsmouth (2002) paper 3401
44.15 North Atlantic Treaty Organization: Standard Interfaces of UAV Control System (UCS) for NATO UAV Interoperability, NATO STANAG, 4586 (2004)
44.18 J.D. Anderson: Introduction to flight (McGraw-Hill, New York 1989)
44.23 J. Sanderson: Private Pilot Manual (Jeppesen Sanderson, Englewood 1984)
44.25 N. Mayer: Lighter-than-air systems, Aerosp. Am. 12, 30–31 (2005)
Part F


44.28 A. Conway: Autonomous Control of an unstable model helicopter using carrier-phase GPS only. Ph.D. Thesis (Stanford University, Stanford 1995)


44.30 B. Anderson, J.B. Moore: Optimal Filtering (Prentice-Hall, Upper Saddle River 1979)


44.32 A.E. Bryson: Control of Spacecraft and Aircraft (Princeton Univ. Press, Princeton 1994)


44.49 K. Nonami: Personal communication (2003)


44.51 P. Corke: An inertial and visual sensing system for a small autonomous helicopter, J. Robot. Syst. 21, 43–51 (2004)

44.52 S. Saripalli, G.S. Sukhatme: Landing on a moving target using an autonomous helicopter, International Conference on Field and Service Robotics, Mt Fuji (2003)


44.55 T. Merz, S. Duranti, G. Conte: Autonomous landing of an unmanned helicopter based on vision and inertial sensing. In: Experimental Robotics, ed.


This article presents a survey on publicly available open-source projects (OSPs) on quadrotor unmanned aerial vehicles (UAVs).

Recently, there has been increasing interest in quadrotor UAVs. Exciting videos have been published on the Internet by many research groups and have attracted much attention from the public [1]–[7]. Relatively simple structures of quadrotors has promoted interest from academia, UAV industries, and radio-control (RC) hobbyists alike. Unlike conventional helicopters, swashplates, which are prone to failure without constant maintenance, are not required. Furthermore, the diameter of individual rotors can be reduced as a result of the presence of four actuators [8].

Many research groups or institutions have constructed their own quadrotors to suit specific purposes. Successes have been reported from academia, such as the X4-flyer [9], OS4 [10], STARMAC [11], and Pixhawk [12] to mention a few. To the commercial market, the Draganflyer X4, Asctec Hummingbird, Gauí Quad flyer, Parrot ARDrone, and DJI Wookong have been introduced. At the same time, a number of OSPs for quadrotors have emerged as shown in Figure 1, with contributions from RC hobbyists, universities [12], [13], and corporations.
Quadrotor OSPs use community-hosting sites (e.g., Google code and Github) to create code, blueprints, or schematics, which are freely available under open-source licenses such as the general public license (GPL) [14]. These tools help talented independent developers to join OSPs freely from all over the world. This setting allows very fast development processes because new features can be tested not only by the developer but also by other people in the community. Feedback is given in real time from various conditions and configurations, which make open-source software more robust in a relatively short period of time.

OSPs have been successful in many disciplines and remain competitive with commercial alternatives with Linux being the most famous operating system. In the robotics area, more than 2,000 projects have been established based on the robot operating system [15] with its well-organized framework that encourages open-source development.

In the case of quadrotor OSPs, one of the main reasons to use them is flexibility in both hardware and software, which makes modification easier to meet the specific requirements of a user. In addition, OSPs allow researchers to replicate and extend the results of others and provide a baseline for comparison among various approaches.

In this article, we introduce eight quadrotor OSPs and compare them in terms of hardware and software to provide a compact overview for use in a variety of areas as well as in academic research.

Open-Source Projects for Quadrotor UAVs

In this section, we introduce the quadrotor OSPs that are listed in Table 1. These projects have been selected based on the user volume, activity, and project maintenance status. All of these projects are still in development. Therefore, readers should note that the information described in this article is as of May 2012. We use the term OSP to refer to code, electronics, or auxiliary software such as the ground-control software (GCS), depending on the context.

Arducopter

Arducopter is a quadrotor autopilot project based on the Arduino framework developed by individual engineers worldwide, which is described earlier [Figure 1(a)]. A graphical-user-interface (GUI)-based software GCS is provided to tune control gains and display flight information [Figure 2(a)]. This project shares the same avionics platform with Ardupilot, which is a fixed-wing aircraft autopilot OSP. A helicopter autopilot is also supported. There are more than 30 contributors on the project Web site and it uses the GNU Lesser GPL (LGPL) [16].

Openpilot

Openpilot is an OSP led by RC hobbyists [Figure 1(b)] using GPL [14]. This project features a real-time operating system modified from FreeRTOS, which is an open-source operating system. Openpilot supports fixed-wing aircraft and helicopters with the same autopilot avionics. A GUI-based GCS is provided to tune

Table 1. OSPs on quadrotor autopilot.

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Web site URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arducopter</td>
<td><a href="http://code.google.com/p/arducopter">http://code.google.com/p/arducopter</a></td>
</tr>
<tr>
<td>Openpilot</td>
<td><a href="http://www.openpilot.org">http://www.openpilot.org</a></td>
</tr>
<tr>
<td>Paparazzi</td>
<td><a href="http://paparazzi.enac.fr">http://paparazzi.enac.fr</a></td>
</tr>
<tr>
<td>Pixhawk</td>
<td><a href="http://pixhawk.ethz.ch">http://pixhawk.ethz.ch</a></td>
</tr>
<tr>
<td>Mikrokopter</td>
<td><a href="http://www.mikrokopter.de">http://www.mikrokopter.de</a></td>
</tr>
<tr>
<td>KKmulticopter</td>
<td><a href="http://www.kkmulticopter.com">http://www.kkmulticopter.com</a></td>
</tr>
<tr>
<td>Multiwii</td>
<td><a href="http://www.multiwii.com">http://www.multiwii.com</a></td>
</tr>
<tr>
<td>Aeroquad</td>
<td><a href="http://www.aeroquad.com">http://www.aeroquad.com</a></td>
</tr>
</tbody>
</table>
gains and receive flight data [Figure 2(b)]. On the Web site of this project, various videos are available describing gain tuning, assembly processes, and flight principles to help users.

**Paparazzi**

Paparazzi is an autopilot system oriented toward inexpensive autonomous aircraft of all types [Figure 1(c)]. It has been in development since 2003 [13]. Originally a fixed-wing autopilot, it now supports quadrotor configurations by modifying the control mixing rule. Nine different autopilot hardware systems are developed under the lead of the Paparazzi team at ENAC University. Paparazzi provides GUI-based GCS with flight-path scripting that makes mission planning in outdoors convenient [Figure 2(c)]. This project uses GPL for hardware and software.

**Pixhawk**

Pixhawk [12] uses onboard computer-vision algorithms developed by ETHZ Computer Vision Group [Figure 1(e)]. Among the projects introduced here, only the Pixhawk project has computer vision equipment which has been used in several papers [17]–[19]. It also provides GUI-based GCS [Figure 2(d)] called Qgroundcontrol, which is a separate OSP in collaboration with the MAVlink protocol project. The Pixhawk project is available under the GPL.

**Mikrokopter**

Mikrokopter is a quadrotor autopilot system developed by a subsidiary of HiSystems GmbH in 2006 [Figure 1(e)]. GUI-based software for gain tuning and health monitoring is provided as shown in Figure 2(e). Mikrokopter is operated in well-organized Internet shops for their autopilot boards. In 2010, the University of Tasmania and the Australian Antarctic Division made use of Mikrokopter to monitor moss beds in Antarctica.
The source code is available for noncommercial purposes only.

**KKmulticopter**
KKmulticopter is contributed by 20 people around the world [Figure 1(f)]. This project has targeted hobbyists who want to capture aerial photographs using quadrotors. The autopilot hardware of this project is the most basic among the projects described in this article. It is equipped only with a triaxis gyroscope for inertial measurement and an 8-b microcontroller for control. No GCS is provided and gains are tuned by variable resistors on board.

**Multiwii**
Multiwii is a quadrotor autopilot system developed by RC hobbyists [Figure 1(g)]. This project uses an Arduino board as a main processor while the sensor system can vary. This project aims to make the fabrication of electronics easy. It uses gyroscopes and accelerometers of the commercial off-the-shelf Wii motion controller from Nintendo, which needs less soldering. GUI-based GCS is provided as shown in Figure 2(f). GPL is used for this project.

**Aeroquad**
Aeroquad is a quadrotor autopilot based on Arduino [Figure 1(h)]. Similar to Multiwii, it uses a standard Arduino board instead of making its own fully fledged single board. Aeroquad also provides GUI-based GCS software as shown in Figure 2(g). Arducopter was separated from this project in May 2010. GPL is used by Aeroquad.

**Arduino Platform**
Although the Arduino platform is not a quadrotor autopilot, we introduce it here because the Arducopter, Multiwii, and Aeroquad projects all use it. Arduino is the name of both the open-source single-board microcontroller circuit and the integrated development environment (IDE). Arduino has a well-organized device driver library for different sensors and actuators. It is frequently used for rapid prototyping because of the following advantages:

- **Simple setup:** Arduino’s IDE is easy to install, and firmware can be easily downloaded via USB or RS-232 without an expensive JTAG interface.
- **Rich device drivers:** There are more than 100 libraries related to hardware peripherals and signal analysis on the Arduino platform.
- **Operating systems supported:** The Arduino development IDE is ported to Mac OS X, Windows, and Linux.

**Components of Open-Source Projects for Quadrotor UAVs**

Figure 3 shows overall configuration of a typical quadrotor UAV, which consists of flight avionics, sensor systems, radio transmitters, receivers, and communication systems.
Flight Avionics

Flight avionics for the various projects mentioned in the section “Open-Source Projects for Quadrotor UAVs” are shown in Figure 1. Most of the introduced projects provide electronic schematics for self-production. Typically, flight avionics consists of a processor, input/output (I/O) pins, and sensors. The I/O pins connect an off-the-shelf electronic speed controller (ESC) and RC receiver to the flight controller. The sensor suite consists of a gyroscope, accelerometer, barometer, magnetometer, and global positioning system (GPS).

Table 2 describes flight avionics composition. Most flight avionics are full-fledged with six degrees of freedom (6DoF) inertial measurement unit (IMU), magnetometer,
and barometer. However, KKmulticopter has only three gyroscopes, because it is devoted to manual flight. KKMulticopter implements a stability augmented system (SAS), which will be discussed in the section “Open-Source Projects Internals.” Most flight controllers implement proportional-integral-derivative (PID) control for stabilization of the quadcopter, although the structure of the PID controllers between the projects varies slightly. This will also be discussed in detail in the section “Open-Source Projects Internals.”

Sensors
Detailed specification of the sensors used in the OSPs is given in Tables 3–6. For the accelerometer, five different chips are used in the OSPs, which are shown in Table 3. For the gyroscope, there are seven different chips used as listed in Table 4. Magnetometers are used to correct attitude information and estimate drift of gyroscopes. There are three different magnetometers used in the OSPs as shown in Table 5. Three types of barometers used to measure altitude are shown in Table 6.

