Embedded UAS Autopilot and Sensor Systems

Randal W. Beard
Department of Electrical and Computer Engineering
Brigham Young University
beard@byu.edu

February 17, 2010

Abstract

This objective of this article is to give an overview of autopilot software design for small and miniature unmanned air systems. An architecture that is typical of commercially available autopilots is described. A detailed explanation of the inner-loop control structure is described, including the lateral directional feedback loops and the longitudinal directional feedback loops. We give a brief description of the sensor package that is usually on small UAS, and describe simple filtering schemes for extracting estimates of the states needed by the autopilot. Finally, we briefly describe GPS navigation for following straight lines and orbits.

Keywords: Autopilot design, small unmanned air vehicles, sensors on small UAVs, Kalman filtering, GPS guidance.

1 Introduction

Unmanned Air Systems (UAS) are playing an increasingly prominent role in military operations. Technology advancements have enabled the development of large UAS like the Northrop Grumman Global Hawk and the General Atomics Predator, and also smaller UAS like the AeroVironment Raven and the InSitu ScanEagle. As recent conflicts have demonstrated, there are numerous military applications for small UAS [1] including reconnaissance, surveillance, battle damage assessment, and communications relays. Civil and commercial applications are not as well developed, although potential applications are extremely broad in scope. Possible applications for UAS technology include environmental monitoring, forest fire monitoring, homeland security, border patrol, drug interdiction, aerial surveillance and mapping, traffic monitoring, precision agriculture, disaster relief, ad-hoc communications networks, and rural search and rescue.

The design of the autopilot system for a small UAS differs from manned aircraft in several ways. The first obvious difference is that a human pilot is not in the loop. This

[1] In this article, we will use the term *small UAS* to refer to fixed wing aircraft with wing span between 1-10 feet, and payloads that may range from 0.25-10 kg. Operation times may range from 15 minutes to 10 hours.

has several important implications. In particular, the pilot is not able to set trim conditions which must therefore be determined automatically by the autopilot. Additional functions that are typically performed by the autopilot include take-off and landing and waypoint navigation. The second difference between small UAS and larger aircraft is the type of sensors that are available. Large aircraft typically have high quality inertial measurement unit (IMU) sensors, angle-of-attack sensors, and GPS. For a small UAS, the sensor suite is limited to global position system (GPS), microelectromechanical systems (MEMS) gyros and accelerometers, and differential and absolute pressure sensors. The implication is that estimates of the states of the aircraft will be of much poorer quality with small UAS. Another difference is that the airspeed and the wind speed are more closely aligned for small UAS than they are for larger aircraft. The implication is that navigation algorithms must be explicitly designed for high wind situations. Finally, small UAS have much less payload capacity than larger aircraft and can therefore not carry significant computational resources on-board. The consequence is that the autopilot algorithms must have relatively modest computational and memory requirements.

There are several commercially available autopilots for small UAS. These include the Kestrel Autopilot produced by Procerus Technologies [1], the Piccolo Autopilot produced by Cloud Cap Technologies [2], and the MP2128 Autopilot produced by MicroPilot Inc [3]. These autopilots are shown in Figure 1 and some of their specifications are shown in Table 1.

(a) Kestrel (b) Piccolo II (c) MP2128

Figure 1: (a) The Kestrel autopilot by Procerus Technologies. (b) The Piccolo II. Reproduced by permission of Cloud Cap Technology. (c) The MP2128 by MicroPilot.

	Kestrel	Piccolo II (includes datalink)	MP2128
size (inches)	2 x 1.37 x 0.47	5.6 x 1.8 x 0.24	1.5 x 4 x 0.5
weight (grams)	17	226	28
power (Watts)	2	4	1

Table 1: Size, weight, and power specifications for the Kestrel, Piccolo II, and MP2128 autopilots.

The objective of this article is to describe autopilot technologies for small UAS.

The principles will be common to all autopilots in this size and weight class. In Section 2 we will describe a general autopilot architecture. In Section 3 we will describe low-level control design using successive loop closure. Section 4 describes sensors that are typically on-board small UAS and sensor processing necessary to extract state information, and Section 5 discusses GPS navigation.

2 Autopilot Architecture

A typical system architecture for a small UAS is shown in Figure 2. The UAS is composed of a ground component and an air component. The air system includes the autopilot and a camera, which is often gimbaled. The autopilot communicates to the ground component via two communication links: one for telemetry and command and control, and the other for video. Typically, a 900 MHz transmitter is used for the command and control link, and a 2.4 GHz transmitter is used for the video link. The autopilot software consists of a state estimator, servo loops, and a path following module.

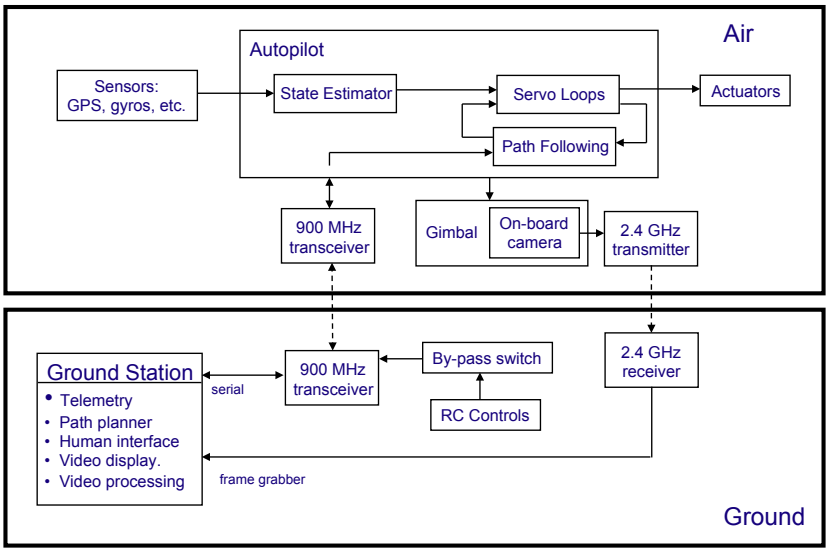

Figure 2: Autopilot architecture for small UAS. The architecture includes a ground component and an air component.

The ground system includes a ground station, the corresponding communication hardware, and a mechanism to allow radio control (RC) override. The RC override is for safety and regulatory requirements. The ground station typically includes telemetry recording, video display, a human interface, and possibly a waypoint path planning module.

This article focuses primarily on the autopilot software. In Section 3 we discuss the servo loops. The input to the servo loops is the estimated state $\hat{x}$ as well as the

commanded altitude h^c, airspeed V^c, pitch angle θ^c, and heading angle ψ^c. The servo loops send commands to the actuators which include a rudder δ_r, aileron δ_a, elevator δ_e, and throttle δ_t. The path following module receives commands from the ground station to go to a specified waypoint, or to orbit a specific point on the ground at a certain radius and airspeed. Section 5 discusses the path following module. The autopilot and the path following blocks rely on accurate state estimates which are obtained by filtering the on-board sensors which include accelerometers, rate gyros, pressure sensors, and GPS. The state estimator is discussed in Section 4.

3 Inner-loop Control Structure

The equations of motion for fixed-wing aircraft are typically decomposed into lateral and longitudinal dynamics [4, 5, 6, 7]. The lateral dynamics include the roll angle, the inertial heading direction (measured from North), and the inertial North-East position of the aircraft. The longitudinal dynamics include the pitch angle, the airspeed, and the altitude of the aircraft. Feedback loops are designed separately for the lateral and longitudinal dynamics. While there is coupling between lateral and longitudinal motion, the feedback loops are designed to reject the disturbances that are due to this coupling. Design of the feedback control loops for lateral motion is discussed in Section 3.1 and for longitudinal motion in Section 3.2. The autopilot design discussed in this article would not be appropriate for highly aggressive maneuvers where the coupling between lateral and longitudinal motion must be directly addressed.

Figure 3 depicts the definitions of the variables used throughout the rest of the article. The body frame x-axis is out the nose of the aircraft, the y-axis is out the right wing, and the z-axis is out the bottom. The rotation rates about the body frame axes are denoted by p, q, and r, and are called the roll rate, pitch rate, and yaw rate. The yaw angle ψ is defined by aligning the body axis with the inertial North-East-Down (NED) axes and rotating about the body z-axis (Down axis) by ψ. The pitch angle is defined by rotating the resulting coordinate system about the body y-axis by θ. The roll angle is defined by rotating the yawed and pitched coordinate system about the body x-axis by ϕ. The North-East-Down position of the aircraft is denoted by (p_n, p_e, p_d). The altitude is measured along the negative Down axis and is denoted by $h = -p_d$. The airspeed of the airframe, which is defined as the speed of the aircraft relative to the surrounding air mass, is V. The flight path angle, which is the angle that the velocity vector makes with the North-East plane is given by γ and is equal to the pitch angle minus the aircraft angle of attack: $\gamma = \theta - \alpha$. The course angle χ is the angle of the velocity vector from North and is equal to the yaw angle plus the crab angle: $\chi = \psi + \chi_c$, where the crab angle is the angle between the inertial velocity vector of the aircraft, and the body frame x-axis which points out the noise of the aircraft. Note that the crab angle is different than the sideslip angle, which is the difference between the relative wind vector and the body frame x-axis. The lateral dynamics involve the variables p, ϕ, r, ψ, χ, p_n, and p_e. The longitudinal dynamics involve the variables q, θ, χ, V, and h.

