

Research Article Lateral Control Implementation for an Unmanned Aerial Vehicle

R. Samar,¹ M. Zamurad Shah,¹ and M. Nzar²

¹ Mohammad Ali Jinnah University, Islamabad 44000, Pakistan

² Centres of Excellence in Science & Applied Technologies (CESAT), Islamabad, Pakistan

Correspondence should be addressed to R. Samar; raza.samar@gmail.com

Received 31 March 2013; Accepted 29 April 2013

Academic Editors: A. Desbiens, Z. Mazur, and K. Peng

Copyright © 2013 R. Samar et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents practical aspects of guidance and control design for UAV and its flight test results. The paper focuses on the lateral-directional control and guidance aspects. An introduction to the mission and guidance problem is given first. Waypoints for straight and turning flight paths are defined. Computation of various flight path parameters is discussed, including formulae for real-time calculation of down-range (distance travelled along the desired track), cross-track deviation, and heading error of the vehicle; these are then used in the lateral guidance algorithm. The same section also describes how to make various mission-related decisions online during flight, such as when to start turning and when a waypoint is achieved. The lateral guidance law is then presented, followed by the design of a robust multivariable H_{∞} controller for roll control and stability augmentation. The controller uses the ailerons and rudder for control of roll angle and stabilization of yaw rate of the vehicle. The reference roll angle is generated by the nonlinear guidance law. The sensors available on-board the vehicle do not measure yaw rate; hence, a practical method of its estimation is proposed. The entire guidance and control scheme is implemented on the flight control computer of the actual aerial vehicle and taken to flight. Flight test results for different mission profiles are presented and discussed.

1. Introduction

Successful control system design for high performance UAVs requires efficient and effective techniques for the design of guidance and control algorithms that ensure satisfactory operation in the face of system uncertainties. In this paper, we develop and present one such technique. The UAV under consideration is shown in Figure 1. The cruising speed of said UAV is about 45 m/sec. The main tasks of the guidance and control system are to:

- (i) fly the vehicle on the desired mission path, with minimum cross-track deviation,
- (ii) make decisions about various flight events, such as to start/stop turning or achievement of waypoints,
- (iii) provide robust stabilization and control during flight.

A lateral track control law for small UAVs has been discussed in [1]. This is based on a pure geometrical concept. The idea is to make the ratio of lateral deviation to lateral velocity equal to the ratio of longitudinal distance to longitudinal velocity. Simulation results indicate that the yaw rate command generated by the guidance law exhibits oscillations in the vehicle roll channel, this could be a problem for implementation on a real vehicle. A receding horizon based controller for UAV guidance is presented in [2]. The dynamics of the problem are linearized and simplified, and a quadratic cost function based on the model predictive control methodology is set up. The output prediction horizon and the control horizon need to be selected keeping in view the computational resources available, and the actuator and vehicle dynamical constraints have to be modelled carefully. The problem is challenging with regards to real-time implementation (specially for critical applications like high performance UAVs), and infeasibility conditions for solutions have to be guarded against. Mixed integer linear programming-(MILP-) based guidance for UAVs has also been considered [3, 4]; however, here the optimization program generates a sequence of waypoints (positions) and velocities for the vehicle to follow. In other words a mission plan is generated, and deviations from this plan need to be corrected through a lower level guidance algorithm.



FIGURE 1: A photograph of the test vehicle.

A conceptually different guidance scheme employing vector fields for curved path following has also been pursued, see, for example, [5, 6]. Here, a vector field of course (or heading) commands is generated which is a function of vehicle position relative to the desired track. The difference between the actual and commanded headings forms the heading error which is driven to zero using an appropriate control algorithm. The course vector field can in cases give rise to large and sudden heading changes which can tax the capability of the control system.

The contribution of this paper is to present a practical and integrated guidance and control approach and demonstrate its effectiveness by experimental testing. Formulae for basic flight path computations are derived that provide the requisite input parameters for the guidance algorithm to function. The guidance algorithm is a computationally inexpensive algorithm based on the work presented in [7] with some improvements for practical considerations. The output of the guidance algorithm drives the control system, and design of a robust multivariable controller that tracks the guidance commands is presented. The overall system is integrated in a seamless fashion and implemented and test flown on an actual vehicle.

This paper is organized as follows. Section 2 describes the mission plan on which we intend the UAV to fly, including definition of straight, turn, and loiter waypoints. Here, we also outline the guidance and control problem in specific terms. Section 3 develops the basic formulae used for flight path computation. These form the basis for taking real-time decisions during flight and for implementing the guidance law. The nonlinear guidance law is discussed in Section 4. Section 5 describes the design of a robust multivariable H_{∞} controller and discusses robustness of the controller across the flight envelope. Flight test results are presented and discussed in Section 6; Section 7 concludes the paper.

2. The Mission Plan

The mission of the UAV consists of a number of waypoints which define the path the UAV is required to take. The mission may be planned on-line or off-line. Here, we assume that information regarding the waypoints through which the UAV is desired to navigate is available; our main objective here is to discuss the practical aspects of how to guide and control the aerial vehicle through the given mission. The entire mission can be represented by a series interconnection of straight line segments, arcs, and circles (for loiter). Figure 2 shows on the left-hand side, part of a mission comprising of a straight line segment. In this case, the UAV is required to fly through the waypoint WP2. The right side of Figure 2 shows part of a mission with a turn. Here, the UAV is not required to go through the central waypoint (WP2) but rather fly on a circular arc AB close to WP2. Point A is where the turn starts, and B is where it ends. Similarly, loiter missions are also possible where the vehicle flies in a circular orbit around a given waypoint. Figure 3 shows part of a mission which involves loiter at WP2. The arcs AB and CD are used for transitioning from the straight flight segments to the circular loiter pattern. In this study, we shall assume that the mission data in terms of waypoints is available.