Radio Transmitters and Receivers
Recently, some groups have modified off-the-shelf RC transmitters to fit their requirement such as complex control mixing or curve shaping of a stick. As a result, custom firmwares for a few RC transmitters have been released as open source. In addition, open-source RC transmitters and receivers have been emerging [Figure 4(a) and (b)]. The OpenLRS project was initiated for open-source RC radio transmitter and receiver development. The OSRC project has developed not only a radio part but also controller hardware as shown in Figure 4(c). These projects are useful when a flight-avionics package needs to be more compact without additional hardware such as an RC receiver.

Communication Systems
XBee is a popular communication system because of its simple setup, low cost, and reasonable communication range when compared with its size. All the projects addressed here use Xbee.

Arducopter and Pixhawk implement the MAVLink protocol for ground control. One advantage of the MAVLink protocol is that one can use Qgroundcontrol without a need to develop separate GCS.

Open-Source Projects Internals

Attitude Estimation
Because a sensor suite is typically composed of a three-axis gyroscope and a three-axis accelerometer, which provide linear accelerations and angular rates only, a proper attitude estimation algorithm should be employed.

Extended Kalman Filter
The Openpilot and Pixhawk projects have designed an attitude estimation algorithm based on the extended Kalman filter (EKF). Here, we provide only an overview of the EKF-based attitude estimation of the Openpilot project, and the complete EKF algorithm can be found in [20].

Let $p$ and $v$ be three-dimensional (3-D) position and velocity in earth-fixed frame, $q$ the quaternion, and $b$ the gyro bias. Let $\mathbf{R}_{pb}(q)$ and $\Omega(q)$ be rotation matrix that converts body-fixed frame to earth-fixed frame and quaternion rates matrix, respectively, as a function of the unit quaternion. Let $a$ denotes linear acceleration in body-fixed frame and $\omega$ the angular velocity in body-fixed frame. Then, the state equation in discrete time can be written as

$$x_k = \begin{bmatrix} p_k \\ v_k \\ q_k \\ b_k \end{bmatrix} = \begin{bmatrix} v_{k-1} \\ \mathbf{R}_{pb}(q_{k-1}) \cdot a_{k-1} \\ \frac{1}{2} \Omega(q_{k-1}) \cdot \omega_{k-1} \\ w_{b,k-1} \end{bmatrix}. \quad (1)$$

In (1), the gyro bias $b$ is modeled with noise $w_b$. The system input $u$ consists of measurements of angular velocity $\omega_m$ and linear acceleration $a_m$:

$$u_k = \begin{bmatrix} \omega_{m,k} \\ a_{m,k} \end{bmatrix} = \begin{bmatrix} \omega_k - w_{\omega,k} + b_b \\ a_k - w_{a,k} - \mathbf{R}_{eb}(q_k)[0 \ 0 g]_T \end{bmatrix}. \quad (2)$$

where $w_{\omega}$ and $w_a$ represent noise and $g$ is gravitational acceleration. Substitution of (2) into (1) yields the following nonlinear model:

$$x_k = f(x_{k-1}, u_{k-1}) + w_{k-1} = \begin{bmatrix} v_{k-1} \\ \mathbf{R}_{pb}(q_{k-1})(a_{m,k-1} + w_{a,k-1}) + [0 \ 0 g]_T \\ \frac{1}{2} \Omega(q_{k-1})(\omega_{m,k-1} + w_{\omega,k-1} - b_k) \\ w_{b,k-1} \end{bmatrix}. \quad (3)$$

Figure 4. Open-source RC transmitters and receivers. (a) OpenLRS transmitter, (b) openLRS receiver, and (c) OSRC transmitter.
where \( w_k = [w_{o,k}, w_{a,k}, w_{b,k}]^T \) is process noise. The nonlinear measurement model is (we omit time index \( k \) for notational simplicity)

\[
z_k = h(x_k) + v_k = \begin{bmatrix} p \\ m_b \\ h_b \end{bmatrix} = \begin{bmatrix} p \\ R_p^T q m_c \\ -P_z \end{bmatrix},
\]

where \( m_b \) is the measurement of the magnetic field of the earth \( m_c \) in body frame, \( h_b \) is the height measured by the barometric sensor reading \( P_z \), and \( v_k \) is the measurement noise.

The states are estimated by the standard EKF algorithm and measurements from accelerometers, gyroscopes, magnetometers, GPS, and barometer are fused to estimate the states.

**Linear Complementary Filter**

The Mikrokopter project implements the linear complementary filter (LCF) and is shown in Figure 5 on each axis of the accelerometer and gyroscope. It is designed to fuse multiple independent noisy measurements of the same signal that have complementary spectral characteristics. The details of complementary filters can be found in [21] and [22].

Let \( y_u \) be the rate measurement of the angle \( \theta \) and \( y_x \), the angle measured by accelerometer. The complementary filter to estimate the angle \( \theta \) is given by

\[
\dot{\theta} = y_u + k_p(y_x - \hat{\theta}),
\]

where \( \hat{\theta} \) denotes the estimate of \( \theta \) and \( k_p \) is a gain that determines crossover frequency. The complementary filter described in (5) assumes that there is no steady-state estimation error. However, in practice, the gyro bias varies over time. To compensate for this, an integrator [Figure 5(b)] is added to obtain the following:

\[
\dot{\hat{\theta}} = y_u - \hat{\theta} + k_p(y_x - \hat{\theta}),
\]

\[
\dot{\hat{\theta}} = -k_p(y_x - \hat{\theta}).
\]

**Nonlinear Complementary Filters on the SO(3) Group**

An LCF is extended to the nonlinear SO(3) group [22] (Figure 6). The final form of the filter with bias estimate is given by

\[
\dot{\hat{R}} = \hat{R} (\Omega_y - \hat{\theta} + \lambda) \times \quad (8)
\]

\[
\dot{\hat{\theta}} = -\pi_\theta (\hat{\theta}_y) \times \quad (9)
\]

\[
\hat{\theta}_y = \pi_\theta (\hat{\theta}_y) \times \quad (10)
\]

where \( \pi_\theta (\hat{\theta}_y) = 1/2(\hat{\theta}_y - \hat{\theta}_y^T) \) and \( \hat{\theta}_y \in SO(3) \) are attitude estimate and estimate error, respectively, and the vex operator is the inverse operation of a skew-symmetric matrix.

**Controllers**

It is well known that the open-loop rotational dynamics of a quadrotor are unstable as studied in [23]. The identified model reveals that poles are located in the right-half plane.
of the real-imaginary axis and damping ratio is negative. Therefore, it needs to be stabilized by a feedback control algorithm, for example, SASs [24].

SAS, which makes the aircraft stable via the rate measurement in the feedback loop, is popular in aircraft control [25]. SAS is shown in Figure 8 with a dotted-line box. It consists of rate feedback with gain. If SAS is applied to a quadrotor, damping is increased. As a result, the quadrotor becomes controllable by a user.

The KKmulticopter project implements its SAS exactly as shown in Figure 9(f). Consequently, it has only three gyroscopes. Because the SAS only provides rate regulation, an autopilot is required to maintain the attitude of a quadrotor. We describe different autopilot structures. We begin with a proportional-derivative controller to illustrate how different architectures produce different characteristics.

The Pixhawk project implements a single-feedback loop as shown in Figure 10(a). In this case, the controller is

\[ G_c(s) = K_P + K_D s. \] (11)

Then, the unity-feedback closed-loop transfer function is

\[ \frac{Y}{R} = \frac{(K_P + K_D s)G_P}{1 + (K_P + K_D s)G_P}. \] (12)

Desirable closed-loop poles can be achieved by adjusting \( K_P \) and \( K_D \). In addition to the noise problem due to differentiation, there is now a

Figure 7. Attitude estimates obtained from the Vicon system plotted in black dashed curves, and the NCF on the SO(3) group plotted in red solid curves.

Figure 8. General attitude autopilot configuration [24]. Vehicle stability is enhanced by the SAS and the vehicle attitude is controlled by the outer loop with an integrator.

Figure 9. Various PID control structures of the OSPs. (a) Arducopter, (b) Openpilot, (c) Paparazzi, Multiwii (d) Pixhawk, Aeroquad, (e) Mikrokopter, and (f) KKMulticopter.
closed-loop zero at \( s = -K_P / K_D \), and this zero can cause a large overshoot in case of a step disturbance unless the plant poles are heavily damped.

To resolve the above problem, consider the inner-loop configuration of Figure 10(b),

\[
G_c = K_P, H_i = K_D s.
\]  

Then, the close-loop transfer function is

\[
\frac{Y}{R} = \frac{K_P G_P}{1 + (K_P + K_D s) G_P}. \tag{14}
\]

Therefore, with rate feedback, we can achieve the same closed-loop poles, but without an undesirable zero. Gyroscopes directly provide the rate information of the aircraft attitude, so this particular compensator structure is useful. All the other OSPs employ inner-loop configurations, while Pixhawk uses the single-loop configuration.

The Arducopter project implements a controller based on an inner-/outer-loop structure as shown in Figure 9(a). To mitigate steady-state error, an integrator is added in the forward path. The controller of the Openpilot project [Figure 9(b)] closely resembles Figure 8, which is proposed in [24]. The Paparazzi project also has the same structure, but the derivative term in the forward path is removed as shown in Figure 9(c). Therefore, Paparazzi controller is exactly the same as Figure 8. The Multiwii project has the same control configuration as Paparazzi. The Pixhawk project implemented a standard PID controller as shown in Figure 9(d). The rate measurement is not used in this controller. Only the derivative of the error signal is used to provide control inputs to the plant. As mentioned earlier, this controller may not be suitable for dynamic maneuvering, because a step change in the reference input will cause an undesirable initial spike in the control signal. However, Pixhawk is designed to support indoor navigation with relatively slow motion.

so it is suitable for this purpose. Mikrokopter has only a proportional-integral (PI) controller in the forward path as shown in Figure 9(e).

As mentioned earlier, the KKmulticopter project only implements SAS that controls the body rate as shown in Figure 9(f). This controller is particularly useful for system identification experiments. This controller only depends on the gyroscope output: \( u = -K_y \). In this case, we can perform closed-loop system identification to obtain an open-loop system model. Consider the following linear system:

\[
\dot{x} = Ax + Bu, \quad y = Cx, \quad u = -K_y. \tag{15}
\]

Then, the closed-loop system becomes \( \dot{x} = A_{cl} x \), where \( A_{cl} = A - BKC \). Therefore, once we perform closed-loop identification to obtain \( A_{cl} \), the open-loop A matrix can also be obtained. This is only possible when there is no

![Figure 10. Feedback control with two configurations. \( G_c \) denotes the plant and \( H_i \) is the inner-loop controller. (a) Feedback control with a single-loop configuration. (b) Feedback control with an inner-loop configuration.](image)

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Number of Gains on Each Axis</th>
<th>Controller Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arducopter</td>
<td>3 + 1 (antiwindup)</td>
<td>PI + P</td>
</tr>
<tr>
<td>Openpilot</td>
<td>4 + 2 (antiwindup)</td>
<td>PI + PI</td>
</tr>
<tr>
<td>Paparazzi</td>
<td>3 + 1 (antiwindup)</td>
<td>PI + P</td>
</tr>
<tr>
<td>Pixhawk</td>
<td>3 + 1 (antiwindup)</td>
<td>PID</td>
</tr>
<tr>
<td>Mikrokopter</td>
<td>2 + 1 (antiwindup)</td>
<td>PI</td>
</tr>
<tr>
<td>KKmulticopter</td>
<td>1</td>
<td>P</td>
</tr>
<tr>
<td>Multiwii</td>
<td>3 + 1 (antiwindup)</td>
<td>PI + P</td>
</tr>
<tr>
<td>Aeroquad</td>
<td>3 + 1 (antiwindup)</td>
<td>PID</td>
</tr>
</tbody>
</table>

![Table 7. Summarized control structure and the number of gains to be tuned.](image)

![Figure 11. Attitude tracking result of the quadrotor. Arducopter autopilot is tested with ground truth. The delay between reference and roll angle is due to communication delay.](image)
integrator involved and K is fully known, which is not always the case for typical quadrotors in the market.