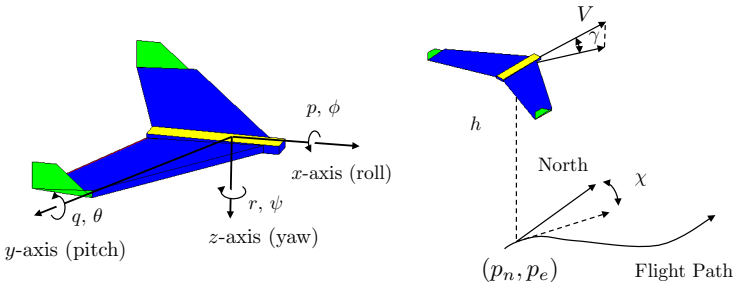

Figure 3: Definition of variables used throughout this article. The angular rates are given by p, q, and r. The Euler angles are denoted as ϕ, θ, ψ. The airspeed is V, the heading angle is χ and the flight path angle is γ. The inertial position is (p_n, p_e), and the altitude is h.

3.1 Lateral Autopilot

Figure 4 shows the block diagram for a lateral-directional autopilot using successive loop closure. There are five gains associated with the lateral-directional autopilot. The derivative gain k_{d_ϕ} provides roll rate damping at the innermost loop. The roll attitude is regulated with the proportional and integral gains k_{p_ϕ} and k_{i_ϕ}. The heading is regulated with the proportional and integral gains k_{p_χ} and k_{i_χ}. The idea with successive loop closure, is that the gains are successively chosen one at a time beginning with the inner loop and working outward. In particular, k_{d_ϕ} is usually selected first, k_{p_ϕ} second, k_{i_ϕ} third, k_{p_χ} fourth, and k_{i_χ} last.

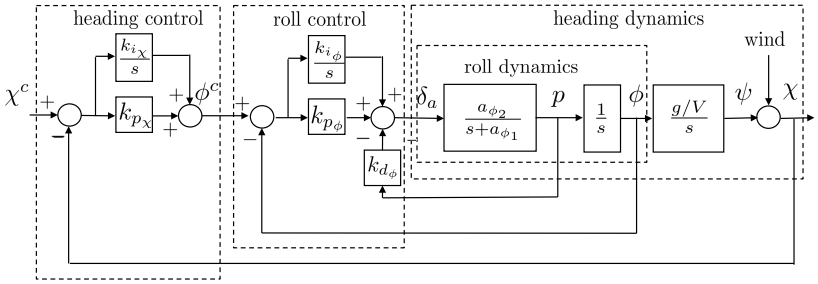

Figure 4: Control loops for the lateral autopilot using successive loop closure.

If a model of the dynamics is available, then the gains can be selected analytically. From Figure 4, note that if the integrator gain $k_{i_\phi} = 0$, then the transfer function from the commanded roll angle to the roll angle is given by

$$H_{\phi/\phi^c}(s) = \frac{k_{p_\phi} a_{\phi_2}}{s^2 + (a_{\phi_1} + a_{\phi_2} k_{d_\phi})s + k_{p_\phi} a_{\phi_2}}. \tag{1}$$

If the desired response is given by the canonical second-order transfer function

$$\frac{\omega_{n_\phi}^2}{s^2 + 2\zeta_\phi \omega_{n_\phi} s + \omega_{n_\phi}^2}, \tag{2}$$

then equating denominator polynomial coefficients, we obtain

$$\omega_{n_\phi}^2 = k_{p_\phi} a_{\phi_2} \tag{3}$$

$$2\zeta_\phi \omega_{n_\phi} = a_{\phi_1} + a_{\phi_2} k_{d_\phi}. \tag{4}$$

Solving these expressions for k_{p_ϕ} and k_{d_ϕ} gives

$$k_{p_\phi} = \frac{\omega_{n_\phi}^2}{a_{\phi_2}} \tag{5}$$

$$k_{d_\phi} = \frac{2\zeta_\phi \omega_{n_\phi} - a_{\phi_1}}{a_{\phi_2}}. \tag{6}$$

Therefore, selecting the desired damping ratio and natural frequency fix the values for k_{p_ϕ} and k_{d_ϕ}.

Since the roll dynamics is a type one system, zero steady-state tracking error in roll should be achievable without an integrator [8]. However, coupling from the longitudinal dynamics introduces a disturbance that enters at the summing junction before δ_a in Figure 4. Disturbances are also introduced into the roll dynamics by gusts or turbulence. To reject these disturbances, an integrator must be included. If a_{ϕ_1} and a_{ϕ_2} are known, then k_{i_ϕ} can be effectively selected using root locus techniques. The closed loop poles of the system are given by

$$s^3 + (a_{\phi_1} + a_{\phi_2} k_{d_\phi}) s^2 + a_{\phi_2} k_{d_\phi} s + a_{\phi_2} k_{i_\phi} = 0, \tag{7}$$

which can be placed in Evans form as

$$1 + k_{i_\phi} \left(\frac{a_{\phi_2}}{s\left(s^2 + (a_{\phi_1} + a_{\phi_2} k_{d_\phi}) s + a_{\phi_2} k_{d_\phi}\right)} \right) = 0. \tag{8}$$

Figure 5 shows the root locus of the characteristic equation plotted as a function of k_{i_ϕ}. For small values of gain, the system remains stable.

The next step in the successive-loop-closure design of the lateral-directional autopilot is to design the heading hold outer loop. If the inner loop from ϕ^c to ϕ has been adequately tuned, then $H_{\phi/\phi^c} \approx 1$ over the range of frequencies from 0 to ω_{n_ϕ}. Under this condition, the block diagram of Figure 4 can be simplified to the block diagram in Figure 6 for the purposes of designing the outer loop, where the wind disturbance has been transformed to the input of the plant.

The objective of the heading hold design is to select k_{p_χ} and k_{i_χ} in Figure 6 so that the heading angle χ asymptotically tracks steps in the commanded heading angle χ^c. From the simplified block diagram, the transfer functions from the inputs χ^c and the input disturbance d_χ to the output χ are given by

$$\chi = \frac{(g/V)s}{s^2 + k_{p_\chi}(g/V)s + k_{i_\chi}(g/V)} d_\chi + \frac{k_{p_\chi}(g/V)s + k_{i_\chi}(g/V)}{s^2 + k_{p_\chi}(g/V)s + k_{i_\chi}(g/V)} \chi^c. \tag{9}$$

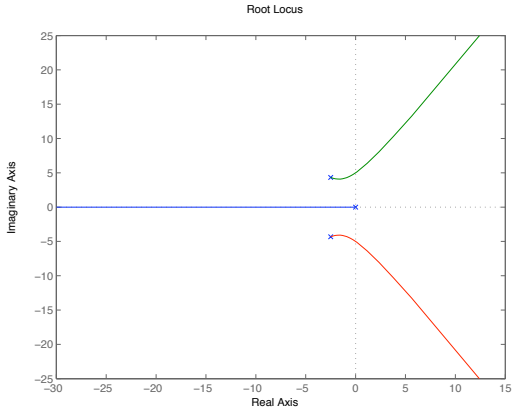

Figure 5: Roll loop root locus as a function of the integral gain k_{i_ϕ}.

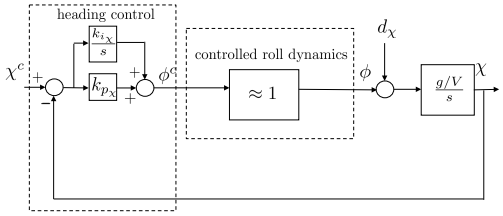

Figure 6: Heading hold outer feedback loop.

Note that if d_χ and χ^c are constants, then the final value theorem implies that $\chi \to \chi^c$. The transfer function from χ^c to χ has the form

$$H_\chi = \frac{2\zeta_\chi \omega_{n_\chi} s + \omega_{n_\chi}^2}{s^2 + 2\zeta_\chi \omega_{n_\chi} s + \omega_{n_\chi}^2}. \tag{10}$$

As with the inner feedback loops, we can choose the natural frequency and damping of the outer loop and from those values calculate the feedback gains k_{p_χ} and k_{i_χ}. Figure 7 shows the frequency response and the step response for H_χ. Note that because of the zero the standard intuition for the selection of ζ does not hold for this transfer function. Larger ζ results in larger bandwidth and smaller overshoot.