We now formulate the problem in more specific terms. The task of the lateral guidance and control system is defined as to

- (i) compute the down-range covered from one waypoint to the next and the cross-track deviation from the designated track/trajectory,
- (ii) compute the error in heading angle of the vehicle,
- (iii) determine if a waypoint is achieved,
- (iv) compute "turn start" and "turn stop" decision flags for the autopilot,
- (v) solve the guidance equations to calculate the roll reference command for steering the vehicle back onto the desired path,
- (vi) stabilize and control the lateral-directional dynamics of the vehicle across the flight envelope in the presence of disturbances.

All the above computations have to be done in real time. The available control inputs to the vehicle are the ailerons and the rudder. Note that the cross-track deviation and heading error are used in the guidance equations.

3. Flight Path Computations

Here, we present the basic flight path computations that are performed in real time to enable navigation of the vehicle through a sequence of straight and turning waypoints. Waypoints are defined as geographic positions in terms of, latitude and longitude. For any three consecutive waypoints WP1, WP2 and WP3, if the turn angle (defined as the difference of azimuths of WP1 and WP3 at WP2) at the central waypoint (WP2) is nearly 180°, then the path from WP1 to WP2 will be considered a straight line, and WP2 will be referred to as a *straight* waypoint. On the other hand, if the turn angle is significant (say greater than 5°), then WP2 will be referred to as a *turning* waypoint, and the path from WP1 to WP3 will consist of straight and curved parts (Figure 2). The flight path computations are worked out separately for straight and turning parts of the flight.

The algorithm presented computes deviations from the reference track in the form of cross-track and heading errors both for straight and curved paths. The cross-track and heading errors are used in the guidance logic to generate appropriate roll commands for the autopilot so that the



FIGURE 2: Waypoints for straight and turning flight.





FIGURE 3: Mission with loiter about WP2.

vehicle is steered back onto the desired path. In addition to cross-track and heading errors, "waypoint achieved," "turn start," and "turn stop" decision flags are also computed for the autopilot.

Let WP1(ϕ_1 , λ_1), WP2(ϕ_2 , λ_2), and WP3(ϕ_3 , λ_3) be three consecutive waypoints (ϕ and λ denote the latitude and longitude, resp.), and let $M(\phi_M, \lambda_M)$ be the current position of the vehicle. We define an Earth-Centred Earth-Fixed (ECEF) frame with its origin at the centre of mass of the earth, the *z*-axis directed northward along the polar axis, the *x*-axis in the equatorial plane and passing through the Greenwich Meridian, and the *y*-axis also in the equatorial plane and passing through 90° east longitude. The position vector of a point on the earth's surface with latitude ϕ and longitude λ in ECEF frame is given by

$$r_{x} = a \cos \phi_{r} \cos \lambda, \qquad r_{y} = a \cos \phi_{r} \sin \lambda, r_{z} = b \sin \phi_{r},$$
(1)

where *a* is the earth's equatorial radius, *b* is the polar radius, and $\phi_r = \tan^{-1}((b/a) \tan \phi)$ is the reduced latitude of the point under consideration.

3.1. Straight Path Formulae

3.1.1. Distance Covered. Here, we consider the straight path from WP1 to WP2 (Figure 4). Let $\overrightarrow{r_1}$ and $\overrightarrow{r_2}$ be the position vectors of WP1 and WP2 in ECEF frame, respectively. The distance between WP1 and WP2 can be calculated as:

$$r_{12} = \frac{1}{2} \left(\left| \overrightarrow{r_1} \right| + \left| \overrightarrow{r_2} \right| \right) \cos^{-1} \left(\widehat{r_1} \cdot \widehat{r_2} \right), \tag{2}$$

where \hat{r} is a unit vector along \overrightarrow{r} .

3.1.2. Cross-Track Deviation. Cross-track error is the instantaneous normal displacement of the vehicle from the desired track. To compute the cross-track error, we first define a vector $\overrightarrow{r_N}$ normal to the trajectory plane (plane containing WP1, WP2, and the centre of the earth) as:

$$\overrightarrow{r_N} = \widehat{r}_1 \times \widehat{r}_2. \tag{3}$$

If $\overrightarrow{r_M}$ is the position vector of the vehicle in ECEF frame, then the cross-track error is

$$Cross track = -\overrightarrow{r_M} \cdot \widehat{r}_N.$$
(4)

3.1.3. Heading Error. Let V_E and V_N denote the east and north velocity of the vehicle, respectively, then the velocity heading of the vehicle is given by $\psi_G = \tan^{-1}(V_E/V_N)$. The heading error is the difference between the actual velocity heading ψ_G and the desired heading ψ_R ; that is, $\psi_E = \psi_G - \psi_R$.

If *P* is the projection of the vehicle's current position on the trajectory plane (Figure 4), then the desired heading angle (ψ_R) is defined as the angle between the trajectory plane and



FIGURE 4: Straight flight path definitions.

the meridian plane passing through the projected point *P*. The position vector $\overrightarrow{r_P}$ of *P* in ECEF frame is given by

$$\overrightarrow{r_P} = \overrightarrow{r_M} - \left(\overrightarrow{r_M} \cdot \widehat{r}_N\right)\widehat{r}_N.$$
(5)

The desired heading angle ψ_R works out to be

$$\psi_R = \tan^{-1} \left[\frac{\left| \left(\hat{r}_P \times \hat{k} \right) \times \hat{r}_N \right|}{\left(\hat{r}_P \times \hat{k} \right) \cdot \hat{r}_N} \right],\tag{6}$$

where \vec{k} is a unit vector along the earth's spin axis. The heading error ψ_E can now be readily computed.

3.1.4. Waypoint Achieved Criterion. The straight waypoint WP2 is achieved when the straight path distance covered exceeds r_{12} . The straight distance covered from WP1 to the current position of the vehicle is $(1/2)(|\vec{r_1}| + |\vec{r_P}|)\cos^{-1}(\hat{r_1} \cdot \hat{r_P})$.

