A RECONFIGURABLE NONLINEAR AUTOPILOT

Marc Bodson[*]

University of Utah

Abstract

The paper proposes a reconfigurable flight control system for the tracking of altitude, heading, sideslip, and velocity commands. The control law may serve for the command of unmanned air vehicles or as an autopilot for piloted aircraft. The inner core of the algorithm consists of a reconfigurable control system providing tracking of pitch, roll, and yaw rate commands. It is based on a model reference control law and a stabilized recursive least-squares algorithm. The outer loop is based on a linear design, with compensation for the nonlinear couplings arising from flight dynamics. Some parameters of the outer loop are identified in real-time, in order to adapt to varying flight conditions. The algorithm is evaluated using a nonlinear F-16 simulation model. The results demonstrate the consistent performance of the algorithm through various flight conditions, as well as its turn coordination capabilities, its reconfiguration after a floating left elevator failure, its ability to move across the power curve, and its tolerance to measurement noise and turbulence.

[*] Professor, Department of Electrical & Computer Engineering, University of Utah , 50 S Central Campus Dr Rm 3280, Salt Lake City, UT 84112, U.S.A.

Introduction

The development of adaptive flight control systems has attracted significant interest recently. Adaptive systems are able to reconfigure after damages and failures, providing benefits in terms of safety. Lower development costs may also result from their self-designing capabilities. Recent articles include those reporting on flight tests of the VISTA F-16 [1], on the results of the RESTORE program [2], [3], on simulations of the X-33 hypersonic vehicle [4], and on other studies of a variety of reconfigurable control algorithms [5], [6], [7], [8], [9], [10], [11]. Most of the available algorithms have been designed to replace conventional flight control systems for piloted aircraft. The command variables are angular rates, or blends of angular rates and aerodynamic angles. There have been limited analyses of reconfigurable control laws for flight path variables. For example, algorithms for tracking of angular positions are discussed in [12], [13]. Techniques that enable a higher level of command, such as trajectory tracking or tracking of a velocity vector reference, would be useful for the implementation of cooperative control laws for unmanned air vehicles (UAV) (see, *e.g.,* [14], [15]), or as autopilots for piloted aircraft.

Traditionally, autopilots have been designed using linearized models and gain-scheduling ([16], pp. 358-418). A challenge in developing a reconfigurable autopilot is to obtain a control law that depends only on parameters that can be identified reliably in real-time. One must also be able to translate the control design into an automatic computational process that, also, can be performed reliably and in real-time. Finally, there is an advantage in obtaining an autopilot that exploits the reconfigurable flight control systems developed for piloted aircraft.

The objective of this paper is to propose an autopilot that provides tracking of altitude, heading, sideslip, and velocity commands. The control algorithm adjusts its parameters automatically throughout the flight envelope, and is able to reconfigure after failures without explicit failure detection or identification. Prior information is needed in the design, but only in the form of desired closed-loop dynamics of the system. Although the desired dynamics are specified as linear dynamics, the autopilot exploits knowledge of the nonlinear flight dynamics to achieve the desired result.

The basic structure of the algorithm is shown in Fig. 1. The inner core is a reconfigurable flight control system based on a model reference control law for tracking of pitch, roll, and yaw rate commands, with a stabilized recursive least-squares algorithm for adaptation. The outer loop provides tracking of altitude, heading, sideslip, and velocity commands. It is mostly a linear regulator, but it includes compensation of known nonlinear dynamic couplings. Automatic adjustment of some parameters of the nonlinear autopilot is obtained using estimated parameters from the inner loop.

The paper demonstrates the satisfactory performance of the overall adaptive system in simulations, A detailed F-16 model is used, including 6-DOF dynamics, nonlinear aerodynamic tables, first-order actuator models with position and rate limits, and an engine thrust model. Measurement noise and turbulence are also included in one simulation to demonstrate the robustness of the algorithm.

Reconfigurable Flight Control System

This section describes the inner loop of the reconfigurable autopilot. This inner loop is designed as a reconfigurable flight control system for piloted aircraft, and provides tracking of angular rate commands. The algorithm is based on model reference adaptive control theory, and evolved from previous work reported in [1], [12], [17], and [18].

Model Reference Control Law

Consider the state-space representation

$$\begin{aligned}\dot{x} &= Ax + Bu + d \\ y &= Cx\end{aligned} \tag{1}$$

where x, d , u, and y are vectors. The disturbance d is assumed to be constant, and may originate from the linearization of a nonlinear system around an unknown operating point. Model reference control laws, sometimes referred to as dynamic inversion control laws, rely on a reference model that represents the desired dynamics of the closed-loop system. For example, one may choose

$$\dot{y}_M = A_M y_M + B_M r_M \tag{2}$$

where r_M is a reference input vector and y_M represents the desired output of the system. Since the derivative of y is given by

$$\dot{y} = CAx + CBu + Cd \tag{3}$$

the objective may be achieved by setting

$$u = (CB)^{-1}\left(-CAx - Cd + A_M y + B_M r_M\right) \tag{4}$$

Model matching follows if the matrix CB is square and invertible, and if the original system is minimum phase. If the matrix CB is not square but has full row rank, the same model reference control law can be used if one defines

$$a_d = -CAx - Cd + A_M y + B_M r_M \tag{5}$$

and the control input u is chosen such that $(CB)u = a_d$.