Comparing coefficients in Equations (9) and (10), we find

$$\omega_{n_\chi}^2 = (g/V) k_{i_\chi} \tag{11}$$

$$2\zeta_\chi \omega_{n_\chi} = (g/V) k_{p_\chi}. \tag{12}$$

Solving these expressions for k_{p_χ} and k_{i_χ} gives

$$k_{p_\chi} = 2\zeta_\chi \omega_{n_\chi} V/g \tag{13}$$

$$k_{i_\chi} = \omega_{n_\chi}^2 V/g. \tag{14}$$

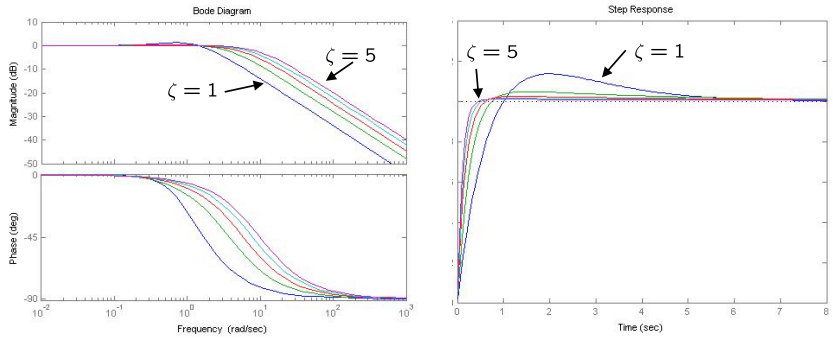

Figure 7: Frequency and step response for second order system.

To ensure proper function of this successive-loop-closure design, it is essential that the inner (roll) and outer (heading) feedback loops are sufficiently separated. Adequate separation can be achieved using the rule of thumb

$$\omega_{n_\phi} > 5\omega_{n_\chi}. \tag{15}$$

Generally, more bandwidth separation is better. More bandwidth separation requires either slower response in the χ loop (lower ω_{n_χ}), or faster response in the ϕ loop (higher ω_{n_ϕ}). Faster response usually comes at the cost of requiring more actuator control authority, which may not be possible given the physical constraints of the actuators.

The rudder is typically used to regulate the side-slip angle β to zero in order to maintain coordinated turn conditions. Because of space limitations we will not give the equations for rudder control.

3.2 Longitudinal Autopilot

The longitudinal autopilot is more complicated than the lateral autopilot because airspeed plays a significant role in the longitudinal dynamics. Our objective in designing the longitudinal autopilot will be to regulate airspeed and altitude using the throttle and the elevator as actuators. The method used to regulate altitude and airspeed depends on the altitude error. The flight regimes are shown in Figure 8.

In the take-off zone, full throttle is commanded and the pitch attitude is regulated to a fixed pitch angle θ^c using the elevator.

In the climb zone, the objective is to maximize the climb rate given the current atmospheric conditions. To maximize the climb rate, full throttle is commanded and the airspeed is regulated using the pitch angle. If the airspeed increases above its nominal value, then the aircraft is caused to pitch up which results in an increased climb rate and a decrease in airspeed. Similarly, if the airspeed drops below the nominal value, the airframe is pitched down thereby increasing the airspeed but also decreasing the climb rate. Regulating the airspeed using pitch attitude effectively keeps the airframe away from stall conditions. Note however, that we would not want to regulate airspeed with

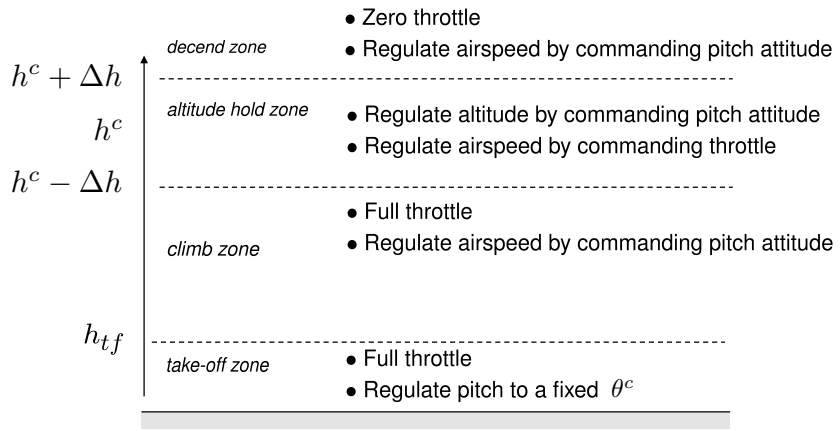

Figure 8: Flight regimes for the longitudinal autopilot

pitch attitude immediately after take-off because after take-off the airframe is always trying to gain airspeed but pitching down will drive the aircraft into the ground.

The descend zone is similar to the climb zone except that the throttle is commanded to zero. Again, stall conditions are avoided by regulating airspeed using the pitch angle thus maximizing the descent rate at a given airspeed. On some airframes zero throttle may not be possible or desirable. In these cases, the throttle is set to a minimum value.

In the altitude hold zone, the airspeed is regulated by adjusting the throttle, and the altitude is regulated by commanding the pitch attitude.

To implement the longitudinal autopilot shown in Figure 8 we need the following feedback loops: (1) pitch attitude hold using elevator, (2) airspeed hold using throttle, (3) airspeed hold using pitch attitude, and (4) altitude hold using pitch attitude. The design of each of these loops will be discussed in the next four subsections. Finally, the complete longitudinal autopilot will be presented in Section 3.2.5.

3.2.1 Pitch Attitude Hold

The pitch attitude hold loop is similar to the roll attitude hold loop and we will follow a similar line of reasoning in its development. From Figure 9, the transfer function from θ^c to θ is given by

$$H_{\theta/\theta^c}(s) = \frac{k_{p_\theta} a_{\theta_3}}{s^2 + (a_{\theta_1} + k_{d_\theta} a_{\theta_3})s + (a_{\theta_2} + k_{p_\theta} a_{\theta_3})}. \tag{16}$$

Note that in this case, the DC gain is not equal to one.

If the desired response is given by the canonical second order transfer function

$$\frac{K_{\theta_{\text{DC}}} \omega_{n_\theta}^2}{s^2 + 2\zeta_\theta \omega_{n_\theta} s + \omega_{n_\theta}^2}, \tag{17}$$

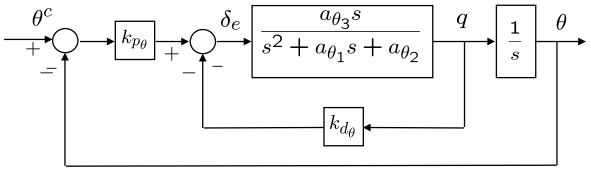

Figure 9: Pitch attitude hold feedback loop.

then equating denominator coefficients we obtain

$$\omega_{n_\theta}^2 = a_{\theta_2} + k_{p_\theta} a_{\theta_3} \tag{18}$$

$$2\zeta_\theta \omega_{n_\theta} = a_{\theta_1} + k_{d_\theta} a_{\theta_3}. \tag{19}$$

Solving these expressions for k_{p_θ} and k_{d_θ} gives

$$k_{p_\theta} = \frac{\omega_{n_\theta}^2 - a_{\theta_2}}{a_{\theta_3}} \tag{20}$$

$$k_{d_\theta} = \frac{2\zeta_\theta \omega_{n_\theta} - a_{\theta_1}}{a_{\theta_3}}. \tag{21}$$

Therefore, selecting the desired damping ratio and natural frequency fixes the value for k_{p_θ} and k_{d_θ}.

The DC gain of this inner-loop transfer function approaches one as the $k_{p_\theta} \to \infty$. The DC gain is given by

$$K_{\theta_{\text{DC}}} = \frac{k_{p_\theta} a_{\theta_3}}{(a_{\theta_2} + k_{p_\theta} a_{\theta_3})}, \tag{22}$$

which for typical gain values is significantly less than one. For the design of the outer loop, we will use this DC gain to represent the gain of the inner loop over its full bandwidth. An integral feedback term could be employed to ensure unity DC gain on the inner loop. The addition of an integral term, however, can severely limit the bandwidth of the inner loop.

3.2.2 Altitude Hold Using Commanded Pitch

The altitude hold autopilot again utilizes a successive loop closure strategy with the pitch attitude hold autopilot as an inner loop. Assuming that the pitch loops function as designed and $\theta \approx K_{\theta_{\text{DC}}} \theta^c$, the altitude hold loop using the commanded pitch is shown in Figure 10.