3.2. Curved Path Formulae. For turning paths (Figure 5), we define another coordinate system (the East-North-Up or ENU frame) at the central waypoint WP2. The origin of the ENU frame is at WP2, the positive *x*-axis points towards east, the *y*-axis points towards north, and the *z*-axis points up along the local vertical. The position vector of a point on the earth's surface with latitude ϕ and longitude λ can be expressed in the ENU frame as:

$$x = R_N (\lambda - \lambda_2) \cos \phi_2, \qquad y = R_M (\phi - \phi_2), \qquad z = 0,$$
(7)

where R_N and R_M are the normal and meridian radii of curvature (of the earth's ellipsoidal model), respectively. These are given by

$$R_{N} = \frac{a}{\left(1 - e^{2}\sin\phi_{2}\right)^{1/2}},$$

$$R_{M} = \frac{a\left(1 - e^{2}\right)}{\left(1 - e^{2}\sin\phi_{2}\right)^{3/2}},$$
(8)



FIGURE 5: Flight path with turn.

where *e* denotes the eccentricity of the ellipsoidal earth:

$$e = \frac{\sqrt{a^2 - b^2}}{a}.$$
 (9)

Let $\vec{q_1}$ and $\vec{q_2}$ be the position vectors of WP1 and WP3 in the ENU frame; that is,

$$\vec{q_1} = \begin{bmatrix} R_N (\lambda_1 - \lambda_2) \cos \phi_2 & R_M (\phi_1 - \phi_2) & 0 \end{bmatrix},$$

$$\vec{q_2} = \begin{bmatrix} R_N (\lambda_3 - \lambda_2) \cos \phi_2 & R_M (\phi_3 - \phi_2) & 0 \end{bmatrix}.$$
(10)

The turn angle β (≤180°) is given by

$$\beta = \cos^{-1}\left(\widehat{q}_1 \cdot \widehat{q}_2\right). \tag{11}$$

3.2.1. Turn Centre. The turn centre will be at a distance of $R/\sin(\beta/2)$ from WP2 on the line bisecting the turn angle β , R being the turn radius. The vector \vec{q} along this line in the ENU frame is given by $\vec{q} = \hat{q}_1 + \hat{q}_2$. Thus, the position of the turn centre C in the ENU frame is

$$\vec{C} = \frac{R}{\sin\left(\beta/2\right)\hat{q}}.$$
(12)

The projection of the vehicle on the earth's surface (ϕ_M, λ_M) in the ENU frame is given as:

$$\vec{M} = \left[R_N \left(\lambda_M - \lambda_2 \right) \cos \phi_2 \ R_M \left(\phi_M - \phi_2 \right) \ 0 \right], \quad (13)$$

and the position vector of the vehicle's projection w.r.t. the turn centre *C* is $\overrightarrow{S_M} = \overrightarrow{M} - \overrightarrow{C}$.

3.2.2. Turn Start Criterion. The turn start point is at a distance of $R/\tan(\beta/2)$ from WP2 on the line joining WP1 and WP2. Its position vector \overrightarrow{A} in the ENU frame is $\overrightarrow{A} = R/\tan(\beta/2)\widehat{q}_1$, and its position w.r.t. the turn centre C is $\overrightarrow{S}_A = \overrightarrow{A} - \overrightarrow{C}$. Turning is started when the cross-product $\overrightarrow{S}_M \times \overrightarrow{S}_A$ changes sign. During turn, the total distance (down-track) from WP1 to the current position is computed as:

distance from WP1 =
$$r_{12} - \frac{R}{\tan(\beta/2)} + R\cos^{-1}(\widehat{S}_A \cdot \widehat{S}_M)$$
. (14)

3.2.3. Cross-Track Deviation. The cross track deviation during turning is given by $R - |\overrightarrow{S_M}|$.

3.2.4. Heading Error. During turn, the desired heading angle (ψ_R) is given by $\psi_R = \psi_s + \psi_T$, where ψ_s is the desired heading at the turn start point (computed using (6) at the turn start point) and ψ_T is the angular displacement from the turn start point to the current position:

$$\psi_T = \cos^{-1}\left(\widehat{S}_A \cdot \widehat{S}_M\right). \tag{15}$$

3.2.5. Waypoint Achieved Criterion. The waypoint WP2 is said to be achieved when the vehicle crosses the bisector line; that is, when the vector cross-product $\overrightarrow{S_M} \times \overrightarrow{C}$ changes sign.

3.2.6. Turn Stop Criterion. The turn stop point is at a distance of $R/\tan(\beta/2)$ from WP2 on the line joining WP2 and WP3. Its position vector \vec{B} in the ENU frame is $\vec{B} = R/\tan(\beta/2)\hat{q}_2$, and its position w.r.t. the turn centre C is $\vec{S}_B = \vec{B} - \vec{C}$. Turning is stopped when $\vec{S}_M \times \vec{S}_B$ changes sign.

4. Lateral Guidance Logic

The block diagram of the lateral-directional guidance and control system is shown in Figure 6. The outputs of the UAV that are measured by on-board sensors include the roll, pitch, and heading attitude angles and the (GPS) position and velocity of the vehicle. The roll and pitch angles are measured by a vertical gyro, and the heading angle is provided by a magnetic sensor. The outer loop is the lateral guidance loop that looks at the measured position and velocity of the vehicle, compares it with the desired mission path, and generates a roll angle command that acts as a reference for the inner loop to track. The inner loop has roll angle and yaw rate feedbacks; the yaw-rate is computed from the attitude angles as discussed in Section 5.1 below.

The lateral guidance scheme used is based on the work in [7]. The basic scheme is modified by introducing an adaptive tuning of the reference length to enhance the performance for large cross-track errors. Integral action is also added to improve tracking in the presence of disturbances. Here, we first introduce the basic scheme, followed by a discussion of the modifications.