**Controller Parameters**
Control structure and the number of gains to be tuned in each project are shown in Table 7. KKmulticopter is the simplest one, which has only one gain for tuning. Among many controller configurations, PI+P is dominant. P is for the inner loop (rate feedback), and PI is for the forward attitude error compensation.

**Controller Evaluation**
We have constructed quadrotors using five different autopilots among mentioned OSPs: the Arducopter, Paparazzi, Mikrokopter, KKmulticopter, and Multiwii. Among these projects, Arducopter, Paparazzi, and Multiwii share the same controller composition as shown in Table 7. For qualitative evaluation, we mount markers on a quadrotor to acquire ground-truth data from the Vicon system. The desired angle is transmitted to the Arducopter-based quadrotor while quadrotor attitude from the Vicon and the transmitted commands are recorded simultaneously. The satisfactory attitude tracking result is shown in Figure 11. The delay is due to RC signal processing.

**Selection Guidelines**
We have analyzed eight OSPs with attitude estimation algorithm, control configuration, electronic components, and features. A comparison of features between OSPs is given in Table 8.

**Availability of Flight Avionics**
All the projects we described provide electronic schematic and bill of materials to reproduce their flight avionics. However, it takes high initial cost to manufacture electronics individually. Only five projects among them are available for purchase now: Arducopter, Paparazzi, Mikrokopter, KKmulticopter, and Aeroquad. It is recommended to start with these projects if a reader prefers to avoid electronics fabrication.

![Figure 11.](image-url)

**Figure 11.** Pixhawk-based quadrotor platform with a camera and onboard computer [31]. (a) Pixhawk quadrotor platform and (b) flight environment of the Pixhawk quadrotor with ARToolkit markerboard on the floor.

<table>
<thead>
<tr>
<th>Attitude estimation algorithm</th>
<th>Arducopter</th>
<th>Openpilot</th>
<th>Paparazzi</th>
<th>Pixhawk</th>
<th>Mikrokopter</th>
<th>KKmulticopter</th>
<th>Multiwii</th>
<th>Aeroquad</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPS-based waypoint navigation</td>
<td>✔️</td>
<td>Δ</td>
<td></td>
<td></td>
<td></td>
<td>Δ</td>
<td>Δ</td>
<td></td>
</tr>
<tr>
<td>Altitude hold</td>
<td>✔️</td>
<td>Δ</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Hardware in the loop simulation</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Support of other multirotor airframes</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td></td>
<td>✔️</td>
<td>✔️</td>
<td>□</td>
</tr>
<tr>
<td>Support of computer vision</td>
<td></td>
<td>□</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td></td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>GCS provided</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Airframe design provided</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Camera stabilization</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td></td>
<td>✔️</td>
<td>✔️</td>
<td>□</td>
</tr>
<tr>
<td>Availability</td>
<td>✔️</td>
<td>□</td>
<td>✔️</td>
<td></td>
<td>✔️</td>
<td></td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>Open-source license</td>
<td>LGPL</td>
<td>GPL</td>
<td>GPL</td>
<td>GPL</td>
<td>Δd</td>
<td>GPL</td>
<td>GPL</td>
<td>GPL</td>
</tr>
<tr>
<td>Used by</td>
<td>[26], [27]</td>
<td>□</td>
<td>□</td>
<td>[28], [12]</td>
<td>□</td>
<td>[29], [30]</td>
<td>□</td>
<td>□</td>
</tr>
</tbody>
</table>

✔️: supported, □: partially supported (e.g., additional navigation electronics), —: not supported.

*Only GPS-based homing is supported.

*The project provides a quadrotor airframe design in computer-aided design files.

*The project avionics on sale.

*Only noncommercial purposes.
Attitude Estimation Algorithm Development

For attitude estimation tests, Arducopter and Paparazzi will be a good choice. The other projects are equipped with two or more gyro chips, which are hard to be calibrated for alignment. Only Arducopter and Paparazzi are equipped with 6-DoF IMU in a single chip: MPU 6000. The dynamic range is the best among the accelerometers and gyroscopes as described in Tables 3–6.

Minimalistic Configuration

As studied in [23], an open-loop model can be easily identified when control input is fully known and no integrators exist in a controller as mentioned in the section “Attitude Estimation Algorithm Development.” Because SAS is implemented in [23] to identify the open-loop dynamics, KKmulticopter is a good choice to this end. The system is simple to understand and modify because a source code for attitude control is less than 500 lines in C.

GPS-Based Navigation

For GPS-based outdoor missions (e.g., waypoint navigation and hovering): Arducopter, Openpilot, Paparazzi, or Mikrokopter will be a good choice. Only these projects support GPS-based navigation. Although Multiwii has GPS, it only supports a homing capability to move a quadrotor back to the initial position.

Vision-Based Navigation

Only the Pixhawk project supports vision-based navigation capability. It can synchronize an IMU and a camera in hardware level, which allows tight integration of IMU measurements into the computer vision pipeline.

Open-Source Projects in Research

Vision-Based Navigation

The Pixhawk UAV is designed to be a research platform for computer-vision-based autonomous flight [28]. The Pixhawk team has constructed a localization test setup using augmented reality ToolKit+ (ARToolKit+). They successfully performed waypoint navigation using a camera on the localization test bed as shown in Figure 12.

In [26], adaptive image-based visual serving (IBVS) was integrated with adaptive sliding mode control based on Arducopter. Figure 13 shows the experiment in process where the inset picture is the image obtained from the onboard camera. The fiducial marker and its tracking result are shown.

Real-time vision-based localization was performed on a quadrotor system based on Arducopter [Figure 14(a)] [27]. This quadrotor is equipped with a frontal-view grayscale USB2.0 camera with 640 × 480 pixel resolution. Image data from the camera are transferred to a single-board computer and processed in a real time [Figure 14(b)] to obtain the vehicle location based on a map created in advance.

Indoor Flight

A Mikrokopter-based quadrotor flew autonomously using a laser range finder (LRF) [29]. Equipped with LRF, Gumstix, and external IMU, it successfully performed autonomous indoor navigation without external localization sensors. Indoor position control based on an onboard LRF was performed on the Mikrokopter-based quadrotor platform shown in Figure 15 [32]. An autoregressive moving average with exogenous terms model of the stabilized Mikrokopter was identified in [30]. Recently, the quadrotor platform with shared autonomy was investigated for infrastructure inspection [33].

Multiagent-related research can be easily performed on the indoor quadrotor flight system. Especially, as the communication topology between agents can be user-defined within the GCS, various settings and algorithms can be exploited. Figure 16 shows three quadrotors in flight where an auction algorithm is being tested for online task
assignment. As described in Figure 17, each quadrotor is equipped with an onboard controller to track input commands sent by the GCS that collects position and/or attitude data of the quadrotors from the Vicon motion capture system. Data from the onboard vision sensors are sent to the GCS using a dedicated communication link.

**Conclusions**

This article has presented eight quadrotor OSPs with descriptions of their avionics, sensor composition, analysis of attitude estimation and control algorithms, and comparison of additional features. Several research projects that use OSPs as a main flight controller are described.

Among the eight OSPs summarized in this article, we have implemented five by utilizing the benefit of OSPs that allow to build own systems at a low cost with less effort. To bring out continued improvements based on communities’ work, objective evaluations of OSPs remain an important open problem.

The meaning of OSP had been more about software, but it is expanding to hardware and even products. There is already a project that has open hardware blueprints and a 3-D model of the quadrotor airframe that can be ordered from 3-D printing services. Sharing the same platform will become easier with such services. We expect that more OSPs for UAV will be initiated in the future.
### Acknowledgments
This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (nos. 20120000921 and 2012014219), and by the New and Renewable Energy Program of the Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Korea government Ministry of Knowledge Economy (no. 20104010100490).

### References


**Hyon Lim**, Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul 151-742, South Korea. E-mail: hyonlim@snu.ac.kr.

**Jaemann Park**, Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul 151-742, South Korea. E-mail: pjm7981@snu.ac.kr.

**Daewon Lee**, Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul 151-742, South Korea. E-mail: dwsh001@snu.ac.kr.

**H.J. Kim**, Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul 151-742, South Korea. E-mail: hjinkim@snu.ac.kr.
his article provides a tutorial introduction to modeling, estimation, and control for multirotor aerial vehicles that includes the common four-rotor or quadrotor case.

Aerial robotics is a fast-growing field of robotics and multirotor aircraft, such as the quadrotor (Figure 1), are rapidly growing in popularity. In fact, quadrotor aerial robotic vehicles have become a standard platform for robotics research worldwide. They already have sufficient payload and flight endurance to support a number of indoor and outdoor applications, and the improvements of battery and other technology is rapidly increasing the scope for commercial opportunities. They are highly maneuverable and enable safe and low-cost experimentation in mapping, navigation, and control strategies for robots that move in three-dimensional (3-D) space. This ability to move in 3-D space brings new research challenges compared with the wheeled mobile robots that have driven mobile robotics research over the last decade. Small quadrotors have been demonstrated for exploring and mapping 3-D environments; transporting, manipulating, and assembling objects; and acrobatic tricks such as juggling, balancing, and flips. Additional rotors can be added, leading to generalized N-rotor vehicles, to improve payload and reliability.

Multirotor Aerial Vehicles

Modeling, Estimation, and Control of Quadrotor

By Robert Mahony, Vijay Kumar, and Peter Corke
This tutorial describes the fundamentals of the dynamics, estimation, and control for this class of vehicle, with a bias toward electrically powered micro (less than 1 kg)-scale vehicles. The word *helicopter* is derived from the Greek words for spiral (screw) and wing. From a linguistic perspective, since the prefix quad is Latin, the term quadrotor is more correct than quadcopter and more common than tetra-rotor; hence, we use the term *quadrotor* throughout.

**Modeling of Multirotor Vehicles**
The most common multirotor aerial platform, the quadrotor vehicle, is a very simple machine. It consists of four individual rotors attached to a rigid cross airframe, as shown in Figure 1. Control of a quadrotor is achieved by differential control of the thrust generated by each rotor. Pitch, roll, and heave (total thrust) control is straightforward to conceptualize. As shown in Figure 2, rotor \( i \) rotates anticlockwise (positive about the \( z \) axis) if \( i \) is even and clockwise if \( i \) is odd. Yaw control is obtained by adjusting the average speed of the clockwise and anticlockwise rotating rotors. The system is underactuated, and the remaining degrees of freedom (DoF) corresponding to the translational velocity in the \( x-y \) plane must be controlled through the system dynamics.