In the Laplace domain we have

$$h(s) = \left(\frac{K_{\theta_{\text{DC}}} V k_{p_h} \left(s + \frac{k_{i_h}}{k_{p_h}} \right)}{s^2 + K_{\theta_{\text{DC}}} V k_{p_h} s + K_{\theta_{\text{DC}}} V k_{i_h}} \right) h^c(s) + \left(\frac{s}{s^2 + K_{\theta_{\text{DC}}} V k_{p_h} s + K_{\theta_{\text{DC}}} V k_{i_h}} \right) d_h(s), \tag{23}$$

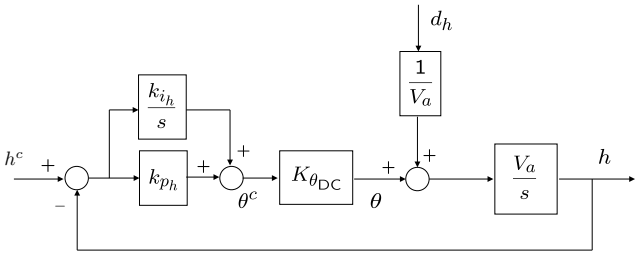

Figure 10: The altitude hold loop using the commanded pitch angle.

where again we see that the DC gain is equal to one, and input disturbances d_h are rejected for low frequencies. The closed loop transfer function is again independent of airframe parameters and is only dependent on the known airspeed. The gains k_{p_h} and k_{i_h} should be chosen such that the bandwidth of the altitude-using-pitch loop is approximately five times less than the bandwidth of the pitch attitude hold loop.

If the desired response of the altitude-hold loop is given by the canonical second order transfer function

$$\frac{\omega_{n_h}^2}{s^2 + 2\zeta_h \omega_{n_h} s + \omega_{n_h}^2}, \tag{24}$$

then equating denominator coefficients and solving for k_{i_h} and k_{p_h} we obtain

$$k_{i_h} = \frac{\omega_{n_h}^2}{K_{\theta_{\text{DC}}} V} \tag{25}$$

$$k_{p_h} = \frac{2\zeta_h \omega_{n_h}}{K_{\theta_{\text{DC}}} V}. \tag{26}$$

3.2.3 Airspeed Hold Using Commanded Pitch

The airspeed can also be regulated using the pitch angle as a control variable since in level flight pushing the nose down increases airspeed and pulling the nose up decreases airspeed. The block diagram for airspeed hold using pitch attitude hold as an inner loop is shown in Figure 11. Disturbance rejection requires a proportional-integral (PI) controller.

In the Laplace domain we have

$$\Delta V(s) = \left(\frac{(-K_{\theta_{\text{DC}}} g k_{p_{V_2}})(s + \frac{k_{i_{V_2}}}{k_{p_{V_2}}})}{s^+ (a_{V_1} - K_{\theta_{\text{DC}}} g k_{p_{V_2}}) s - K_{\theta_{\text{DC}}} g k_{i_{V_2}}} \right) \Delta V^c(s) + \left(\frac{s}{s^2 + (a_{V_1} - K_{\theta_{\text{DC}}} g k_{p_{V_2}}) s - K_{\theta_{\text{DC}}} g k_{i_{V_2}}} \right) d_V(s). \tag{27}$$

Note that the DC gain is equal to one and that step disturbances are rejected. The angle of attack α enters into the block diagram at the same location as d_V. If we assume that

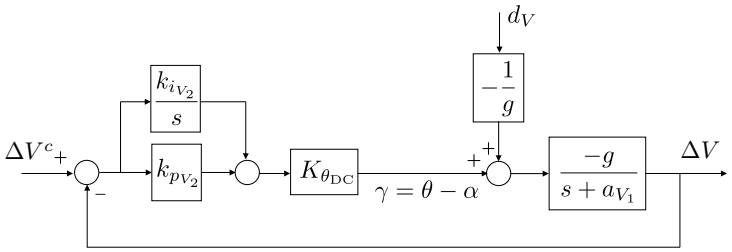

Figure 11: PI Controller to regulate airspeed using the pitch angle.

the angle of attack is constant, then the PI controller winds up the integrator to reject α. Therefore, the output of the PI controller can be used as the input to the pitch attitude hold loop.

The gains $k_{p_{V_2}}$ and $k_{i_{V_2}}$ should be chosen such that the bandwidth of the airspeed to pitch loop is less than the bandwidth of the pitch attitude hold loop. Following a procedure similar to previous sections, we can determine values for the feedback gains by matching denominator coefficients in Equation (27) with those of a canonical second-order transfer function. Denoting the desired natural frequency and damping ratio we seek to achieve with feedback as $\omega_{n_{V_2}}^2$ and ζ_{V_2}, matching coefficients and solving for the control gains gives

$$k_{i_{V_2}} = -\frac{\omega_{n_{V_2}}^2}{K_{\theta_{\mathrm{DC}}} g} \tag{28}$$

$$k_{p_{V_2}} = \frac{a_{V_1} - 2\zeta_{V_2}\omega_{n_{V_2}}}{K_{\theta_{\mathrm{DC}}} g}. \tag{29}$$

3.2.4 Airspeed Hold Using Throttle

The closed loop system for the airspeed using the throttle as an input is shown in Figure 12. The transfer function is given by

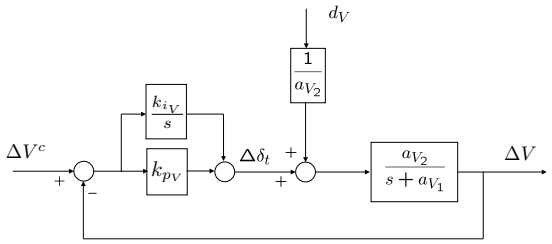

Figure 12: Airspeed hold using throttle.

$$\Delta V = \left(\frac{a_{V_2}(k_{p_V} s + k_{i_V})}{s^2 + (a_{V_1} + a_{V_2} k_{p_V})s + a_{V_2} k_{i_V}} \right) \Delta V^c + \left(\frac{1}{s^2 + (a_{V_1} + a_{V_2} k_{p_V})s + a_{V_2} k_{i_V}} \right) d_V \quad (30)$$

where ΔV is the deviation of the airspeed from trim, ΔV^c is the commanded deviation from trim, where it is clear that the DC gain is one and that input disturbances d_V are rejected for low frequencies. If a_{V_1} and a_{V_2} are known, then the gains k_{p_V} and k_{i_V} can be effectively determined using the same technique we have used previously. Equating the closed-loop transfer function denominator coefficients with those of a canonical second-order transfer function, gives

$$\omega_{n_V}^2 = a_{V_2} k_{i_V} \quad (31)$$

$$2\zeta_V \omega_{n_V} = a_{V_1} + a_{V_2} k_{p_V}. \quad (32)$$

Inverting these expressions gives the control gains

$$k_{i_V} = \frac{\omega_{n_V}^2}{a_{V_2}} \quad (33)$$

$$k_{p_V} = \frac{2\zeta_V \omega_{n_V} - a_{V_1}}{a_{V_2}}. \quad (34)$$

Note that since $\Delta V \triangleq V - \hat{V}$ and $\Delta V^c \triangleq V^c - \hat{V}$, then

$$\Delta V^c - \Delta V = V^c - V. \quad (35)$$

Therefore, the control loop shown in Figure 12 can be implemented without knowledge of the trim velocity $\hat{V}$. Similarly, since the integrator will wind up to reject step disturbances, and a constant error in $\hat{\delta}_t$ can be thought of as a step disturbance, we can set $\delta_t = \Delta\delta_t$.

3.2.5 Altitude Control State Machine

The longitudinal autopilot modes can be combined to create the altitude control state machine shown in Figure 13. In the climb zone, the throttle is set to its maximum value ($\delta_t = 1$) and the airspeed hold from commanded pitch mode is used to control the airspeed and thus ensure that the airframe avoids stall conditions. In simple terms, this causes the UAS to climb at its maximum possible climb rate until it is close to the altitude set point. Similarly, in the descend zone, the throttle is set to its minimum value ($\delta_t = 0$) and the airspeed hold from commanded pitch mode is again used to control airspeed. In this way, the UAS descends at a steady descent rate until it reaches the altitude hold zone. In the altitude hold zone, the airspeed from throttle mode is used to regulate the airspeed around V^c, and the altitude from pitch mode is used to regulate the altitude around h^c.

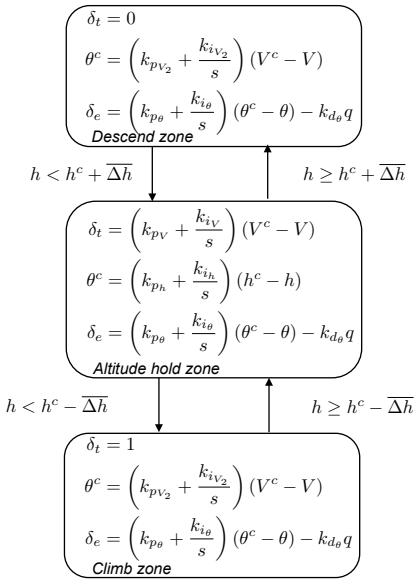

Figure 13: Altitude control state machine. The commanded altitude is h^c and $\bar{\delta h}$ defines a small altitude window around h^c.