4.1. The Basic Guidance Scheme. Figure 7 illustrates the guidance scheme. Let AB be the desired flight path of the vehicle, and let the instantaneous position of the vehicle be denoted by C, at a perpendicular distance y from the line AB. We define a point D at a distance L_1 from C such that the vehicle may follow a circular arc of radius R from C to the desired path at D. The centripetal acceleration a_L required for the vehicle to fly along this circular arc is given by

$$a_L = \frac{2V^2}{L_1} \sin \eta, \tag{16}$$

where V is the velocity of the vehicle and η is the angle between the velocity vector and the line CD. Since we shall

employ a bank-to-turn scheme, the lateral acceleration command needs to be converted into a roll reference command. When the vehicle is banked at an angle ϕ , the lift vector *L* can be resolved into two components $L \cos \phi = mg$ and $L \sin \phi = (2mV^2/L_1) \sin \eta$, from which, we have

$$\phi = \tan^{-1} \left(\frac{2V^2}{gL_1} \sin \eta \right). \tag{17}$$

From Figure 7, we have $\eta = \eta_1 + \eta_2$, where the subangles are

$$\eta_1 = \sin^{-1}\left(\frac{y}{L_1}\right), \qquad \eta_2 = \psi_E, \tag{18}$$

where ψ_E is the heading error. Here, $y \leq L_1$ because the argument of the arcsin function should be less than or equal to 1. Now, we may write (17) as follows:

$$\phi_c = \tan^{-1} \left[\frac{2V^2}{gL_1} \sin \left\{ \sin^{-1} \left(\frac{y}{L_1} \right) + \psi_E \right\} \right], \quad (19)$$

where the subscript in ϕ_c stands for roll angle *command*. Equation (19) defines the nonlinear lateral guidance law; it gives the roll angle which the vehicle should fly to get the lateral acceleration required to follow a circular arc back to its desired flight path.

4.2. Selection of L_1 Using Linear Analysis. A guideline for selection of the length L_1 is provided in [7] by linearizing (19) to yield a second-order system with damping and natural frequency given by

$$\zeta = \frac{1}{\sqrt{2}}, \qquad \omega_n = \frac{\sqrt{2}V}{L_1}, \qquad (20)$$

where ζ and ω_n refer to the damping ratio and natural frequency of the standard second order system. It is stressed, however, that this is only a guideline as a very strong assumption of no inner loop dynamics ($a_L = -\ddot{y}$) is employed in its derivation.

4.3. Adaptive Adjustment of Length L_1 . For system stability, we must have $y < L_1$ and $|\eta| < \pi/2$ [7]. For UAVs however, there is a real possibility of large cross-track errors developing during flight. This can be due to the runway not being in the direction of the first waypoint so that a cross-track deviation is necessarily generated in the initial phase of the flight or the GPS signal becoming masked for extended periods during flight. Furthermore, since lateral control is usually engaged sometime after take-off (to avoid large roll angles during takeoff), the cross-track may build to a large value initially. In such cases, the condition $y < L_1$ may not be satisfied making ϕ_c undefined. Table 1 gives values of L_1 computed using (20) for different speeds typical of modern UAVs. The table shows that for small-to-medium UAVs, L_1 can vary from 200 m to 850 m depending upon the desired response time of the guidance algorithm (or ω_n). If a fast response is desired (high ω_n), a small L_1 will be selected, leading to large roll angles to bring the vehicle back on the desired path quickly, but with



FIGURE 6: Block diagram of the overall guidance and control system.



FIGURE 7: Nonlinear guidance logic setup.

possibly high overshoots. A larger L_1 will make the system slower and reduce overshoot. This is discussed in [8] with reference to gain selection for a conventional proportionalderivative guidance law where the gains are scheduled with the magnitude of the cross-track error. It may be noted here that to cater for large cross-track errors (which are possible as discussed above), L_1 will have to be kept unduly large which may yield poor performance otherwise. On the other hand, if a relatively small L_1 is selected, the resulting roll angle might become too large and exceed the set limit. Table 2 shows the commanded roll angle ϕ_c for different cross-track errors for a fixed speed and L_1 chosen to yield an ω_n of 0.2 rad/sec. The last row of the table gives a ϕ_c of 55°, which may be too large for most vehicles, and thereafter for cross tracks larger than L_1 , ϕ_c becomes undefined. A solution to this is to introduce an adaptive adjustment of L_1 with cross-track error [9]. We

TABLE 1: L_1 (m) for different flight speeds and ω_n .

Speed (m/s)	$\omega_n = 0.1 \text{ rad/s}$	$\omega_n = 0.2 \text{ rad/s}$	$\omega_n = 0.3 \text{ rad/s}$
40	565	280	190
50	705	350	235
60	850	425	285

TABLE 2: ϕ_c for different cross-track errors ($\omega_n = 0.2 \text{ rad/s}$).

Cross-track deviation (% of L_1)	ϕ_c (deg)
12.5	23
25	31
50	43
100	55

introduce a compensated reference length \hat{L}_1 to be used in (19) instead of a fixed L_1 :

$$\hat{L}_1 = L_1 + k_1 |y|, \qquad (21)$$

where $k_1 > 0$ is a gain to be tuned (a value of 1.5 for our application gave good results). Roll angle commands are computed using L_1 and \hat{L}_1 and compared (for V = 50 m/s and $\omega_n = 0.2$ rad/s) in Table 3. The table shows that for large cross-track errors, the roll angle command computed using \hat{L}_1 remains reasonably small and can be designed to lie within a desired range.

4.4. Addition of Integral Action. The roll angle command as given in (19) is modified by addition of an integrator for strict path tracking in the presence of constant (wind) disturbances. Addition of an integrator in the loop can pose problems such as wind-up leading to control surface saturation, for which the following actions are taken:

(i) integral action is activated only when the cross-range error becomes smaller than a set threshold y_{th} ,

Cross-track deviation (% of L_1)	ϕ_c (deg) using fixed L_1	ϕ_c (deg) using \hat{L}_1
12.5	23	19
25	31	21
50	43	21
100	55	19
200	Undefined	14

TABLE 3: ϕ_c computed using L_1 and \hat{L}_1 .

(ii) the integrator state is limited to a predefined value; the integrator is switched off if its state exceeds the set limit.