**Rigid-Body Dynamics of the Airframe**
Let \( \{\tilde{x}, \tilde{y}, \tilde{z}\} \) be the three coordinate axis unit vectors without a frame of reference. Let \( \{A\} \) denote a right-hand inertial frame with unit vectors along the axes denoted by \( \{\tilde{a}_1, \tilde{a}_2, \tilde{a}_3\} \) expressed in \( \{A\} \). One has algebraically that \( \tilde{a}_1 = \tilde{x}, \tilde{a}_2 = \tilde{y}, \tilde{a}_3 = \tilde{z} \) in \( \{A\} \). The vector \( r = (x, y, z) \in \{A\} \) denotes the position of the center of mass of the vehicle. Let \( \{B\} \) be a right-hand body fixed frame for the airframe with unit vectors \( \{\tilde{b}_1, \tilde{b}_2, \tilde{b}_3\} \), where these vectors are the axes of frame \( \{B\} \) with respect to frame \( \{A\} \). The orientation of the rigid body is given by a rotation matrix \( R = R_b = [\tilde{b}_1, \tilde{b}_2, \tilde{b}_3] \in SO(3) \) in the special orthogonal group. One has \( \tilde{b}_1 = R \tilde{x}, \tilde{b}_2 = R \tilde{y}, \tilde{b}_3 = R \tilde{z} \) by construction.

We will use \( X-Y-Z \) Euler angles to model this rotation, as shown in Figure 3. To get from \( \{A\} \) to \( \{B\} \), we first rotate about \( \tilde{a}_3 \) by the the yaw angle, \( \psi \), and we will call this intermediary frame \( \{E\} \) with a basis \( \{\tilde{e}_1, \tilde{e}_2, \tilde{e}_3\} \) where \( \tilde{e}_i \) is expressed with respect to frame \( \{A\} \). This is followed by a rotation about the \( x \) axis in the rotated frame through the roll angle, \( \phi \), followed by a third pitch rotation about the new \( y \) axis through the pitch angle \( \theta \) that results in the body-fixed triad \( \{\tilde{b}_1, \tilde{b}_2, \tilde{b}_3\} \)

\[
R = \begin{pmatrix}
c \psi c \theta - s \psi s \phi s \theta & -c \psi s \phi + c \phi s \psi s \theta & c \phi c \psi + s \psi c \phi s \theta \\
c \phi s \psi + c \psi s \phi \theta & c \phi c \psi - s \psi c \phi s \theta & -c \phi c \psi + s \psi c \phi s \theta \\
- c \psi s \theta & s \psi c \psi s \theta & c \phi c \theta
\end{pmatrix},
\]

where \( c \) and \( s \) are shorthand forms for cosine and sine, respectively.

Let \( \nu \in \{A\} \) denote the linear velocity of \( \{B\} \) with respect to \( \{A\} \) expressed in \( \{A\} \). Let \( \Omega \in \{B\} \) denote the angular velocity of \( \{B\} \) with respect to \( \{A\} \); this time expressed in \( \{B\} \). Let \( m \) denote the mass of the rigid object, and \( I \in \mathbb{R}^{3 \times 3} \) denote the constant inertia matrix (expressed in the body fixed frame \( \{B\} \)). The rigid body equations of motion of the airframe are [2] and [3]

\[
\dot{\psi} = \nu, \\
\dot{\nu} = mg\tilde{a}_3 + RF, \\
\dot{R} = R\Omega \times, \\
I\dot{\Omega} = -\Omega \times I\Omega + \tau.
\]

The notation \( \Omega \times \) denotes the skew-symmetric matrix, such that \( \Omega \times \nu = \Omega \times \nu \) for the vector cross product \( \times \) and any vector \( \nu \in \mathbb{R}^3 \). The vectors \( F, \tau \in \{B\} \) combine the principal nonconservative forces and moments applied to the quadrotor airframe by the aerodynamics of the rotors.

**Dominant Aerodynamics**
The aerodynamics of rotors was extensively studied during the mid 1900s with the development of manned helicopters, and detailed models of rotor aerodynamics are available in the literature [4], [5]. Much of the detail about these aerodynamic models is useful for the design of rotor systems, where the whole range of parameters (rotor
geometry, profile, hinge mechanism, and much more) are fundamental to the design problem. For a typical robotic quadrotor vehicle, the rotor design is a question for choosing one among five or six available rotors from the hobby shop, and most of the complexity of aerodynamic modeling is best ignored. Nevertheless, a basic level of aerodynamic modeling is required.

The steady-state thrust generated by a hovering rotor (i.e., a rotor that is not translating horizontally or vertically) in free air may be modeled using momentum theory [5, Sec. 2.26] as

\[ T_i := C_T \rho A_i r_i^2 \sigma_i^2, \quad (2) \]

where, for rotor \( i \), \( A_i \) is the rotor disk area, \( r_i \) is the radius, \( \sigma_i \) is the angular velocity, \( C_T \) is the thrust coefficient that depends on rotor geometry and profile, and \( \rho \) is the density of air. In practice, a simple lumped parameter model

\[ T_i = c_T \sigma_i^2 \quad (3) \]

is used, where \( c_T > 0 \) is modeled as a constant that can be easily determined from static thrust tests. Identifying the thrust constant experimentally has the advantage that it will also naturally incorporate the effect of drag on the airframe induced by the rotor flow.

The reaction torque (due to rotor drag) acting on the airframe generated by a hovering rotor in free air may be modeled as [5, Sec. 2.30]

\[ Q_i := c_Q \sigma_i^2, \quad (4) \]

where the coefficient \( c_Q \) (which also depends on \( A_i, r_i \), and \( \rho \)) can be determined by static thrust tests.

As a first approximation, assume that each rotor thrust is oriented in the \( z \) axis of the vehicle, although we note that this assumption does not exactly hold once the rotor begins to rotate and translate through the air, an effect that is discussed in “Rotor Flapping.” For an \( N \)-rotor airframe, we label the rotors \( i \in \{1 \cdots N\} \) in an anticlockwise direction with rotor 1 lying on the positive \( x \) axis of the vehicle (the front), as shown in Figure 2. Each rotor has associated an angle \( \Phi_i \) between its airframe support arm and the body-fixed frame \( x \) axis, and it is the distance \( d \) from the central axis of the vehicle. In addition, \( \sigma_i \in \{-1,+1\} \) denotes the direction of rotation of the \( i \)th rotor: +1 corresponding to clockwise and −1 to anticlockwise. The simplest configuration is for \( N \) even and the rotors distributed symmetrically around the vehicle axis with adjacent rotors counter rotating.

Figure 3. The vehicle model. The position and orientation of the robot in the global frame are denoted by \( \zeta \) and \( R \), respectively.

The total thrust at hover (\( T_\Sigma \)) applied to the airframe is the sum of the thrusts from each individual rotor

\[ T_\Sigma = \sum_{i=1}^{N} T_i = c_T \left( \sum_{i=1}^{N} \sigma_i^2 \right). \quad (5) \]

The hover thrust is the primary component of the exogenous force

\[ F = T_\Sigma \hat{z} + \Delta \quad (6) \]

in (1b), where \( \Delta \) comprises secondary aerodynamic forces that are induced when the assumption that the rotor is in hover is violated. Since \( F \) is defined in \( B \), the direction of application is written \( \hat{z} \), although in the frame \( A \) this direction is \( \hat{b}_3 = R \hat{z} \).

The net moment arising from the aerodynamics (the combination of the produced rotor forces and air resistances) applied to the \( N \)-rotor vehicle use are \( \tau = (\tau_1, \tau_2, \tau_3) \).

\[ \tau_1 = c_T \sum_{i=1}^{N} d_i \sin(\Phi_i) \sigma_i^2, \]

\[ \tau_2 = -c_T \sum_{i=1}^{N} d_i \cos(\Phi_i) \sigma_i^2, \]

\[ \tau_3 = c_Q \sum_{i=1}^{N} \sigma_i \sigma_i^2. \quad (7) \]

For a quadrotor, we can write this in matrix form

\[ \begin{pmatrix} T_\Sigma \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{pmatrix} = \begin{pmatrix} c_T & c_T & c_T & c_T \\ 0 & d c_T & 0 & -d c_T \\ -d c_T & 0 & d c_T & 0 \\ -c_Q & c_Q & -c_Q & c_Q \end{pmatrix} \begin{pmatrix} \sigma_1^2 \\ \sigma_2^2 \\ \sigma_3^2 \\ \sigma_4^2 \end{pmatrix}, \quad (8) \]

and given the desired thrust and moments, we can solve for the required rotor speeds using the inverse of the constant matrix \( \Gamma \). In order for the vehicle to hover, one must choose suitable \( \sigma_i \) by inverting \( \Gamma \), such that \( \tau = 0 \) and \( T_\Sigma = mg \).
Blade Flapping and Induced Drag

There are many aerodynamic and gyroscopic effects associated with any rotor craft that modify the simple force model introduced above. Most of these effects cause only minor perturbations and do not warrant consideration for a robotic system, although they are important for the design of a full-sized rotor craft. Blade flapping and induced drag, however, are fundamental effects that are of significant importance in understanding the natural stability of quadrotors and how state observers operate. These effects are particularly relevant since they induce forces in the \(x,y\) rotor plane of the quadrotor, the underactuated directions in the dynamics, that cannot be easily dominated by high gain control. In this section, we consider a single rotor and we will drop the subscript \(i\) used in the “Dominant Aerodynamics” section to refer to particular rotors.

Quadrotor vehicles are typically equipped with lightweight, fixed-pitch plastic rotors. Such rotors are not rigid, and the aerodynamic and inertial forces applied to a rotor during flight are quite significant and can cause the rotor to flex. In fact, allowing the rotor to bend is an important property of the mechanical design of a quadrotor and fitting rotors that are too rigid can lead to transmission of these aerodynamic forces directly through to the rotor hub and may result in a mechanical failure of the motor mounting or the airframe itself. Having said this, rotors on small vehicles are significantly more rigid relative to the applied aerodynamic forces than rotors on a full-scale rotor craft. Blade-flapping effects are due to the flexing of rotors, while induced drag is associated primarily with the rigidity of the rotor, and a typical quadrotor will experience both. Luckily, their mathematical expression is equivalent and a single term is sufficient to represent both effects in a lumped parameter dynamic model.

When a rotor translates laterally through the air it displays an effect known as rotor flapping (see “Rotor Flapping”). A detailed derivation of rotor flapping involves a mechanical model of the bending of the rotor subject to aerodynamic and centripetal forces as it is swept through a full rotation [5, Sec. 4.5]. The resulting equations of motion are nonlinear second-order dynamical system with a dominant highly damped oscillatory response at the forced frequency corresponding to the angular velocity of the rotor. For a typical rotor, the flapping dynamics converge to steady state with one cycle of the rotor [5, p. 137], and for the purposes of modeling, only the steady-state response of the flapping dynamics need be considered.