4 On-Board Sensors and Sensor Processing

One of the biggest challenges with small UASs is to accurately estimate the dynamic states of the aircraft. Commercially available autopilots like the Kestrel, Piccolo II, and MP 2128 include the following sensor suite:

- Three-axis MEMS rate gyros,
- Three-axis MEMS accelerometers,
- Absolute and differential pressure sensors,
- GPS.

Optional sensors that may be available include a two-axis magnetometer and an ultrasonic range finder (for landing). In Section 3 we saw that the low-level autopilot requires knowledge of the following states:

- Roll rate p, pitch rate q, yaw rate r,
- Roll angle ϕ, pitch angle θ,
- Airspeed V, altitude h,
- Heading angle χ.

The navigation algorithm discussed in Section 5 will also make use of the North-East position of the aircraft, which will be denoted by p_n and p_e. The objective of this section is to describe how the available sensors are used to estimate the required states.

4.1 Angular rates, airspeed, and altitude

The angular rates, airspeed, and altitude of the aircraft can be estimated by directly inverting the sensor models. A brief discussion of this process is described below.

A MEMS rate gyro contains a small vibrating lever. When the lever undergoes an angular rotation, Coriolis effects change the frequency of the vibration, thus detecting the rotation. The output of the rate gyro is given by

$$y_{gyro} = k_{gyro}\Omega + \beta_{gyro} + \eta_{gyro}, \tag{36}$$

where y_{gyro} is in Volts, k_{gyro} is a gain, Ω is the angular rate in radians per second, β_{gyro} is a bias, and η_{gyro} is zero mean white noise. The gain k_{gyro} and the bias β_{gyro} are strongly dependent on temperature and need to be carefully measured. The commercially available autopilots described in Section 1 have preprogrammed firmware routines that include look-up tables for the gain and bias as a function of temperature. Unfortunately, this process does not completely compensate for the bias term β_{gyro} which drifts with time. Therefore, during the preflight phase, it is necessary to calibrate the rate gyros.

If three rate gyros are aligned along the x, y, and z axes of the vehicle, then the rate gyros measure the angular body rates p, q, and r as follows:

$$y_{gyro,x} = k_{gyro,x}p + \beta_{gyro,x} + \eta_{gyro,x} \tag{37}$$

$$y_{gyro,y} = k_{gyro,y}q + \beta_{gyro,y} + \eta_{gyro,y} \tag{38}$$

$$y_{gyro,z} = k_{gyro,z}r + \beta_{gyro,z} + \eta_{gyro,z}. \tag{39}$$

Therefore, assuming knowledge of the gain and bias, the angular rates p, q, and r can be estimated by low-pass filtering as

$$\hat{p} = \left(H_{lpf}(y_{\text{gyro,x}}) - \beta_{gyro,x}\right)/k_{gyro,x} \tag{40}$$

$$\hat{q} = \left(H_{lpf}(y_{\text{gyro,y}}) - \beta_{gyro,x}\right)/k_{gyro,x} \tag{41}$$

$$\hat{r} = \left(H_{lpf}(y_{\text{gyro,z}}) - \beta_{gyro,x}\right)/k_{gyro,x}, \tag{42}$$

where $H_{lpf}(\cdot)$ denotes the application of a low-pass filter, and where the "hat" notation denotes an estimated quantity.

The airspeed is measured using a pitot tube attached to a differential pressure sensor. If the pitot tube is oriented in the direction of motion, then the output of the differential pressure sensor is

$$y_{\text{diff pres}} = \frac{1}{2}\rho V^2 + \eta_{\text{diff pres}}, \tag{43}$$

where V is the airspeed, ρ is the density of air, and $\eta_{\text{diff pres}}$ is zero mean white noise. A simple estimation scheme for the airspeed is

$$\hat{V} = \sqrt{\frac{2}{\rho}H_{lpf}(y_{\text{diff pres}})}. \tag{44}$$

The altitude of the UAS is measured using an static pressure sensor, where the output of the sensor is given by

$$y_{\text{static pres}} = \rho g h + \eta_{\text{static pres}}, \tag{45}$$

where ρ is the density of air, g is the gravity constant, h is the altitude of the UAS, and $\eta_{\text{static pres}}$ is zero mean white noise. Therefore, the altitude of the UAS can be estimated as

$$\hat{h} = \frac{H_{lpf}(y_{\text{static pres}})}{\rho g}. \tag{46}$$

4.2 Roll and Pitch angles

The most difficult states to measure are the vehicle roll and pitch angles. The basic idea that we will discuss in this section is to estimate the roll and pitch angles by integrating the rate gyros and using the accelerometers to correct for integration constants and biases and drift in the rate gyros.

A MEMS accelerometer contains a small plate attached to torsion levers. The plate rotates under acceleration and changes the capacitance between the plate and the surrounding walls [9]. The output of a single-axis accelerometer is given by

$$y_{acc} = k_{acc}A + \beta_{acc} + \eta_{acc}, \tag{47}$$

where y_{acc} is in Volts, k_{acc} is a gain, A is the acceleration in meters per second, β_{acc} is a bias term, and η_{acc} is zero mean white noise. The gain k_{acc} and the bias term β_{acc} are strongly dependent on temperature. Therefore the accelerometer must be calibrated in a temperature chamber.

Accelerometers measure the specific force in the body frame of the vehicle. A physically intuitive explanation is given in [10, p. 13-15]. Additional explanation is given in [11, p. 27]. Mathematically we have

$$\begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} = \frac{1}{m}\left(\mathbf{F} - \mathbf{F}_{\text{gravity}}\right), \tag{48}$$

where $\mathbf{F}$ is the total force acting on the center of mass of the UAS, and $\mathbf{F}_{\text{gravity}}$ is the force due to gravity. For a flying vehicle, the total force is composed of four components:

$$\mathbf{F} = \mathbf{F}_{\text{thrust}} + \mathbf{F}_{\text{drag}} + \mathbf{F}_{\text{lift}} + \mathbf{F}_{\text{gravity}}, \tag{49}$$

where $\mathbf{F}_{\text{thrust}}$ is the thrust force, $\mathbf{F}_{\text{drag}}$ is the drag force, and $\mathbf{F}_{\text{lift}}$ is the lift force. For a fixed-wing aircraft in unaccelerated flight, the total forces on the UAS sum to zero. Therefore, from equation (48) we have that the accelerometers measure the direction of the gravity vector, which can be used to extract the roll and pitch angles.

We note here that while accelerometers can be used effectively to estimate roll and pitch for fixed-wing vehicles or for hovercraft in the forward motion regime, they are not useful for hovercraft in the hover regime. Near hover, the lift and drag forces are essentially zero. Therefore, from Equation (48) the accelerometers measure the thrust vector and not the gravity vector. Since the thrust vector is a body fixed quantity, the orientation of the aircraft cannot be extracted.

The acceleration of any rigid body is given by

$$\dot{\mathbf{v}} + \boldsymbol{\omega}_{b/i} \times \mathbf{v} = \frac{1}{m}\mathbf{F} \tag{50}$$

where $\mathbf{v} \triangleq (u, v, w)^T$ is the inertial velocity expressed in the body frame, and $\boldsymbol{\omega}_{b/i} \triangleq (p, q, r)^T$ is the angular velocity of the body with respect to the inertial frame, expressed in the body frame. Plugging Equation (50) into Equation (48), and expressing the gravity vector in terms of the roll angle ϕ and the pitch angle θ gives

$$a_x = \dot{u} + qw - rv + g\sin\theta \tag{51}$$

$$a_y = \dot{v} + ru - pw - g\cos\theta\sin\phi \tag{52}$$

$$a_z = \dot{w} + pv - qu - g\cos\theta\cos\phi, \tag{53}$$

where a_* is the specific acceleration along the $*$-axis of the body. If the gains of the accelerometers are calibrated and normalized to $1/g$, and the biases calibrated and removed, then the outputs of the accelerometers are given by

$$y_{acc,x} = \frac{\dot{u} + qw - rv + g\sin\theta}{g} + \eta_{acc,x} \tag{54}$$

$$y_{acc,y} = \frac{\dot{v} + ru - pw - g\cos\theta\sin\phi}{g} + \eta_{acc,y} \tag{55}$$

$$y_{acc,z} = \frac{\dot{w} + pv - qu - g\cos\theta\cos\phi}{g} + \eta_{acc,z}. \tag{56}$$

The body frame velocity vector can be expressed as

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = V \begin{pmatrix} \cos\alpha\cos\beta \\ \sin\beta \\ \sin\alpha\cos\beta \end{pmatrix}, \tag{57}$$

where α is the angle of attack and β is the sideslip angle [10]. Since α and β are typically not measured on small UAS, and since for small fixed-wing vehicles α and β are typically small, we assume that $\alpha \approx \theta$ and $\beta \approx 0$ to obtain

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} \approx V \begin{pmatrix} \cos\theta \\ 0 \\ \sin\theta \end{pmatrix}. \tag{58}$$