With the addition of integral action and substitution of \hat{L}_1 in place of L_1 , (19) becomes

$$\phi_c = \tan^{-1} \left[\frac{2V^2}{g\hat{L}_1} \sin \left\{ \sin^{-1} \left(\frac{y}{\hat{L}_1} \right) + \psi_E \right\} \right] + k_2 \int \hat{y} dt,$$
(22)

where the integral gain k_2 is tuned by simulation and

$$\widehat{y} = \begin{cases} y, & \text{if } y \le y_{\text{th}}, \\ 0, & \text{if } y > y_{\text{th}}. \end{cases}$$
(23)

5. Robust Multivariable Controller Design

5.1. The Plant Model. The lateral-directional control system is a multivariable control system as shown in Figure 6. The roll and pitch angles are sensed by the vertical gyro which has very fast dynamics as compared to the vehicle, and so this sensor is modelled as a simple gain in the feedback loop. The magnetic sensor (which senses the heading angle) dynamics are also neglected. The model of the UAV for controller design is taken as a linear approximation obtained at a cruise altitude of 2000 m and a speed of 42 m/sec. However, the flight envelope consists of an altitude range from 10 to 5000 m and a speed bracket from 35 to 60 m/sec. A number of linear models are available across the flight envelope to test the robustness of the control system at different operating conditions. Note that the plant has 2 inputs and 2 outputs. The inputs are the aileron and rudder deflection commands to the actuators. The outputs are the roll angle and the yaw-rate of the vehicle. The computation of the yaw-rate and its filtering is discussed below, but first, we look at computation of the heading angle.

5.1.1. Computation of the Heading Angle. The magnetic sensor installed on-board the vehicle measures the three components of the magnetic field of the earth in the body axes. This magnetic field vector is resolved into a local level frame using the pitch and roll measurements from the vertical gyro. We define the local level axis system $(X_IY_IZ_I)$ in which the Z_I axis is upwards along the local vertical and the X_I and Y_I axes are in the local horizontal plane, with the Y_I axis directed along the projection of the longitudinal axis of the vehicle in the local level plane. The pitch and roll rotations θ and ϕ occur about

the X_l and Y'_l axes (the Y'_l axis is obtained by rotating the Y_l axis by the pitch angle θ), respectively. The transformation matrix from the local level to the body frame is given by

$$T_{lb} = \begin{bmatrix} \cos\phi & \sin\phi\sin\theta & -\sin\phi\cos\theta\\ 0 & \cos\theta & \sin\theta\\ \sin\phi & -\sin\theta\cos\phi & \cos\phi\cos\theta \end{bmatrix}.$$
 (24)

The magnetic field vector is transformed from the body axes to the local level axes by $M_l = T_{lb}^T M_b$, where M_b and M_l represent the magnetic field vector in the body and local level axes, respectively. The magnetic heading is now determined as $\tan^{-1}(M_{lx}/M_{ly})$, where M_{lx} and M_{ly} are the first two components of M_l . Note that this gives the magnetic heading of the vehicle, the true heading ψ (w.r.t. the true north) can be found by applying the declination correction to the magnetic heading [10].

5.1.2. Yaw-Rate Computation. Feedback of the yaw-rate is considered useful for providing damping to the Dutch-roll mode of the vehicle. The vertical gyro however does not provide measurement of angular rates; it only measures roll and pitch angles. The magnetic heading sensor measures the heading angle. Body angular rates are computed from attitude angles as follows.

We define a navigation axis system $(X_nY_nZ_n)$ in which the Z_n axis is upwards along the local vertical, and the X_n and Y_n axes are in the local horizontal plane directed eastward and northward, respectively. The (true) heading, pitch, and roll angles are denoted by ψ , θ , and ϕ , respectively, and these rotations occur in this specific order to align the navigation axes with the body axes. The heading rotation occurs first about the Z_n axis, followed by the pitch rotation about the X'_n axis, finally followed by the roll rotation about the Y''_n axis (note that the primed axes are the intermediate axes obtained while going from the navigation to the body frame through the ψ , θ , and ϕ into body axes, we can solve for the body axes rates. Denoting the roll, pitch, and yaw rates in the body axes by P, Q, and R, respectively, we have

$$P = \dot{\psi}\sin\theta + \dot{\phi},$$

$$Q = \dot{\theta}\cos\phi - \dot{\psi}\cos\theta\sin\phi,$$

$$R = \dot{\theta}\sin\phi + \dot{\psi}\cos\theta\cos\phi.$$
(25)

It may be noted that the attitude sensors measure the body angles and not their rates. The derivatives of the attitude angles are computed by fitting a least squares line to *n* consecutive attitude measurements. We have chosen n = 4 here as we have seen it to yield good results; the derivative, thus, computed has enough noise smoothing and an acceptably small time delay. However, other choices for *n* can be made depending on the application and on the sampling time of the attitude sensor measurements. If, for example, the pitch angle measurements from the vertical gyro are denoted by θ_0 , θ_1 , θ_2 , and θ_3 , where θ_0 corresponds to the current measurement and θ_3 corresponds to the measurement taken three samples





FIGURE 9: Measured and computed yaw rates during test flight.

FIGURE 8: Delay introduced because of the 4-point derivative approximation.

previously, then the slope of the least squares line for these points approximates the derivative [11]:

$$\dot{\theta} \approx \frac{3\theta_0 + \theta_1 - \theta_2 - 3\theta_3}{10\Delta t}.$$
(26)

Here, Δt is the time interval at which the angle θ is sampled (20 msec for our case). This approximation of the derivative by slope of a 4-point least squares line gives a good compromise between sensitivity to measurement noise and the delay introduced into the estimation of $\dot{\theta}$. Figure 8 shows the delay between the actual and computed derivatives for a pure sinusoid of frequency 10 rad/sec, which is about 25 msec. This method of estimating body rates and using them for feedback control is demonstrated to work well for normal flight maneuvers. A test flight was carried out in which a rate sensor was specially installed on the vehicle for validation purposes only; this sensor is not available for feedback control otherwise. The comparison between the measured and computed yaw-rates shown in Figure 9 indicates the validity of the approximation. It is, thus, concluded that the approximation (26) can be used for Euler angle rate estimation from attitude measurements, and thereafter (25) can be employed for transforming the Euler angle rates into body-axes angular rates which are used for feedback control.