Assuming that the velocity of the vehicle is directly aligned with the \(X\) axis in the inertial frame, \(v = (v_x, 0, 0)\), a simplified solution is given by

\[
\beta := -\frac{\mu A_{1c}}{1 - \frac{1}{2} \mu^2} , \quad \beta^\perp := -\frac{\mu A_{1s}}{1 + \frac{1}{2} \mu^2} \tag{9}
\]

for positive constants \(A_{1c}\) and \(A_{1s}\), and where \(\mu := |v_x|/\sigma r\) is the advance ratio, i.e., the ratio of magnitude of the horizontal velocity of the rotor to the linear velocity of rotor tip. The flapping angle \(\beta\) is the steady-state tilt of the rotor away from the incoming apparent wind and \(\beta^\perp\) is the tilt orthogonal to the incident wind. Here, we use equations (4.46) and (4.47) from [5, p. 138], noting that adding the effects of a virtual rotor hinge model [5, Sec. 4.7] results in additional phase lag between the sine and cosine components of the flapping angles [5, Question 4.7, p. 157] that are absorbed into the constants \(A_{1c}\) and \(A_{1s}\) in (9).

Rotor flapping is important because the thrust generated by the rotor is perpendicular to the rotor plane and not to the hub of the rotor. Thus, when the rotor disk tilts the rotor thrust is inclined with respect to the airframe and contains a component in the \(x\) and \(y\) directions of the body-fixed frame.

In practice the rotors are stiff and oppose the aerodynamic force which is lifting the advancing blade so that its increased thrust due to tip velocity is not fully counteracted by a lower angle of attack and lower lift coefficient—the thrust is increased. Conversely for the retreating blade the thrust is reduced. For any airfoil that generates lift (in our case the rotor blade) there is an associated induced drag

\[\begin{align*}
\text{Figure S1} & \\
\text{Rotor Flapping} & \\
\text{When a rotor translates horizontally through the air, the} & \\
\text{advancing blade has a higher absolute tip velocity and} & \\
\text{will generate more lift than the retreating blade. Thinking} & \\
\text{of the rotor as a spinning disk, the mismatch in lift} & \\
\text{generates an overall moment on the rotor disk in the} & \\
\text{direction of the apparent wind (Figure S1). The high} & \\
\text{angular momentum of the rotor disk makes it act like} & \\
\text{a gyroscope, which causes the rotor disk to tilt around} & \\
\text{the axis given by the cross product of rotor hub axis} & \\
\text{and the torque axis, i.e., an axis that is offset from the} & \\
\text{apparent wind by 90° in the horizontal plane of the} & \\
\text{rotor. Since the motor shaft is vertical, the blade} & \\
\text{flaps up as it advances into the wind and back down} & \\
\text{again as it retreats from the apparent wind. Equilibrium} & \\
\text{is established because the advancing blade rises and} & \\
\text{decreases its angle of attack, which reduces its lift} & \\
\text{coefficient, thereby counteracting the additional lift} & \\
\text{that would have been generated due to its increased} & \\
\text{tip velocity. Conversely for the retreating blade, the} & \\
\text{reduced lift due to decreased tip velocity is countered by} & \\
\text{the increased angle of attack and increased thrust} & \\
\text{coefficient. In this state, the rotor will have a} & \\
\text{stable constant tilt away from the apparent wind} & \\
\text{caused by a translational motion of the rotor. This effect} & \\
\text{is known as rotor flapping} & \\
\text{and is ubiquitous in rotor vehicles [6].} & \\
\end{align*}\]
due to the backward inclination of aerodynamic force with respect to the airfoil motion. The induced drag is proportional to the lift generated by the airfoil. In normal hover conditions for a rotor, this force is equally distributed in all directions around the circumference of the rotor and is responsible for the torque $Q$ (4). However, when there is a thrust imbalance, then the sector of the rotor travel with high thrust (for the advancing rotor) will generate more induced drag than the sector where the rotor generates less thrust (for the retreating blade). The net result will be an induced drag that opposes the direction of apparent wind as seen by the rotor, and that is proportional to the velocity of the apparent wind. This effect is often negligible for full scale rotor craft, however, it may be quite significant for small quadrotor vehicles with relatively rigid blades. The consequence of blade flapping and induced drag taken together ensures that there is always a noticeable horizontal drag experienced by a quadrotor even when maneuvering at relatively slow speeds.

We will now use the insight from the discussion above to develop a lumped parameter model for exogenous force generation (6). We assume that all four rotors are identical and rotate at similar speeds so that, at least to a first approximation, the flapping responses of the rotors and the unbalanced aerodynamic forces are the same. It follows that the reactive torques on the airframe transmitted by the rotor masts due to rotor stiffness cancel. For general motion of the vehicle, the apparent wind results in the advance ratio

$$\mu = \sqrt{\mu_x^2 + \mu_y^2 / \sigma_T},$$

where $\mu = R^{-1}v$ is the linear velocity of the vehicle expressed in the body-fixed frame, with $\mu_x$ and $\mu_y$ being the components in the body-fixed $x$-$y$ plane. Define

$$A_{\text{flap}} = \frac{1}{\sigma R} \left( \begin{array}{ccc} A_{1c} & -A_{1s} & 0 \\ A_{1s} & A_{1c} & 0 \\ 0 & 0 & 0 \end{array} \right),$$

where $\sigma$ is the set point for the rotor angular velocity. This matrix describes the sensitivities of the flapping angle to the apparent wind in the body-fixed frame, given that $\mu$ is small and $\mu^2$ is negligible in the denominators of (9). The first row encodes (9) for the velocity along the body-fixed frame $x$ axis. The second row of $A_{\text{flap}}$ is a $\pi / 2$ rotation of this response to account for the case where a component of the wind is incoming from the $y$ axis, while the third row projects out velocity in the $z$ axis of the body-fixed frame.

We model the stiffness of the rotor as a simple torsional spring so that the induced drag is directly proportional to this angle and is scaled by the total thrust. The flapping angle is negligible with regard to the orientation of the induced drag, and in the body-fixed frame the induced drag is

$$D_{\text{ind}} v' \approx \text{diag}(d_x, d_y, 0)v',$$

where $d_x = d_y$ is the induced drag coefficient.

The exogenous force applied to the rotor can now be modeled by

$$F := T_\Sigma \tilde{z} - T_\Sigma Dv',$$  \hspace{1cm} (10)

where $D = A_{\text{flap}} + \text{diag}(d_x, d_y, 0)$, and $T_\Sigma$ is the nominal thrust (5).

An important consequence of blade flapping and induced drag is a natural stability of the horizontal dynamics of the quadrotor [7]. Define

$$P_h := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix},$$

(11)

to be the projection matrix onto the $x$-$y$ plane. The horizontal component of a velocity expressed in $\{A\}$ is

$$v_h := P_h v = (v_x, v_y)^\top \in \mathbb{R}^2.$$

(12)

Recalling (1b) and projecting onto the horizontal component of velocity, one has

$$mv_h = -T_\Sigma P_h (\tilde{z} + RDv').$$

If the vehicle is flying horizontally, i.e., $v_z = 0$, then $v = P_h v_h$ and one can write

$$mv_h = -T_\Sigma P_h (\tilde{z} + RDv').$$

(13)

where the last term introduces damping since, for a typical system, the matrix $D$ is a positive semidefinite.

A detailed dynamic model of the quadrotor, including flapping and induced drag, is included in the robotics toolbox for MATLAB [8]. This is provided in the form of Simulink library blocks along with a set of inertial and aerodynamic parameters for a particular quadrotor. The graphical output of the animation block is shown in Figure 4. Simulink models, based on these blocks, that illustrate path following and vision-based stabilization are described in detail in [1].

The discussion provided above does not consider several additional aerodynamic effects that are important for high-speed and highly dynamic maneuvers for a quadrotor. In particular, we do not consider translational lift and drag that will effect thrust generation at high speed, axial flow modeling and vortex states that may effect thrust during axial motion, and ground effect that will affect a vehicle flying close to the ground. It should be noted that high gain control can dominate all secondary aerodynamic effects, and high
performance control of quadrotor vehicles has been demonstrated using the simple static thrust model [23], [24]. The detailed modeling of the blade flapping and induced drag is provided due to its importance in understanding the state estimation algorithms introduced later the tutorial.

**Size, Weight, and Power (SWAP) Constraints and Scaling Laws**

Reducing the scale of the quadrotor has an interesting effect on the inertia, payload, and ultimately the maximum achievable angular and linear acceleration. To gain insight into scaling, it is useful to develop a simple physics model to analyze a quadrotor’s ability to produce linear and angular accelerations from a hover state.

If the characteristic length is \( d \), the rotor radius \( r \) scales linearly with \( d \). The mass scales as \( d^3 \) and the moments of inertia as \( d^5 \). On the other hand, from (3) and (4), it is clear that the lift or thrust, \( T \), and drag, \( Q \), from the rotors scale with the square of the rotor speed, \( \sigma^2 \). In other words, \( T \sim \sigma^2 d^4 \) and \( Q \sim \sigma^2 d^4 \), the linear acceleration \( a = \dot{v} \), which depends on the thrust and mass, and the angular acceleration \( \omega = \dot{\Omega} \), which depends on thrust, drag, the moment arm, and the moments of inertia, scale as

\[
a \sim \frac{\sigma^2 d^4}{d^5} = \sigma^2 d, \quad \omega \sim \frac{\sigma^2 d^5}{d^6} = \sigma^2.
\]

To explore the scaling of rotor speed with length, it is useful to adopt the two commonly accepted approaches to study scaling in aerial vehicles [9]. Mach scaling is used for compressible flows and essentially assumes that the blade tip speed, \( v_b \), is a constant leading to \( \sigma \sim (1/r) \). Froude scaling is used for incompressible flows and assumes that, for similar aircraft configurations, the Froude number, \( (v_b^2/g) = (\sigma^2 r^2/g) \), is constant. Here, \( g \) is the acceleration due to gravity. Assuming \( r \sim d \), we get \( \sigma \sim (1/\sqrt{r}) \). Thus, Mach scaling predicts

\[
a \sim \frac{1}{d^2}, \quad \omega \sim \frac{1}{d^2}, \quad \text{(14)}
\]

while Froude scaling leads to the conclusion

\[
a \sim 1, \quad \omega \sim \frac{1}{d}. \quad \text{(15)}
\]

Of course, Froude or Mach number similitudes take neither motor characteristics nor battery properties into account. While motor torque increases with length, the operating speed for the rotors is determined by matching the torque–speed characteristics of the motor to the drag versus speed characteristics of the rotors. Further, the motor torque depends on the ability of the battery to source the required current. All these variables are tightly coupled for smaller designs since there are fewer choices available at smaller length scales. Finally, as discussed in the previous subsection, the assumption that rotor blades are rigid may be wrong. Further, the aerodynamics of the blades may be different for blade designs optimized for smaller helicopters and the quadratic scaling of the lift with speed may not be accurate.