Approximating $\dot{u} = \dot{v} = \dot{w} = 0$, Equation (55) becomes

$$y_{acc,x} \approx \frac{qV\sin\theta}{g} + \sin\theta + \eta_{acc,x} \tag{59}$$

$$y_{acc,y} \approx \frac{rV\cos\theta - pV\sin\theta}{g} - \cos\theta\sin\phi + \eta_{acc,y} \tag{60}$$

$$y_{acc,z} \approx \frac{-qV\cos\theta}{g} - \cos\theta\cos\phi + \eta_{acc,z}. \tag{61}$$

Therefore, a simple inversion scheme that does not include the integration of rate gyros

is given by

$$\hat{\theta}_{\text{accel}} = \sin^{-1}\left[\frac{H_{lpf}(y_{acc,x})}{\frac{\hat{q}\hat{V}}{g}+1}\right], \tag{62}$$

$$\hat{\phi}_{\text{accel}} = \tan^{-1}\left[\frac{H_{lpf}(y_{acc,y}) - \left(\frac{\hat{r}\hat{V}\cos\hat{\theta} - \hat{p}\hat{V}\sin\hat{\theta}}{g}\right)}{H_{lpf}(y_{acc,z}) + \left(\frac{\hat{q}\hat{V}\cos\hat{\theta}}{g}\right)}\right]. \tag{63}$$

A fixed gain Kalman filter can be used to fuse the accelerometer and rate gyro information. The relationship between the angular rates $\dot{\phi}$ and $\dot{\theta}$ and the body fixed angular velocities p and q is given by [10]

$$\dot{\phi} = p + q\sin\phi\tan\theta + r\cos\phi\tan\theta \tag{64}$$

$$\dot{\theta} = q\cos\phi - r\sin\phi \tag{65}$$

$$= q + (q(\cos\phi - 1) - r\sin\phi). \tag{66}$$

If we lump the nonlinearities into noise variables and make the (admittedly bad) assumption that the noise will be zero mean and Gaussian, then we have

$$\dot{\phi} = p + \xi_\phi \tag{67}$$

$$\dot{\theta} = q + \xi_\theta, \tag{68}$$

where we assume that $\xi_\phi \sim \mathcal{N}(0, Q_\phi)$ and $\xi_\theta \sim \mathcal{N}(0, Q_\theta)$.

The accelerometers are used for measurement correction. Therefore

$$y_\phi = \hat{\phi}_{\text{accel}} = \phi + \eta_\phi \tag{69}$$

$$y_\theta = \hat{\theta}_{\text{accel}} = \theta + \eta_\theta, \tag{70}$$

where we again assume that $\eta_\phi \sim \mathcal{N}(0, R_\phi)$ and $\eta_\theta \sim \mathcal{N}(0, R_\theta)$. The steady-state Kalman filter equations are [12]

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x}) \tag{71}$$

$$L = PC^T R^{-1} \tag{72}$$

$$0 = AP + PA^T + GQG^T - PC^T R^{-1} CP. \tag{73}$$

The Riccati equation for the roll angle simplifies to

$$Q_\phi - \frac{P_\phi^2}{R_\phi} = 0. \tag{74}$$

Therefore we have

$$P_\phi = \sqrt{Q_\phi R_\phi}, \tag{75}$$

and the Kalman gain becomes

$$L_\phi = P_\phi R_\phi^{-1} = \sqrt{\frac{Q_\phi}{R_\phi}}. \tag{76}$$

We can calculate L_θ using similar reasoning. Therefore, the steady state Kalman filters for roll and pitch angles, which fuse rate gyro and accelerometer data, are given by

$$\dot{\hat{\phi}} = p + \sqrt{\frac{Q_\phi}{R_\phi}} \left(\hat{\phi}_{\text{accel}} - \hat{\phi} \right) \tag{77}$$

$$\dot{\hat{\theta}} = q + \sqrt{\frac{Q_\theta}{R_\theta}} \left(\hat{\theta}_{\text{accel}} - \hat{\theta} \right). \tag{78}$$

4.3 Inertial position and heading

This section describes an estimation scheme for the inertial position and heading using GPS as an input. GPS signals have an inertial position bias due to timing and other errors that will not not be corrected by the techniques discussed in this section. We assume that position, heading, and groundspeed are directly measured by the GPS unit.

The simplest technique for estimating position and heading is to simply low-pass filter the GPS signals. However, since the GPS update rate is on the order of one second, we desire to estimate the states in between GPS updates.

The kinematic equations of motion for inertial position and heading are given by

$$\begin{pmatrix} \dot{p}_n \\ \dot{p}_n \\ \dot{\chi} \end{pmatrix} = \begin{pmatrix} V_g \cos\chi \\ V_g \sin\chi \\ q\frac{\sin\phi}{\cos\theta} + r\frac{\cos\phi}{\cos\theta} \end{pmatrix} + \xi \tag{79}$$

where V_g is the ground speed of the UAS and ξ is a zero mean Gaussian process with covariance Q. Obviously Equation (79) has problems as θ approaches $\pm\pi/2$, and as such the scheme described in this section assumes relatively small pitch angles. If we let $x = (p_n, p_e, \chi)^T$ and $u = (V_g, q, r, \phi, \theta)^T$, then we have

$$\dot{x} = f(x, u) + \xi. \tag{80}$$

GPS returns measurements of p_n, p_e, and χ directly. Therefore the output model is

$$y_{gps} = \begin{pmatrix} p_n \\ p_e \\ \chi \end{pmatrix} + \eta, \tag{81}$$

where $\eta \sim \mathcal{N}(0, R)$ and $C = I$. To implement the Kalman filter we need the Jacobian of f which can be calculated as

$$\frac{\partial f}{\partial x} = \begin{pmatrix} 0 & 0 & -V_g \sin\chi \\ 0 & 0 & V_g \cos\chi \\ 0 & 0 & 0 \end{pmatrix}. \tag{82}$$

The resulting Kalman filter is listed in Algorithm 1.

Algorithm 1 Kalman Filter for p_n, p_e and χ

Initialize: $\hat{x} = 0$.
Pick an output sample rate T_{out} which is much less than the sample rates of the sensors.
At each sample time T_{out}:
for $i = 1$ to N **do** {Propagate the equations.}
  $\hat{x} = \hat{x} + \left(\frac{T_{out}}{N}\right) f(\hat{x}, u)$
  $A = \frac{\partial f}{\partial x}(\hat{x}, u)$
  $P = P + \left(\frac{T_{out}}{N}\right)\left(AP + PA^T + Q\right)$
end for
if GPS measurement received, **then** {Measurement Update}
  $L = P(R + P)^{-1}$
  $P = (I - L)P$
  $\hat{x} = \hat{x} + L\left(y_{gps} - \hat{x}\right)$.
end if

5 GPS Navigation

In this section we will describe guidance laws for tracking straight-line paths and orbits. The primary challenge in tracking straight line segments and circular orbits is the constant winds, which are almost always present. For small UAS, wind speeds are commonly 20 to 60 percent of the desired airspeed. Effective path tracking strategies must overcome the effect of this ever present disturbance. For most fixed-wing small UAS, the minimum turn radius is in the range of 10 to 50 m, which places a fundamental limit on the spatial frequency of paths that can be tracked.

Implicit in the notion of trajectory tracking is that the vehicle is commanded to be at a particular location at a specific time. For fixed-wing UAS, the desired position is constantly moving (at the desired ground speed), which can result in significant problems if wind disturbances are not accounted for properly. If the UAS is flying into a strong wind (relative to its commanded groundspeed), the progression of the trajectory point must be slowed accordingly. Similarly, if the UAS is flying down wind, the speed of the tracking point must be increased to keep it from overrunning the desired position. Given that wind disturbances vary and are often not easily predicted, trajectory tracking can be challenging in anything other than calm conditions.

Rather than using a trajectory tracking approach, this section focuses on path following where the objective is to be *on the path* rather than at a certain point at a particular time. With path following, the time dependence of the problem is removed. A key insight is to use groundspeed and heading as opposed to airspeed and yaw angle. By using groundspeed instead of airspeed and heading instead of yaw, asymptotic path following can be guaranteed, even in the presence of constant wind disturbances. The path following approach is discussed in more detail in [13, 14, 15, 16, 17].