5.1.3. The Washout Filter. Yaw-rate feedback is typically useful during transients when the yaw-rate fluctuates or exhibits appreciable changes with time. The rudder is normally employed to suppress the yaw-rate oscillations to provide damping to the Dutch-roll mode of the vehicle. Thus, during steady turns when there is a nearly constant yaw-rate, it is not desirable for the rudder to act to oppose the turning motion. The washout filter is a high-pass filter that is used to feedback the yaw-rate at higher frequencies, while effectively breaking the feedback at dc to enable steady turning of the vehicle. The washout filter is chosen as $W_w = s/(s + 1)$; it may be thought of as a frequency-dependent weight on the yaw-rate feedback, with a small gain at low frequencies and unity gain at high frequencies. The 3 dB point occurs at 1 rad/sec, so that frequencies above 1 rad/sec will be available for feedback.

5.2. H_{∞} Loop-Shaping Design Procedure. For controller design, we shall use the H_{∞} loop-shaping design procedure proposed in [12]. The procedure is intuitive in that it is based on the multivariable generalization of classical loop-shaping ideas. The open-loop plant, once given the desired loop-shape, is robustly stabilized against coprime factor uncertainty. The resulting controller has been shown to enjoy some favourable properties, such as no pole-zero cancellation occurs in the closed-loop system (except for a certain special class of plants), see [13]. In addition, the controllers, thus, designed have been successful in various applications; examples are those described in [14–17].

In practical design applications, the performance specifications are first translated into the frequency domain, and the open-loop plant's singular value frequency response is given the desired shape. This is achieved by augmentation of the nominal plant model *G* by pre- and/or postcompensators (or weighting functions) W_1 and W_2 , respectively. The shaped plant $G_s = W_2 G W_1$ is then robustly stabilized against coprime factor uncertainty, and the controller *K*, thus, obtained is cascaded with the weights to obtain the final controller $W_1 K W_2$. It can be shown that the controller does not significantly alter the specified loop-shape provided a sufficiently small value of the cost γ is achieved; for details, refer to [17].

We now outline a design procedure for designing robust controllers based on open-loop shaping and robust stabilization of the normalized coprime factors of the plant. The procedure consists of the following main steps.

- Plot the singular value frequency response of the open-loop plant G(s). Based on this, select a precompensator W₁ and/or a postcompensator W₂ to give the plant a desired open-loop shape. Form the product W₂(s)G(s)W₁(s).
- (2) Align the singular values of $W_2 G W_1$ at the desired bandwidth. The align gain K_a is the approximate real inverse of the system at the specified frequency. The cross-over (and hence the bandwidth) is thus adjusted to approximately the align frequency. An additional constant diagonal matrix K_q may sometimes be used in front of the align gain to exercise control over actuator usage. It is chosen so that the various actuator rate limits are not exceeded whilst following references or rejecting disturbances. The precompensator can now be written as $W_1 = \widehat{W}_1 K_a K_q$; see Figure 10. Build the shaped plant $G_s(s) = W_2(s)G(s)W_1(s)$, and calculate the optimal cost γ_{opt} [12]. A high value (typically > 10) of γ_{opt} indicates that the specified loop-shape is inconsistent with robust stability; in such a case, the weights W_1 and W_2 should be modified.
- (3) Use a slightly suboptimal value of γ, and compute the corresponding controller.
- (4) Cascade the controller with the weights W₁ and W₂. Controller order reduction may be performed if desired.
- (5) Form the closed-loop, and check the appropriate performance and robustness measures against the given specification.

5.3. Controller Design. Here, we will describe the design of the lateral controller, using the design procedure listed in Section 5.2. The block diagram of the closed-loop system is shown in Figure 11.

(1) Singular values of the open-loop plant G(s) are plotted, and these indicate the need for boosting the low frequency gain for good tracking and disturbance rejection for the roll control channel at these frequencies. The low frequency gain is boosted by introducing integral action in the roll control loop; we choose to place a pole at s = -0.01 instead of using a pure integrator (because of the wind-up problems associated with integrators, and also because we do not want to introduce an unstable pole into the shaped plant G_s). \widehat{W}_1 in Figure 10 is chosen as $\begin{bmatrix} 28(s+2.5)/(s+0.01) & 0 \\ 0 & 40 \end{bmatrix} \text{ and } W_2 \text{ as } \begin{bmatrix} 14(s+2)/(s+70) & 0 \\ 0 & 50/(s+50) \end{bmatrix}.$ Since we are aiming for a closed-loop bandwidth of approximately 10-15 rad/sec, therefore zeros are introduced at -2.5 and -2 in \widehat{W}_1 and W_2 to reduce the roll-off at the cross-over frequencies. The poles at -70 and -50 in W_2 are placed to provide adequate roll-off at higher frequencies. Since \widehat{W}_{11} (the upper left element of \widehat{W}_1) approximates an integrator, it will be implemented in its conditioned form (see [18, 19]). Gains are selected in \widehat{W}_1 and W_2 to adjust the crossover frequency to around 10 rad/sec.



FIGURE 10: The shaped plant and the controller.

- (2) The align matrix K_a is chosen to be the identity matrix for robustness reasons (aligning the nominal plant and designing the controller based on this may not hold good with other models of the plant across the flight envelope) and since both singular values are close enough at cross-over. The matrix K_g is also kept as the identity matrix I_2 . The precompensator is therefore $W_1 = \widehat{W}_1 K_a K_g = \widehat{W}_1$. The shaped plant $G_s =$ $W_2 G W_1$ is now formed, and its singular values are shown in Figure 12. The low frequency gain in the roll channel is boosted, as indicated by the larger singular value. The washout filter in the yaw-rate channel means that the low frequency gain in this channel remains low, as indicated by the smaller singular value of the shaped plant.
- (3) An optimal γ of 1.925 is computed; the controller is obtained for a slightly suboptimal γ of 2.02.
- (4) The controller is reduced using optimal Hankel-norm approximation and cascaded with the weights W_1 and W_2 to form the complete controller W_1KW_2 . This reduced controller is discretized using the bilinear (Tustin's) approximation for implementation.