In spite of the simplifications in the above similitude analysis, the key insight from both Froude and Mach number similitudes is that smaller quadrotors can produce faster angular accelerations while the linear acceleration is at worst unaffected by scaling. Thus, smaller quadrotors are more agile, a fact that is easily validated from experiments conducted with the Ascending Technologies Pelican quadrotor [10] (approximately 2 kg gross weight when equipped with sensors, 0.75 m diameter, and 5,400 r/min nominal rotor speed at hover), the Ascending Technologies Hummingbird quadrotor [11] (approximately 500 g gross weight, 0.5 m diameter, and 5,000 r/min nominal rotor speed at hover), and laboratory experimental prototypes developed at GRASP laboratory at the University of Pennsylvania (approx. 75 g gross weight, 0.21 m diameter, and approximately 9,000 r/min nominal rotor speed).

**Estimating the Vehicle State**

The key state estimates required for the control of a quadrotor are its height, attitude, angular velocity, and linear velocity. Of these states, the attitude and angular velocity are the most important as they are the primary variables used in attitude control of the vehicle. The most basic instrumentation carried by any quadrotor is an inertial measurement unit (IMU) often augmented by some form of height measurement, either acoustic, infrared, barometric, or laser based. Many robotics applications require more sophisticated sensor suites such as VICON systems, global positioning system (GPS), camera, Kinect, or scanning laser rangefinder.

**Figure 4.** Frame from the Simulink animation of quadrotor dynamics.
Estimating Attitude

A typical IMU includes a three-axis rate gyro, three-axis accelerometer, and three-axis magnetometer. The rate gyro measures the angular velocity of \( \mathcal{B} \) relative to \( \mathcal{A} \) expressed in the body-fixed frame of reference \( \mathcal{B} \)

\[
\Omega_{\text{IMU}} = \Omega + b_\Omega + \eta \in \{B\},
\]

where \( \eta \) denotes the additive measurement noise and \( b_\Omega \) denotes a constant (or slowly time-varying) gyro bias. Generally, the gyroscopes installed on quadrotor vehicles are lightweight microelectromechanical systems (MEMS) devices that are reasonably robust to noise and quite reliable. The accelerometers (in a strap down IMU configuration) measure the instantaneous linear acceleration of \( \mathcal{B} \) due to exogenous force

\[
a_{\text{IMU}} = R^T(\dot{v} - g\bar{z}) + b_a + \eta_a \in \{B\},
\]

where \( b_a \) is a bias term, \( \eta_a \) denotes additive measurement noise, and \( \dot{v} \) is in the inertial frame. Here, we use the notation \( \bar{z} = \tilde{a}_0 \) since we will need to deal with the algebraic expressions of the coordinate axes throughout this section. Accelerometers are highly susceptible to vibration and, mounted on a quadrotor, they require significant low-pass mechanical and/or electrical filtering to be usable. Most quadrotor avionics will incorporate an analogue anti-aliasing filter on a MEMS accelerometer before the signal is sampled.

A commonly used technique to estimate the bias \( b_\Omega \) and \( b_a \) is to average the output of these sensors for a few seconds while the quadrotor is on the ground and the motors are not yet active. The bias is then assumed constant for the duration of the flight.

The magnetometers provide measurements of the ambient magnetic field

\[
m_{\text{IMU}} = R^A m + B_m + \eta_b \in \{B\},
\]

where \( A_m \) is the Earth’s magnetic field vector (expressed in the inertial frame), \( B_m \) is a body-fixed frame expression for the local magnetic disturbance, and \( \eta_b \) denotes the measurement noise. The noise \( \eta_b \) is usually low for magnetometer readings; however, the local magnetic disturbance \( B_m \) can be significant, especially if the sensor is placed near the power wires to the motors.

The accelerometers and magnetometers can be used to provide absolute attitude information on the vehicle while the rate gyroscope provides complementary angular velocity measurements. The attitude information in the magnetometer signal is straightforward to understand, in the absence of noise and bias, \( m_{\text{IMU}} \) provides a body-fixed frame measurement of \( R^A m \) and, consequently, constrains two DoF in the rotation \( R \).

The case for using the accelerometer signal for attitude estimation is far more subtle. Using the simplest model (6) with \( \Delta \equiv 0 \), \( a_{\text{IMU}} = R^T(\dot{v} - g\bar{z}) = (T_z/m)\tilde{z} \approx g\tilde{z} \). This shows that the measured acceleration, for this simple model, would always point in the body-fixed frame direction \( \tilde{z} \) and provides no attitude information. In practice, it is the blade-flapping component of the thrust that contributes attitude information to the accelerometer signal [7]. Recalling (10) and ignoring bias and noise terms, the model for \( a_{\text{IMU}} \) can be written as

\[
a_{\text{IMU}} = -\frac{T_z}{m} \tilde{z} - \frac{F}{m} R^T \dot{v}.
\]

As we show later in the section, only the low-frequency information from the accelerometer signal will be used in the observer construction. Thus, it is only the low-frequency or approximate steady-state response \( \dot{v} \) of the velocity \( v \) that we need to estimate to build a model for the low-frequency component of \( a_{\text{IMU}} \). Setting \( \dot{v} = 0 \) in (1b), substituting for \( \dot{v} \) (10), and rearranging, we obtain an estimate of the low-frequency component of the velocity signal

\[
DR^T \dot{v} \approx R^T \tilde{z} - \tilde{z}.
\]

Substituting \( DR^T \dot{v} \) for \( DR^T v \) in (16), we obtain

\[
a_{\text{IMU}} \approx -\frac{T_z}{m} R^T \tilde{z},
\]

where \( a_{\text{IMU}} \) denotes the low-frequency component of the accelerometer signal. That is, the low-frequency content of \( a_{\text{IMU}} \) when the vehicle is near hover is the body-fixed frame expression for the supporting force that is the negative gravitational vector expressed in the body-fixed frame. Most robotics applications involve a quadrotor spending significant periods of time in hover, or slow forward flight, with \( \dot{v} \approx 0 \), and using the accelerometer reading as an attitude reference during this flight regime has been shown to work well in practice.

The attitude kinematics of the quadrotor are given by (1c). Let \( \hat{R} \) denote an estimate for attitude \( R \) of the quadrotor vehicle. The following observer [12] fuses accelerometer, magnetometer, and gyroscope data as well as other direct attitude estimates \( \hat{R}_0 \) (such as provided by a VICON or other external measurement system) should they be available:

\[
\begin{aligned}
\dot{\hat{R}} &= \hat{R}(\Omega_{\text{IMU}} - \dot{\hat{b}}) \times \bar{z}, \\
\dot{\hat{b}} &= k_b \bar{z}, \\
\bar{z} &= \left( \frac{k_d}{\delta} ((\hat{R}^T \tilde{z}) \times \bar{a}_{\text{IMU}}) + \frac{k_m}{|A_m|^2} ((\hat{R}^T A m) \times m_{\text{IMU}}) \right) \times \\
&+ k_E \mathcal{P}_{\text{so}(3)}(\hat{R}^T E),
\end{aligned}
\]

where \( k_d, k_m, k_E, \) and \( k_b \) are arbitrary nonnegative observer gains and \( \mathcal{P}_{\text{so}(3)}(M) = (M - M^T)/2 \) is the Euclidean.
matrix projection onto skew-symmetric matrices. If any one of the measurements in the innovation $x$ are not available or unreliable, then the corresponding gain should be set to zero in the observer. Note that both the attitude estimate $\hat{R}$ and the bias corrected angular velocity $\hat{\Omega} = \hat{\Omega}_{\text{IMU}} - \hat{b}$ are estimated by this observer. The observer (18) has been extensively studied in the literature [12], [13] and shown to converge exponentially (both theoretically and experimentally) to the desired attitude estimate of attitude with $b$ converging to the gyroscope bias $\hat{b}$. The filter has a complementary nature, using the high-frequency part of the gyroscope signal and the low-frequency parts of the magnetometer, accelerometer, and external attitude measurements [12]. The roll-off frequencies associated with each of these signals is given by the gains $k_a$, $k_m$, and $k_e$ in rad.s$^{-1}$, and good performance of the observer depends on how these gains are tuned. In particular, the accelerometer gains must be tuned to a frequency below the normal bandwidth of the vehicle motion, less than 5 rad.s$^{-1}$ for a typical quadrotor. The magnetometer gain and external gain can be tuned for a higher roll-off frequency depending on the reliability of the signals. The bias gain $k_b$ is typically chosen an order of magnitude slower than the innovation gains $k_b < k_a/10$, leading to a rise time of the bias estimate as slow as 30 s or more. This dynamic response is necessary to track slowly varying bias and decouples the bias estimate from the attitude response; however, it is necessary to initialize the observer with a bias estimate at take off to avoid a long transient in the filter response.

A particular advantage of this observer formulation is that the gains can be adjusted in real time as long as care is taken that the bias gain is small. Adjusting the gains in real time allows one to use the accelerometer during a period when the vehicle is in hover and then set the gain $k_a = 0$ during acrobatic maneuvers when the low-frequency assumptions on $\hat{a}_{\text{IMU}}$ no longer hold. The nonlinear robustness, guaranteed asymptotic stability, and flexibility in gain tuning make this observer a preferred candidate for quadrotor attitude estimation compared with classical filters such as the extended Kalman filter (EKF), multiplicative EKF, or more sophisticated stochastic filters.

**Estimating Translational Velocity**

The blade-flapping response provides a way to build an observer for the horizontal velocity of the vehicle based on the IMU sensors [7], at least while the vehicle is flying in the horizontal plane. Assume that a good estimate of the vehicle attitude $\hat{R}$ is available and that the vehicle is flying at constant height.

Recalling the projector (11), the horizontal component of the inertial acceleration can be measured by

$$\Lambda a_h := P_h \Lambda a = P_h \hat{R} a \approx P_h \hat{R} a,$$  \hspace{1cm} (19)

where the signals $a$ and $\hat{R}$ are available. Since we assume that the vehicle is flying at a constant height, one has $v_z \approx 0$, and recalling (12), $P_h^T v_h \approx v$. Further, the thrust $T_z \approx m g$ must compensate the weight of the vehicle. Recalling (16) and taking the horizontal component, one has

$$\Lambda a_h \approx -g P_h \hat{R} \tilde{z} - g P_h \hat{R} \hat{D} \hat{R}^T P_h^T v_h.$$  \hspace{1cm} (20)

Assuming that the attitude filter estimate is good, i.e., $\hat{R} = R$, then (19) and (20) can be solved for an estimate of $v_h$

$$v_h \approx -\frac{1}{g} \left[ P_h \hat{R} \hat{D} \hat{R}^T P_h^T \right]^{-1} (\Lambda a_h + g P_h \hat{R} \tilde{z}).$$  \hspace{1cm} (21)

This estimate of $v_h$ will be well defined as long as the $2 \times 2$ matrix $P_h \hat{R} \hat{D} \hat{R}^T P_h^T$ is invertible, a condition that will hold as long as the vehicle pitches or rolls by less than 90° during flight.