This section is limited to constant altitude motion. The corresponding equations of

motion are given by

$$\dot{p}_n = V\cos\chi + w_n \tag{83}$$
$$\dot{p}_e = V\sin\chi + w_e, \tag{84}$$

where we will assume that the input to the guidance law is the heading angle χ, and where (w_n, w_e) are the magnitude of the wind vector in North-East coordinates. The objective is to develop a method for accurate path following in the presence of wind. For a given airframe, there is an optimal airspeed for which the airframe is the most aerodynamically efficient. Therefore, to conserve fuel, it is desirable that the UAS maintain a constant airspeed. Accordingly, in this section we will assume a constant airspeed V. The relationship between airspeed, windspeed, and groundspeed are shown in Figure 14. The equations of motion for the North and East directions can be written

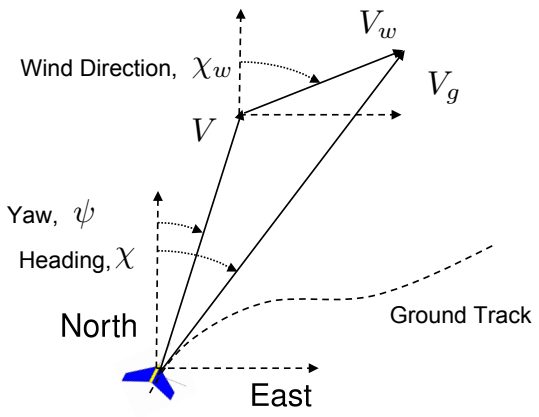

Figure 14: This figure shows the relationship between the airspeed V, the windspeed V_w, and ground speed V_g, as well as the relationship between yaw ψ, heading χ, and wind direction χ_w. The crab angle is defined as heading minus yaw.

in terms of the heading and groundspeed as

$$\dot{p}_n = V\cos\psi + V_w\cos\chi_w = V_g\cos\chi \tag{85}$$
$$\dot{p}_e = V\sin\psi + V_w\sin\chi_w = V_g\sin\chi, \tag{86}$$

where the groundspeed V_g is given by

$$V_g = \sqrt{V^2 + V_w^2 + 2VV_w\cos(\psi - \chi_w)}, \tag{87}$$

and the heading angle χ is given by

$$\chi = \tan^{-1}\left(\frac{V\sin\psi + V_w\sin\chi_w}{V\cos\psi + V_w\cos\chi_w}\right). \tag{88}$$

Section 5.1 will develop a guidance strategy for following straight line paths, and Section 5.2 will develop a guidance strategy for following constant altitude orbits.

5.1 Straight-Line Path Following

A straight line path is described by two vectors in $\mathbb{R}^2$, namely

$$\mathcal{P}_{\text{straight}}(\mathbf{r}, \vec{\mathbf{q}}) = \left\{ \mathbf{x} \in \mathbb{R}^2 : \mathbf{x} = \mathbf{r} + \alpha \vec{\mathbf{q}}, \alpha \in \mathbb{R} \right\}, \tag{89}$$

where $\mathbf{r} \in \mathbb{R}^2$ is the origin of the path, and $\vec{\mathbf{q}} \in \mathbb{R}^2$ is a unit vector whose direction indicates the desired direction of travel. Figure 15 shows a top down view of $\mathcal{P}_{\text{straight}}(\mathbf{r}, \vec{\mathbf{q}})$. The heading of $\mathcal{P}_{\text{straight}}(\mathbf{r}, \hat{\mathbf{q}})$, as measured from North is given by

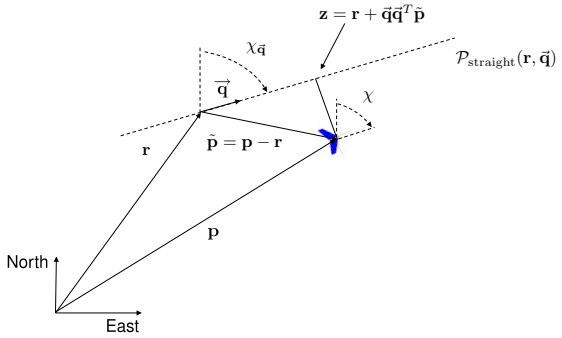

Figure 15: This figure shows the configuration of the UAS indicated by $(\mathbf{p}, \chi)$, and the configuration of the UAS relative to $\mathcal{P}_{\text{straight}}$ indicated by $(\tilde{\mathbf{p}}, \tilde{\chi})$.

$$\chi_{\vec{\mathbf{q}}} \triangleq \tan^{-1} \frac{q_e}{q_n}. \tag{90}$$

The path following problem is most easily solved in a frame relative to the straight-line path. Let

$$R_i^{\mathcal{P}} \triangleq \begin{pmatrix} \cos\chi_{\vec{\mathbf{q}}} & \sin\chi_{\vec{\mathbf{q}}} \\ -\sin\chi_{\vec{\mathbf{q}}} & \cos\chi_{\vec{\mathbf{q}}} \end{pmatrix}, \tag{91}$$

be the transformation from the inertial frame to the path frame, and let

$$\tilde{\mathbf{p}} = \begin{pmatrix} \tilde{p}_x \\ \tilde{p}_y \end{pmatrix} \triangleq R_i^{\mathbf{P}} (\mathbf{p} - \mathbf{r}) \tag{92}$$

be the relative path error in the path frame. Therefore, the relative dynamics in the path frame are given by

$$\begin{aligned} \begin{pmatrix} \dot{\tilde{p}}_x \\ \dot{\tilde{p}}_y \end{pmatrix} &= R_i^{\mathcal{P}} (\dot{\mathbf{p}}^i - \dot{\mathbf{r}}^i) && (93) \\ &= \begin{pmatrix} \cos\chi_{\vec{\mathbf{q}}} & \sin\chi_{\vec{\mathbf{q}}} \\ -\sin\chi_{\vec{\mathbf{q}}} & \cos\chi_{\vec{\mathbf{q}}} \end{pmatrix} \begin{pmatrix} V_g \cos\chi \\ V_g \sin\chi \end{pmatrix} && (94) \\ &= \begin{pmatrix} V_g \cos(\chi - \chi_{\vec{\mathbf{q}}}) \\ V_g \sin(\chi - \chi_{\vec{\mathbf{q}}}) \end{pmatrix}. && (95) \end{aligned}$$

Our strategy is to construct a desired heading angle at every point relative to the straight-line path that results in the UAS moving toward the path. The set of desired heading angles at every point will be called a vector field because the desired heading angle specifies a vector (relative to the straight line) with a magnitude of unity. Figure 16 is an example of a vector field for straight-line path following. The objective is

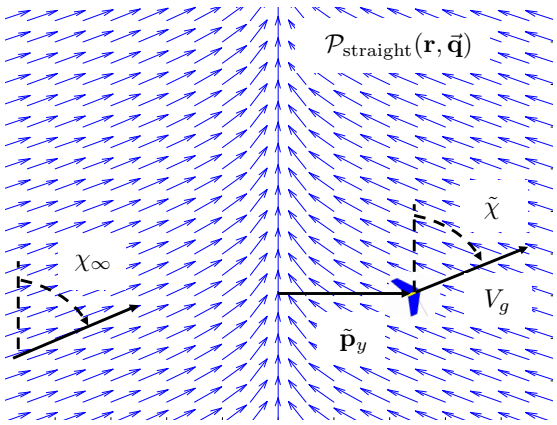

Figure 16: Vector field for straight-line path following. Far away from the waypoint path, the vector field is directed with an angle χ^∞ from the perpendicular to the path.

to construct the vector field so that when $\tilde{p}_y$ is large the UAS is directed to approach the path with heading angle χ^∞, and so that as $\tilde{p}_y$ approaches zero, the heading $\tilde{\chi} \triangleq \chi - \chi_{\vec{\mathbf{q}}}$ also approaches zero. Toward that end, define the commanded heading of the UAS as

$$\chi^c(\tilde{p}_y) = \chi_{\vec{\mathbf{q}}} - \chi^\infty \frac{2}{\pi} \tan^{-1}(k\tilde{p}_y), \tag{96}$$

where k is a positive constant that influences the rate of the transition from χ^∞ to zero. Figure 17 shows how the choice of k affects the rate of transition. Large values of k result in short, abrupt transitions, while small values of k cause long, smooth transitions in the desired heading.

If χ^∞ is restricted to be in the range $\chi^\infty \in (0, \frac{\pi}{2}]$ then clearly

$$-\frac{\pi}{2} < \chi^\infty \frac{2}{\pi} \tan^{-1}(k\tilde{p}_y) < \frac{\pi}{2} \tag{97}$$

for all values of $\tilde{p}_y$. Therefore, since $\tan^{-1}(\cdot)$ is an odd function and $\sin(\cdot)$ is odd over $(-\frac{\pi}{2}, \frac{\pi}{2})$, we can use the Lyapunov function $W_1(\tilde{p}_y) = \frac{1}{2}\tilde{p}_y^2$ to argue that if $\chi = \chi^c(\tilde{p}_y)$, then $\tilde{p}_y \to 0$ asymptotically. Evaluating the Lie derivative of W_1 under the assumption that $\chi = \chi^c(\tilde{p}_y)$ gives

$$\dot{W}_1 = -V_g \tilde{p}_y \sin\left(\tilde{\chi}^\infty \frac{2}{\pi} \tan^{-1}(k\tilde{p}_y)\right), \tag{98}$$

which is less than zero for $\tilde{p}_y \neq 0$. Asymptotic convergence of this scheme in the presence of yaw and roll dynamics is shown in [13].