The singular values of the sensitivity function are shown in Figure 13. One of the singular values, the one that remains flat at low frequencies, corresponds to yaw-rate and indicates no disturbance rejection on that output. The other singular value shows adequate disturbance rejection on the roll angle channel for frequencies less than 5 rad/sec. A number of linearized models are taken along the periphery of the flight envelope. Step responses for these plant models in the envelope using the designed controller are given in Figure 14. The controller performs acceptably well throughout the envelope, and the robustness of the design is illustrated.

5.4. Robustness Analysis. A more formal robustness analysis of the controller is now performed. Note that the controller design is based on a normalized coprime factor representation of the plant: $G_s = W_2 G W_1 = \widetilde{M}^{-1} \widetilde{N}$, where the transfer functions \widetilde{M} and \widetilde{N} represent the normalized coprime factors of the (shaped) plant. The controller is designed to maximize robust stability in the face of coprime factor perturbations. The perturbations (or uncertainties) in the plant are represented as perturbations on the normalized coprime factors of the plant. The perturbations on the normalized coprime factors of the plant. The perturbations of the normalized coprime factors of the plant.

$$G_{s_{\Delta}} = \left(\widetilde{M} + \Delta_{\widetilde{M}}\right)^{-1} \left(\widetilde{N} + \Delta_{\widetilde{N}}\right), \tag{27}$$

where $\Delta_{\widetilde{M}}$, $\Delta_{\widetilde{N}}$ are stable unknown transfer functions representing uncertainty in the plant model. Robust stability to



FIGURE 11: Plant, controller, and weighting functions.



FIGURE 12: Singular values of the original and shaped plants.



FIGURE 13: Sensitivity function $(I + GW_1KW_2)^{-1}$.



FIGURE 14: Step responses for different plant models across the flight envelope.



FIGURE 15: Singular values of $(\Delta_{\widetilde{M}} \ \Delta_{\widetilde{N}})$ for different plants in the envelope.



FIGURE 16: Part of Mission 1 and the path flown by the vehicle.



FIGURE 17: The roll angle and the roll angle error.

coprime factor perturbations is maximized by a controller *K* which stabilizes the nominal plant *G*_s and minimizes

$$\gamma = \left\| \begin{bmatrix} K \\ I \end{bmatrix} \left(I - G_s K \right)^{-1} \widetilde{M}^{-1} \right\|_{\infty}.$$
 (28)

The perturbed closed-loop system is then guaranteed to remain stable in the face of all $\Delta_{\widetilde{M}}$, $\Delta_{\widetilde{N}}$ such that [20]

$$\left\| \begin{bmatrix} \Delta_{\widetilde{M}} & \Delta_{\widetilde{N}} \end{bmatrix} \right\|_{\infty} < \gamma^{-1}.$$
 (29)

Robust stability across the flight envelope is tested by obtaining normalized coprime factorizations of all plants in the



FIGURE 18: The actuator (aileron and rudder) deflections.



FIGURE 19: Cross-track deviation of the vehicle.

envelope and then taking their differences from the nominal plant to get uncertainty transfer functions $[\Delta_{\widetilde{M}_i} \ \Delta_{\widetilde{N}_i}]$, where the subscript *i* refers to the *i*"th" plant in the envelope. Singular values of these uncertainty transfer functions are plotted in Figure 15; the maximum of these across all frequencies equals 0.42. The cost γ for our controller is 2.02, which indicates that (29) is satisfied, thus ensuring robust stability for the family of plants in the envelope; that is,

$$\left\| \left[\Delta_{\widetilde{M}_i} \ \Delta_{\widetilde{N}_i} \right] \right\|_{\infty} = 0.42 < 2.02^{-1}.$$
(30)

It is seen that a single controller is robust enough to stabilize all plants in the flight envelope; also robust performance is



FIGURE 20: The turn performance.

checked for and is found to be adequate by performing step response analysis of the family of plants (Figure 14).

6. Flight Test Results

The complete guidance and control scheme is implemented on a real-time embedded computer and flown on the actual UAV for various missions. Part of the two typical missions and their flight results are presented here. Figure 16 shows part of the first mission in which waypoints are indicated by small squares, and the actual path flown by the vehicle is shown in solid line. The thick portions of the line indicate that the vehicle is in a state of turn; that is, the turn start criterion is satisfied (see Section 3.2). As soon as the turn stop criterion is met, the line thickness is changed back to normal. The figure shows very good path following by the vehicle during flight. Figure 17 shows the performance of the multivariable roll/yaw-rate controller during flight. In the top part of the figure the commanded and actual roll angles are plotted in dotted and solid lines, respectively. The two lines are almost on top of one another, thus demonstrating the tracking performance of the roll controller. The lower part of the figure shows the roll error which peaks to a maximum value of around 3 degrees at times when there is a sudden change in the roll reference; the error in the steady-state is almost zero. Figure 18 shows the actuator (aileron and rudder) deflections



FIGURE 21: Part of Mission 2 and the path flown by the vehicle.



FIGURE 22: The roll angle and the roll angle error for Mission 2.

recorded during the same flight. The figure shows a 100 second zoomed view of the flight data. The deflections are small, and the aileron is deflected to about 5 degrees, whereas the rudder deflection stays within 2 degrees during a turn. The multivariable controller engages both actuators to provide good roll angle tracking and stability augmentation. Figure 19 shows the cross-track deviation of the vehicle from the desired path and thus illustrates the tracking performance of the guidance and control system. The performance is excellent with the off-track deviation controlled to within a few meters. The larger 10–20 m errors arise only when a turn is initiated or when one ends, since the reference path



FIGURE 23: Cross-track deviation of the vehicle for Mission 2.

to be followed suddenly changes at these instants. Figure 20 shows a zoomed view of different variables of interest related to the turn performance of the vehicle. The bank-to-turn scheme is illustrated as the rolling of the vehicle causes the heading angle to change continuously. The yaw-rate of the vehicle is also shown (which is fed back to the controller through the washout filter). The washout filter blocks the yaw-rate feedback during steady turns and only feeds through the higher frequency transients for stabilization of the Dutch-roll mode of the vehicle.