Equation (21) provides a measurement of the horizontal velocity; however, since it directly incorporates the unfiltered accelerometer readings, it is generally too noisy to be of much use. Its low-frequency content can, however, be used to drive a velocity complementary observer that uses the attitude estimate and the system model (1b) along with the thrust model (10) for its high-frequency component. Let $\dot{v}_h$ be an estimate of the horizontal component of the inertial velocity of the vehicle. Recalling (1b), we propose the following observer

$$\dot{\dot{v}}_h = -g P_h^T \left( \hat{R} \tilde{z} + \hat{R} \hat{D} \hat{R}^T P_h^T v_h \right) - k_w (\dot{v}_h - v_h),$$  \hspace{1cm} (22)

where $v_h$ is given by (21). The gain $k_w > 0$ provides a tuning parameter that adjusts the roll-off frequency for the information from $\dot{v}_h$ that is used in the filter. It also uses an estimated velocity $\dot{v}_h$ to provide an approximation of the more correct $\hat{R} \hat{D} \hat{R}^T P_h^T v_h$ term in the feedforward velocity estimate; however, since the underlying dynamics associated with this term are stable, the observer is stable even with this approximation.

**Estimating Position**

The final part of state that must be estimated is position, which is typically considered separately as position in the plane and height. Considering the height first, there are in fact two separate heights that are of importance: the first is the absolute height of the vehicle and the second is the relative height over the terrain at a given time. Unfortunately, there is no effective way to use the IMU to estimate absolute height; at best, some low-frequency information from the $z$ axis of the accelerometer provides limited information about vertical motion. Most quadrotors include a barometric sensor that can resolve absolute height to a few centimeters. Absolute height can also be estimated using GPS, VICON, or a full SLAM system. Relative height can be estimated using acoustic, laser-ranging or infrared sensors.
sensors. Once a sufficiently accurate height measurement is available, it is better to use this directly in the control loop than add additional levels of complexity in designing a height observer, especially since, for a typical system, the only feedforward information available is the noisy accelerometer readings.

Position in the plane can also be determined in a relative or absolute way. Absolute position can be obtained from a GPS (few-centimeter accuracy at up to 10 Hz [6]) or an external localization device such as a VICON motion capture system (50 µm accuracy at 375 Hz). However, a GPS does not work indoors and motion-capture systems are expensive, and their sensor array has a limited spatial extent that is impractical to scale up for large indoor environments.

Relative position can be estimated by measuring the distance to objects in the environment from onboard sensors, typically small onboard laser range finders (LRFs) or RGBD camera systems such as the Kinect. Well-known SLAM techniques, borrowing LRF-based techniques similar to those developed for mobile ground robots over the last decade, have been applied to quadrotors [14]. However, LRFs provide only a cross section of the 3-D environment and this scan plane tilts as the vehicle maneuvers, resulting in apparent changes to the distance of walls, and, in extreme cases, the scan plane can intersect the floor or ceiling. LRFs are heavy and power hungry, which prevents their application to the next generation of much smaller quadcopters.

Vision has the advantage that the sensor is small, lightweight, and low power, which will become increasingly important as the size of aerial vehicles decreases. Vision can provide essential navigational competencies such as odometry, attitude estimation, mapping, place and object recognition, and collision detection. There is a long history of applying vision to aerial robotic systems [15]-[19] for indoor and outdoor environments, and the well-known Parrot AR.Drone game device makes strong use of vision for attitude and odometry [20]. Vision can also be used for object recognition based on color, texture, and shape, as well as collision avoidance.

Vision is not without its challenges. First, vision is computationally intense and can result in a low sample rate. Since onboard computational power is limited (by SWAP consumption), most reported systems transmit the images wirelessly to a ground station, which increases system complexity, control latency, and the susceptibility to interference and dropouts. However, processor speed continues to improve, and we can also utilize the vision and control techniques used by flying insects that perform complex tasks with very limited sensing and neural capability [21]. Second, there is an ambiguity between certain rotational and translational motions, particularly, when a narrow field of view perspective camera is used. Third, the underactuated quadrotor uses the roll and pitch DoF to point the thrust vector in the direction of the desired translational motion. For a camera that is rigidly attached to the quadrotor, this attitude control motion induces a large apparent motion in the image. It is therefore necessary to estimate vehicle attitude at the instant the image was captured by the sensor to eliminate this effect. Biological systems face similar problems, and interestingly, mammals and insects have developed similar solutions: gyroscopic sensors (the vestibular sensors of the inner ear and the halteres, respectively) [22]. Finally, there exists a problem with recovering motion scale when using a single camera. Stereo is possible, but the baseline is constrained, particularly as vehicles get smaller.

Control
The control problem, to track smooth trajectories \((\mathbf{R}'(t), \mathbf{v}'(t)) \in \text{SE}(3)\), is challenging for several reasons. First, the system is underactuated: there are four inputs \(\mathbf{u} = (T_{S}, \mathbf{v}')^{T}\), while \(\text{SE}(3)\) is six dimensional. Second, the aerodynamic model described above is only approximate. Finally, the inputs are themselves idealized. In practice, the motor controllers must overcome the drag moments to generate the required speeds and realize the input thrust \((T_{S})\) and moments \((\tau)\). The dynamics of the motors and their interactions with the drag forces on the propellers can be difficult to model, although first-order linear models are a useful approximation.

A hierarchical control approach is common for quadrotors. The lowest level, the highest bandwidth, is in control of the rotor rotational speed. The next level is in control of vehicle attitude, and the top level is in control of position along a trajectory. These levels form nested feedback loops, as shown in Figure 5.

Controlling the Motors
Rotor speed drives the dynamic model of the vehicle according to (8), so high-quality control of the motor speed is fundamentally important for overall control of the vehicle; high bandwidth control of the thrust \(T_{S}\), denoted by \(u_1\), and the torques \((\tau_x, \tau_y, \tau_z)\), denoted by \(u_2\), lead to high performance attitude and position control. Most quadrotor vehicles are equipped with brushless dc motors that use back electromotive force (EMF) sensing.
for rotor commutation and high-frequency pulsewidth modulation (PWM) to control motor voltage. The simplest systems generally use a direct voltage control of the motors since steady-state motor speed is proportional to voltage; however, the dynamic response is second-order due to the mechanical and electrical dynamics. Improved performance is obtained by incorporating single-input single-output control at the motor/rotor level

\[ V_i = k(\sigma_i^* - \sigma_i) + V_{ii}(\sigma_i^*), \]  

where \( V_i \) is the applied motor voltage, \( \sigma_i^* \) is the desired speed, and the actual motor speed \( \sigma_i \) can be measured from the electronic commutation in the embedded speed controller. This can help to overcome a common problem where the rotor speed for a given PWM command setting will decrease as the battery voltage reduces during flight. The significant load torque due to aerodynamic drag will lead to a tracking error that can be minimized by high proportional gain (\( k \)) and/or a feedforward term. A positive benefit of the drag torque is that the system is heavily damped, which precludes the need for derivative control. The feedforward term \( V_{ii}(\sigma_i^*) \) compensates for the steady-state PWM associated with a given velocity set point by incorporating the best available thrust model determined using static thrust tests and possibly including battery voltage.

The performance of the motor controllers is ultimately limited by the current that can be supplied from the batteries. This may be a significant limiting factor for smaller vehicles. Overly aggressive tuning and extreme maneuvers may cause the voltage bus to drop excessively, reducing the thrust from other rotors and, in extreme cases, causing the onboard electronics to brownout. For this reason, it is common to introduce a saturation, although this destroys the linearity of the motor/rotor response during aggressive maneuvers.

**Attitude Control**

We first consider the design of an exponentially converging controller in SO(3). Given a desired airframe attitude \( R^* \), we want to first develop a measure of the error in rotations. We choose the measure

\[ e_{R_x} = \frac{1}{2} ((R^*)^T R - R^T R^*), \]  

which yields a skew-symmetric matrix representing the axis of rotation required to go from \( R \) to \( R^* \) and whose magnitude is equal to the sine of the angle of rotation.

To derive linear controllers, we linearize the dynamics about the nominal hover position at which the roll (\( \phi \)) and pitch (\( \theta \)) are close to zero and the angular velocities are close to zero. If we write \( R = \hat{A}R_B \) as a product of the yaw rotation \( \hat{A}R_E(\psi) \) and \( \hat{A}R_B(\phi, \theta) \), which is a composition of the roll and pitch, we can linearize the rotation about \( (\psi, \phi, \theta) = (\psi_0, 0, 0) \)

\[ \hat{A}R_B = \hat{A}R_E(\psi_0 + \Delta \psi) \hat{A}R_B(\Delta \phi, \Delta \theta) \]

\[ \begin{pmatrix} \cos \psi & -\sin \psi & \Delta \theta \cos \psi + \Delta \phi \sin \psi \\ \sin \psi & \cos \psi & \Delta \theta \sin \psi - \Delta \phi \cos \psi \\ -\Delta \theta & \Delta \phi & 1 \end{pmatrix}, \]

where \( \psi = \psi_0 + \Delta \psi \). If \( R^* = \hat{A}R_B(\psi_0 + \Delta \psi, \Delta \phi, \Delta \theta) \) and \( R = \hat{A}R_B(\psi_0, 0, 0) \), (24) gives

\[ e_{R_x} = \begin{pmatrix} 0 & \Delta \psi & -\Delta \theta \\ -\Delta \psi & 0 & \Delta \phi \\ \Delta \theta & -\Delta \phi & 0 \end{pmatrix}, \]

which, as we expect, corresponds to the error vector

\[ e_R = (\Delta \phi, \Delta \theta, \Delta \psi)^T, \]

with components in the body-fixed frame. If the desired angular velocity vector is zero, we can compute the proportional and derivative error to obtain the PD control law

\[ u_2 = -k_\Omega e_R - k_\Omega e_\Omega, \]

where \( k_\Omega \) and \( k_\Omega \) are positive definite gain matrices. This controller guarantees stability for small deviations from the hover position.

To obtain convergence for larger deviations from the hover position, it is necessary to revert back to (24) without linearization. This allows us to directly compute the error on SO(3). By compensating for the nonlinear inertial terms and by including the correct error term, we obtain

\[ u_2 = J(-k_\Omega e_R - k_\Omega e_\Omega) + \Omega \times J(\Omega - J(\Omega \times R^T R^* R^T R^* \hat{\Omega}^*)), \]

(27)

This controller is guaranteed to be exponentially stable for almost any rotation [23]. From a practical standpoint, it is possible to neglect the last three terms in the controller and achieve satisfactory performance, but the correct calculation of the error term is important [24].