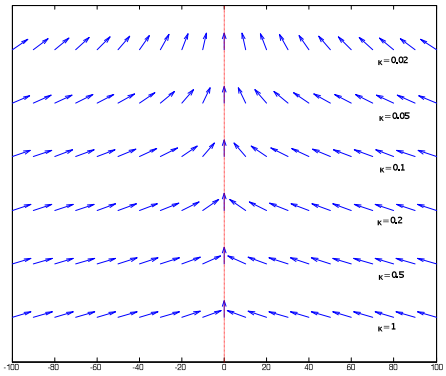

Figure 17: Vector fields for various values of k. Large values of k yield abrupt transitions from χ^∞ to zero, while small values of k give smooth transitions.

5.2 Orbit Following

An orbit is described by a center $\mathbf{c} \in \mathbb{R}^2$, a radius $R \in \mathbb{R}$, and a direction $\lambda \in \{-1, 1\}$, as

$$\mathcal{P}_{\text{orbit}}(\mathbf{c}, R, \lambda) = \left\{ \mathbf{r} \in \mathbb{R}^2 : \mathbf{r} = \mathbf{c} + \lambda R \begin{pmatrix} \cos\varphi, & \sin\varphi \end{pmatrix}^T, \varphi \in [0, 2\pi) \right\}, \quad (99)$$

where $\lambda = 1$ signifies a clockwise orbit and $\lambda = -1$ signifies a counterclockwise orbit. Figure 18 shows a top down view of an orbital path. The guidance strategy for orbit

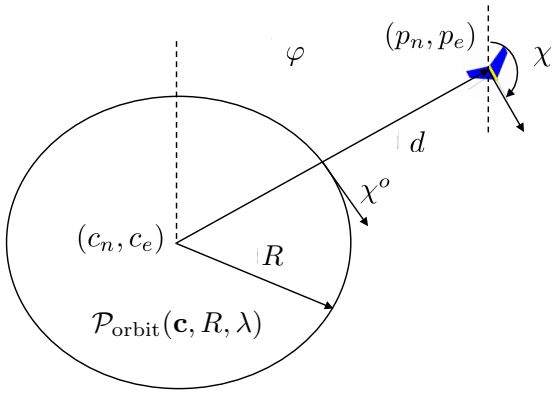

Figure 18: Orbital path with center (c_n, c_e), and radius R. The distance from the orbit center to the UAS is d, and the angular position of the UAS relative to the orbit is φ.

following is best derived in polar coordinates. Let

$$d \triangleq \sqrt{(p_n - c_n)^2 + (p_e - c_e)^2} \quad (100)$$

be the lateral distance from the desired center of the orbit to the UAS, and let

$$\varphi \triangleq \tan^{-1}\left(\frac{p_e - c_e}{p_n - c_n}\right) \tag{101}$$

be the phase angle of the relative position, as shown in Figure 18. Differentiating d and using Equations (85) and (86) gives

$$\dot{d} = \frac{(p_n - c_n)\dot{p}_n + (p_e - c_e)\dot{p}_e}{d} \tag{102}$$

$$= \frac{(p_n - c_n)V_g \cos\chi + (p_e - c_e)V_g \sin\chi}{d}. \tag{103}$$

Using Equation (101) gives

$$\dot{d} = V_g \frac{(p_n - c_n)\cos\chi + (p_e - c_e)\sin\chi}{d} \tag{104}$$

$$= V_g \left(\frac{p_n - c_n}{d}\right)(\cos\chi + \sin\chi\tan\varphi) \tag{105}$$

$$= V_g \cos\varphi\,(\cos\chi + \sin\chi\tan\varphi) \tag{106}$$

$$= V_g\,(\cos\chi\cos\varphi + \sin\chi\sin\varphi) \tag{107}$$

$$= V_g \cos(\chi - \varphi). \tag{108}$$

As shown in Figure 18, for a clockwise orbit, the desired heading angle when the UAS is located on the orbit is given by $\chi^o = \varphi + \pi/2$. Similarly, for a counterclockwise orbit, the desired angle is given by $\chi^o = \varphi - \pi/2$. Therefore, in general we have

$$\chi^o = \varphi + \lambda\frac{\pi}{2}. \tag{109}$$

The control objective is to drive $d(t)$ to the orbit radius R. Defining the error variable $\tilde{d} \triangleq d - R$, the orbital kinematics can be restated as

$$\dot{\tilde{d}} = -\lambda V_g \sin(\chi - \chi^o). \tag{110}$$

Our approach to orbit following is similar to the ideas developed in Section 5.1. The strategy is to construct a desired heading field that moves the UAS onto the orbit $\mathcal{P}_{\text{orbit}}(\mathbf{c}, R, \lambda)$. When the distance between the UAS and the center of the orbit is large, it is desirable for the UAS to fly toward the orbit center. In other words, when $\tilde{d} >> R$ the desired heading is

$$\tilde{\chi}^d \approx \chi^o + \lambda\frac{\pi}{2}, \tag{111}$$

and when $\tilde{d} = 0$ the desired heading is χ^o. The proposed guidance law is given by

$$\chi^c(\tilde{d}) = \chi^o + \lambda\tan^{-1}\left(k\tilde{d}\right), \tag{112}$$

where $k > 0$ is a constant that specifies the rate of transition from $\lambda\pi/2$ to zero. This expression for $\chi^c(\tilde{d})$ is valid for all values of $\tilde{d} \geq -R$.

Consider the Lyapunov function $W_1 = \frac{1}{2}\tilde{d}^2$. When $\chi = \chi^c(\tilde{d})$, the Lie derivative of W_1 is

$$\dot{W}_1 = -\lambda V_g \tilde{d} \sin\left(\lambda \tan^{-1}(k\tilde{d})\right) \tag{113}$$

$$= -V_g \tilde{d} \sin\left(\tan^{-1}(k\tilde{d})\right), \tag{114}$$

which is negative definite since the argument of $\sin$ is in the set $(-\pi/2, \pi/2)$ for all $\tilde{d}$, implying that $\tilde{d} \to 0$ asymptotically. Asymptotic convergence of this scheme in the presence of yaw and roll dynamics is shown in [13].

6 Summary

The objective of this article was to present basic principles in the design of autopilots for small UAS. We presented a basic architecture that is common to commercially available autopilots for small UAS. We discussed in detail the design of the feedback loops and how those loops are structured to control the lateral and longitudinal modes of the aircraft. Small UAS have a limited sensor suite. Common sensors were discussed and simple techniques for extracting the state of the system were described. We also presented a method for tracking waypoint paths and circular orbits.

Acknowledgments

The author gratefully acknowledges many fruitful discussions about the topics of this article with his colleague Tim McLain at BYU.

References

[1] "Procerus technologies," http://procerusuav.com/.

[2] "Cloudcap technology," http://www.cloudcaptech.com.

[3] "Micropilot," http://www.micropilot.com/.

[4] J. H. Blakelock, *Automatic Control of Aircraft and Missiles*, second edition ed. John Wiley & Sons, 1991.

[5] R. C. Nelson, *Flight Stability and Automatic Control*, 2nd ed. Boston, Massachusetts: McGraw-Hill, 1998.

[6] B. Etkin and L. D. Reid, *Dynamics of Flight: Stability and Control*. John Wiley & Sons, 1996.

[7] J. Roskam, *Airplane Flight Dynamics and Automatic Flight Controls, Parts I & II*. Lawrence, Kansas: DARcorporation, 1998.

[8] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 4th ed. Addison Wesley, 2002.

[9] http://www.silicondesigns.com/tech.html.

[10] B. L. Stevens and F. L. Lewis, *Aircraft Control and Simulation*, 2nd ed. Hoboken, New Jersey: John Wiley & Sons, Inc., 2003.

[11] M. Rauw, *FDC 1.2 - A SIMULINK Toolbox for Flight Dynamics and Control Analysis*, February 1998, available at http://www.mathworks.com/.

[12] Dorato, Abdallah, and Cerone, *Linear-Quadratic Control: An Introduction*. Prentice Hall, 1995.

[13] D. R. Nelson, D. B. Barber, T. W. McLain, and R. W. Beard, "Vector field path following for miniature air vehicles," *IEEE Transactions on Robotics*, vol. 37, no. 3, pp. 519–529, June 2007.

[14] A. P. Aguiar, J. P. Hespanha, and P. V. Kokotovic, "Path-following for nonminimum phase systems removes performance limitations," *IEEE Transactions on Automatic Control*, vol. 50, no. 2, pp. 234–238, February 2005.

[15] P. Encarnação and A. Pascoal, "Combined trajectory tracking and path following: An application to the coordinated control of marine craft," in *Proceedings of the IEEE Conference on Decision and Control*, Orlando, FL, 2001, pp. 964–969.

[16] R. Skjetne, T. Fossen, and P. Kokotović, "Robust output maneuvering for a class of nonlinear systems," *Automatica*, vol. 40, pp. 373–383, 2004.

[17] R. Rysdyk, "UAV path following for constant line-of-sight," in *Proceedings of the AIAA 2nd Unmanned Unlimited Conference*. AIAA, September 2003, paper no. AIAA-2003-6626.