Figure 21 shows part of the second mission with the waypoints indicated by small boxes and also the actual path flown by the vehicle. The line width is increased to indicate turnings. The top part of Figure 22 shows the commanded and actual roll angles, the commanded angle being generated by the guidance (22). The roll angle tracking provided by the multivariable controller is quite good. The bottom half of the figure shows the roll error which again is seen to be very small and peaks only during transients. The cross-track deviation is shown in Figure 23.

7. Conclusion

A practical methodology for the design of lateral-directional guidance and control systems for high performance UAVs is presented. The methodology has been developed keeping in view the practical aspects of the problem and implementation on a real-time platform. Flight path formulae are derived both for straight and turning flight. A nonlinear lateral guidance law is presented that uses the angular error and the crosstrack deviation of the vehicle and generates roll angle commands to guide the vehicle back onto the desired path. An adaptive adjustment in the basic law is suggested to cater for large deviations from the nominal track as may arise in case of GPS outages. A robust multivariable controller is designed for the lateral-directional control of the vehicle. The controller uses measurements of the roll angle and yaw-rate of the vehicle and optimally actuates the ailerons and the rudder to provide roll control and Dutch-roll damping. A practical method of computing the heading angle from the magnetometer and then estimating the yaw-rate is presented. The robustness analysis of the controller is performed, and it is seen to perform well throughout the flight envelope of the vehicle. Finally, the flight test results are presented and discussed; these demonstrate the overall performance of the guidance and control system.

References

- M. Niculescu, "Lateral track control law for Aerosonde UAV," in *Proceedings of the 39th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nev, USA, January 2001.
- [2] T. Keviczky and G. J. Balas, "Software-enabled receding horizon control for autonomous UAV guidance," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pp. 527– 552, San Francisco, Calif, USA, August 2005.
- [3] B. Mettler, M. Valenti, T. Schouwenaars et al., "Autonomous UAV guidance build-up: flight-test demonstration and evaluation plan," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Austin, Tex, USA, August 2003.
- [4] T. Schouwenaars, M. Valenti, E. Feron, and J. How, "Implementation and flight test results of MILP-based UAV guidance," in *Proceedings of the IEEE Aerospace Conference*, March 2005.
- [5] S. R. Griffiths, "Vector field approach for curved path following for miniature aerial vehicles," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pp. 3375–3389, Keystone, Colo, USA, August 2006.
- [6] W. J. Pisano, D. A. Lawrence, and P. C. Gray, "Autonomous UAV control using a 3-sensor autopilot," in *Proceedings of the AIAA InfoTech at Aerospace Conference*, pp. 423–436, Rohnert Park, Calif, USA, May 2007.
- [7] S. Park, J. Deyst, and J. P. How, "A new nonlinear guidance logic for trajectory tracking," in *Proceedings of the AIAA Guidance*, *Navigation, and Control Conference*, pp. 941–956, Providence, RI, USA, August 2004.
- [8] R. Samar, S. Ahmed, and F. Aftab, "Lateral control with improved performance for UAVs," in *Proceedings of the 17th IFAC Symposium on Automatic Control in Aerospace (ACA '07)*, pp. 37–42, Toulouse, France, June 2007.
- [9] R. Samar, S. Ahmed, and M. Nzar, "Lateral guidance & control design for an unmanned aerial vehicle," in *Proceedings of the* 17th World Congress, International Federation of Automatic Control (IFAC '08), Seoul, South Korea, July 2008.
- [10] J. M. Quinn, R. J. Coleman, M. R. Peck, and S. E. Lauber, "The joint US/UK, 1990 epoch world magnetic model," Tech. Rep. 304, Naval Oceanographic Office, October 1991.
- J. H. Mathews, Numerical Methods for Mathematics, Science, and Engineering, Prentice-Hall, Englewood Cliffs, NJ, USA, 2nd edition, 1992.
- [12] D. McFarlane and K. Glover, "A loop-shaping design procedure using H∞ synthesis," *IEEE Transactions on Automatic Control*, vol. 37, no. 6, pp. 759–769, 1992.
- [13] M. C. Tsai, E. J. M. Geddes, and I. Postlethwaite, "Pole-zero cancellations and closed-loop properties of an H∞ mixed sensitivity design problem," *Automatica*, vol. 28, no. 3, pp. 519– 530, 1992.
- [14] R. Samar, G. Murad, I. Postlethwaite, and D. W. Gu, "A discrete time H∞ observer-based controller & its application to a glass

tube production process," *European Journal of Control*, vol. 2, no. 2, pp. 112–125, 1996.

- [15] A. J. Smerlas, D. J. Walker, I. Postlethwaite, M. E. Strange, J. Howitt, and A. W. Gubbels, "Evaluating H₀₀ controllers on the NRC Bell 205 fly-by-wire helicopter," *Control Engineering Practice*, vol. 9, no. 1, pp. 1–10, 2001.
- [16] I. Postlethwaite, E. Prempain, E. Turkoglu, M. C. Turner, K. Ellis, and A. W. Gubbels, "Design and flight testing of various H∞ controllers for the Bell 205 helicopter," *Control Engineering Practice*, vol. 13, no. 3, pp. 383–398, 2005.
- [17] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control Analysis and Design*, John Wiley & Sons, West Sussex, UK, 2nd edition, 2005.
- [18] R. Hanus, M. Kinnaert, and J. L. Henrotte, "Conditioning technique, a general anti-windup and bumpless transfer method," *Automatica*, vol. 23, no. 6, pp. 729–739, 1987.
- [19] R. Hanus and M. Kinnaert, "Control of constrained multivariable systems using the conditioning technique," in *Proceedings* of the American Control Conference, pp. 1712–1718, Pittsburgh, Penn, USA, June 1989.
- [20] D. C. McFarlane and K. Glover, Robust Controller Design Using Normalized Coprime Factor Plant Descriptions, vol. 138 of Lecture Notes in Control and Information Sciences, Springer, Berlin, Germany, 1990.