**Trajectory Control**

We now turn our attention to the control of the trajectory along a specified trajectory \( \zeta^*(t) \). As before, we first consider linear controllers by linearizing the dynamics about \( \zeta = \zeta^*(t), \phi = 0, \psi = \psi^*(t), \zeta = 0, \) and
\[ \dot{\phi} = \dot{\theta} = \dot{\psi} = 0, \text{ with the nominal input given by} \]
\[ u_1 = mg, u_2 = 0. \]
Linearizing (1a), we get
\[ \begin{align*}
\ddot{\xi}_1 &= g(\Delta \theta \cos \psi^* + \Delta \phi \sin \psi^*), \\
\ddot{\xi}_2 &= g(\Delta \phi \sin \psi^* - \Delta \phi \cos \psi^*), \\
\ddot{\xi}_3 &= \frac{1}{m} u_1 - g.
\end{align*} \tag{28} \]
To exponentially drive all three components of error, we want to command the acceleration vector \( \ddot{\xi} \) to satisfy
\[ (\ddot{\xi}^*(t) - \ddot{\xi}^{\text{com}}) + K_d(\ddot{\xi}^*(t) - \ddot{\xi}) + K_p(\ddot{\xi}^*(t) - \ddot{\xi}) = 0. \]
From (28), we can immediately write
\[ u_1 = m\left( g + \ddot{\xi}_3 + k_d z_3(\ddot{\xi}_3 - \ddot{\xi}_3) + k_p z_3(\ddot{\xi}_3 - \ddot{\xi}_3) \right), \tag{29} \]
to guarantee \((\ddot{\xi}_3(t) - \ddot{\xi}^{\text{com}}(t)) \to 0\). Similarly, for the other two components, we choose to command the appropriate \( \theta^* \) and \( \phi^* \) to guarantee exponential convergence
\[ \begin{align*}
\phi^* &= \frac{1}{g} \ddot{\xi}^{\text{com}} \sin \psi^* (t) - \ddot{\xi}^{\text{com}} \cos \psi^* (t), \\
\theta^* &= \frac{1}{g} \ddot{\xi}^{\text{com}} \cos \psi^* (t) + \ddot{\xi}^{\text{com}} \sin \psi^* (t),
\end{align*} \tag{30a,b} \]
where the above equations are obtained by replacing \( \Delta \theta \) by \( \theta^* \) and \( \Delta \phi \) by \( \phi^* \) in (28). Finally, \((\psi^*, \phi^*, \theta^*)\) are provided as set points to the attitude controller discussed in the previous section. Thus, as shown in Figure 5, the control problem is addressed by decoupling the position control and attitude control subproblems, and the position control loop provides the attitude set points for the attitude controller.

The position controller can also be obtained without linearization. This is done by projecting the position error (and its derivatives) along \( \mathbf{b}_3 \) and applying the input \( u_1 \) that cancels the gravitational force and provides the appropriate proportional plus derivative feedback
\[ u_1 = m\mathbf{b}_3^T \left( \ddot{\xi}_3 + K_d (\ddot{\xi}_3 - \ddot{\xi}) + K_p (\ddot{\xi}_3 - \ddot{\xi}) + g \mathbf{a}_3 \right). \tag{31} \]
Note that the projection operation is a nonlinear function of the roll and pitch angles, and, thus, this is a nonlinear controller. In [23], it is shown that the two nonlinear controllers (27) and (31) result in exponential stability and allow the robot to track trajectories in SE(3).

**Trajectory Planning**

The quadrotor is underactuated, and this makes it difficult to plan trajectories in 12-dimensional state space (6 DoF position and velocity). However, the problem is considerably simplified if we use the fact that the quadrotor dynamics are differentially flat [25]. To see this, we consider the output position \( \zeta \) and the yaw angle \( \psi \). We show that we can write all state variables and inputs as functions of the outputs (\( \zeta, \psi \)) and their derivatives. Derivatives of \( \zeta \) yield the velocity \( \nu \) and the acceleration, \( \ddot{\nu} \). From Figure 3 we see that
\[ \mathbf{e}_1 = [\cos \psi, \sin \psi, 0]^T, \]
and the unit vectors for the body-fixed frame can be written in terms of the variables \( \psi \) and \( \nu \) as
\[ \mathbf{b}_3 = \frac{\nu - g \mathbf{a}_3}{\| \nu - g \mathbf{a}_3 \|}, \quad \mathbf{b}_2 = \frac{\mathbf{b}_3 \times \mathbf{e}_1}{\| \mathbf{b}_3 \times \mathbf{e}_1 \|}, \quad \mathbf{b}_1 = \mathbf{b}_2 \times \mathbf{b}_3 \]
provided \( \mathbf{b}_3 \times \mathbf{e}_1 \neq 0 \). This defines the rotation matrix \( \mathbf{A}_{\mathbf{B}} \) as a function of \( \mathbf{v} \) (the second derivative of \( \zeta \)) and \( \psi \). In this way, we write the angular velocity and the four inputs as functions of position, velocity, acceleration, jerk (\( \gamma \)), and snap, or the derivative of jerk (\( \sigma \)). From these equations, it is possible to verify that there is a diffeomorphism between the \( 18 \times 1 \) vector
\[ \left( \zeta^T, \nu^T, \mathbf{a}^T, \gamma^T, \sigma^T, \psi^T, \dot{\psi}^T, \ddot{\psi}^T \right)^T \]
and
\[ R \times \left( \zeta^T, \nu^T, \Omega^T, u_1, \dot{u}_1, \ddot{u}_1, u_2^T \right)^T. \]
This property of differential flatness makes it easy to design trajectories that respect the dynamics of the underactuated system. Any four-times-differentiable trajectory in the space of flat outputs, \((\zeta^T(t), \nu(t))^T\), corresponds to a feasible trajectory—one that satisfies the equations of motion. All inequality constraints of states and inputs can be expressed as functions of the flat outputs and their derivatives. This mapping to the space of flat outputs can be used to generate trajectories that minimize a cost functional formed by a weighted combination of the different flat outputs and their derivatives:
\[ \min_{\zeta(t), \nu(t)} \int_0^T L(\zeta, \dot{\zeta}, \ddot{\zeta}, \nu, \psi, \dot{\psi}, \ddot{\psi}) dt, \]
\[ g(\zeta(t), \nu(t)) \leq 0. \tag{32} \]
In [24], minimum snap trajectories were generated by minimizing a cost functional derived from the snap and the angular yaw acceleration with
\[ L(\zeta, \dot{\zeta}, \ddot{\zeta}, \nu, \psi, \dot{\psi}, \ddot{\psi}) = (1 - \gamma)(\dot{\zeta})^4 + \gamma(\ddot{\psi})^2. \]
By suitable parameterizing trajectories with basis functions in the flat space and by considering linear inequalities in the flat space.
space to model constraints on states and inputs (e.g., $u_1 \geq 0$), it is possible to turn this optimization into a quadratic program that can be solved in real time for planning.

Finally, as shown in [11], it is possible to combine this controller with attitude-only controllers to fly through vertical windows or land on inclined perches with close to zero normal velocity. A trajectory controller is used by the robot to build up momentum, while the attitude controller enables reorientation while coasting with the generated momentum.

**Vision-Based Perception and Control**

There are two approaches to the question of controlling an aerial vehicle based on visual information. The first is to use classical robotic SLAM techniques, although with the caveat that the environment and state estimation are inherently 3-D. There are many researchers currently working on this problem, and we will not attempt to discuss this approach further, except to say that should a good-quality environmental estimation and localization algorithm be developed, the control techniques discussed above can be applied. The second approach is direct sensor-based control [26], the most commonly referred to case, being that of image-based visual servo control [27]–[29].

The motion of a point in an image is a function of its coordinate $(u, v)$ and the camera motion

$$
\begin{pmatrix}
\dot{u} \\
\dot{v}
\end{pmatrix} = J(u, v, Z)v,
$$

where $Z$ is the point depth, $v = (v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)^\top$ is the spatial velocity of the camera (and vehicle), and $J(\cdot)$ is the visual Jacobian or interaction matrix. $J$ can be formulated for a perspective camera [30], where $(u, v)$ are pixel coordinates; or a spherical camera [31] where $(u, v)$ are latitude and longitude angles.

The pitch and roll motion of the vehicle are controlled by the attitude subsystem to maintain a position or to follow a path in space, and this causes image motion. We partition the equations as

$$
\begin{pmatrix}
\dot{u} \\
\dot{v}
\end{pmatrix} = J_1(u, v)(v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)^\top + J_2(u, v)\begin{pmatrix}
\omega_x \\
\omega_y
\end{pmatrix},
$$

where the right-most term describes the image motion due to the exogenous roll and pitch motion. Rearranging we can write

$$
\begin{pmatrix}
\dot{u}' \\
\dot{v}'
\end{pmatrix} = \left(\begin{pmatrix}
\dot{u} \\
\dot{v}
\end{pmatrix} - J_2(u, v)\begin{pmatrix}
\omega_x \\
\omega_y
\end{pmatrix}\right) = J_1(u, v)(v_x, v_y, v_z, \omega_x, \omega_y)^\top,
$$

where $(u', v')$ represent image points for which the roll and pitch motion has been removed based on the knowledge of $\omega_x$ and $\omega_y$, which can be obtained from gyroscopes.

Now consider a point in the image $(u'_i, v'_i)$ and its desired location in the image $(u'_*, v'_*)$. This desired position might come from a snapshot of the scene taken when the vehicle was at the desired pose that we wish to return to. The desired image motion is therefore $(u'_*, v'_*) = \lambda(u'_i \oplus u'_i, v'_i \oplus v'_i)$, where the operator $\oplus$ represents the difference on image plane or sphere. For $N$ points, we can write

$$
\begin{pmatrix}
\dot{u}'_1 \\
\dot{v}'_1 \\
\vdots \\
\dot{u}'_N \\
\dot{v}'_N
\end{pmatrix} - \begin{pmatrix}
J_1(u_1, v_1) \\
\vdots \\
J_1(u_N, v_N)
\end{pmatrix}\begin{pmatrix}
\omega_x \\
\omega_y
\end{pmatrix} = \begin{pmatrix}
J_2(u_1, v_1) \\
\vdots \\
J_2(u_N, v_N)
\end{pmatrix}\begin{pmatrix}
v_x \\
v_y \\
v_z
\end{pmatrix},
$$

If $N > 2$ and the matrix $B$ is nonsingular, we can solve for the required translational and yaw velocity to move the vehicle to a pose where the feature points have the desired image coordinates $(u'_i, v'_i)$. The desired velocity is input to a control system as discussed earlier. This is an example of image-based visual servoing for an underactuated vehicle, and the technique can be applied to a wider variety of problems, such as holding station, path following, obstacle avoidance, and landing.

**Conclusions**

In this article, we have provided a tutorial introduction to modeling, estimation, and control for multirotor aerial vehicles, with a particular focus on the most common form—the quadrotor. The dynamic model includes the rigid body motion of the vehicle in SE(3), the simple aerodynamics associated with hover, and the extension to the case of forward motion where blade flapping becomes important. State estimation based on accelerometers, gyroscopes, and magnetometers was discussed for attitude and translational velocity, and GPS, motion-capture systems, and cameras for position estimation. A hierarchy of control techniques was discussed, from the individual rotors through attitude control, aggressive trajectory following, and image-based visual control. The future possibilities of highly agile small-scale vehicles were laid with a discussion on dimensional scaling for which vision will be an important sensing modality.

**Acknowledgment**

This research was partly supported by the Australian Research Council through Future Fellowship FT0991771,
Foundations of Vision Based Control of Robotic Vehicles, the U.S. Army Research Laboratory Grant W911NF-08-2-0004, and the U.S. Office of Naval Research Grants N00014-07-1-0829, N00014-09-1-1051, and N00014-08-1-0696.

References


Robert Mahony, Research School of Engineering, Australian National University, Canberra 0200, Australia. E-mail: Robert.Mahony@anu.edu.au.

Vijay Kumar, Department of Mechanical Engineering and Applied Mechanics, GRASP Laboratory, University of Pennsylvania, Philadelphia, USA. E-mail: vijay.kumar@grasp.upenn.edu.

Peter Corke, School of Electrical Engineering and Computer Science, Queensland University of Technology, Australia. E-mail: Peter.Corke@qut.edu.au.