For the flight control application of this paper, we choose

$$x = \begin{pmatrix} \alpha \\ q \\ \beta \\ p \\ r \end{pmatrix}, \quad y = \begin{pmatrix} q \\ p \\ r \end{pmatrix}, \quad u = \begin{pmatrix} \delta_e \\ \delta_a \\ \delta_r \end{pmatrix} \tag{6}$$

where α is the angle of attack (in deg), q is the pitch rate (in deg/s), β is the angle of sideslip (in deg), p is the roll rate (in deg/s), r is the yaw rate (in deg/s), δ_e is a symmetric elevator command (in deg), δ_a is an antisymmetric aileron command (in deg), and δ_r is the rudder command (in deg). The techniques of this paper apply to a higher number of actuators (in particular, unganged actuators), provided that a control allocation algorithm replaces the matrix inverse of (4).

Given the choice of output vector y_M, the vector r_M is interpreted as a vector of desired rotational rates

$$r_M = \begin{pmatrix} q_{com} \\ p_{com} \\ r_{com} \end{pmatrix} \tag{7}$$

The reference model is chosen to be

$$A_M = -k \cdot I_{3\times3}, B_M = k \cdot I_{3\times3} \tag{8}$$

so that the ideal responses to pitch, roll, and yaw rate commands are those of independent first-order systems with poles at $s = -k$.

The choice of rotational rates as outputs is generally considered to be adequate for piloted systems at low speeds. For higher speeds, a blend of pitch rate and angle of attack is often taken to replace pitch rate. The velocity axis roll rate may be chosen to replace the (body axis) roll rate, and a blend of yaw rate and sideslip angle may be chosen to replace the yaw rate [6]. These changes may be accommodated through linearization and a different choice of matrix C. The algorithm proposed in this paper does not make such a choice, relying instead on the outer loop of the nonlinear autopilot to provide regulation of the angle of attack and angle of sideslip. While a minimum phase condition is required in theory to ensure boundedness of the internal variables (α and β), this boundedness is enforced by the outer loop of the autopilot, so that the condition is actually not required for the scheme proposed in this paper.

Real-Time Parameter Identification

To make the model reference control system reconfigurable, it is necessary to determine the coefficients of the matrices A, B, and d in (1). For example, the second equation contains nine parameters, as indicated by

$$\dot{q} = A_{21}\alpha + A_{22}q + A_{23}\beta + A_{24}p + A_{25}r + B_{21}\delta_e + B_{22}\delta_a + B_{23}\delta_r + d_2 \tag{9}$$

Note that this equation may be written as a vector product

$$y_w = \theta^{*T} w \tag{10}$$

where θ^* is a vector containing the unknown parameters A_{21}, A_{22}, ..., B_{23}, and d_2, and w is a vector containing the signals α, q,..., δ_r, and a signal equal to 1 for all time. The vector w is usually called the *regressor vector*. For real-time identification, measurements of $y_w(k)$ and $w(k)$ are taken at time instants $k = 1,...,n$, and the problem is to find a parameter estimate $\theta(n)$ satisfying the linear relationship (10). Because of measurement noise, and because the model is not an exact representation of the aircraft, an estimate must be found such that the best fit is obtained, using a sufficiently large number of data points.

In the context of flight dynamics, the following considerations are worth noting:

- For the first and third equations of (1), the normal and lateral accelerometer signals at the center of gravity (a_n and a_y) may be used instead of $\dot{\alpha}$ and $\dot{\beta}$. The advantage is that the accelerometer signals are more reliable and less noisy, and that the signals are not perturbed by effects that make direct identification of the $\dot{\alpha}$ and $\dot{\beta}$ equations more difficult. These effects include longitudinal/lateral couplings, gravity effects, and turbulence.

- The derivatives $\dot{q}$, $\dot{p}$, and $\dot{r}$ may be obtained from accelerometer measurements taken at separate locations on the aircraft [1]. Another approach consists in differentiating the signals q, p, and r. Filtering of the variables is necessary to reduce the effect of the noise, and it is desirable to filter all the variables by the same filters, so that the data remains aligned.
- The parameters of the state-space model are strongly affected by the dynamic pressure $\overline{q}$ (in lb/ft^2) and by the speed v (in ft/s). The main form of the dependency is known from aerodynamic theory. Therefore, it is helpful to define modified parameters, which are the parameters of the state-space model divided by $\overline{q}$, except for those associated with q, p, and r, which are divided by $\overline{q}/v$. The modified parameters are similar to the non-dimensional stability derivatives, with the main difference being the multiplicative factors involving the aircraft mass, inertias, mean aerodynamic chord, wing span, and wing area. The $\dot{\alpha}$ and $\dot{\beta}$ derivatives are also merged with other derivatives (mostly the q and r derivatives), and the derivatives are body axis derivatives as opposed to the usual stability axis derivatives.
- Eliminating parameters known to be small improves the robustness of the algorithm. Off-line identification, combined with a sensitivity analysis, may be used to determine which parameters do not affect the responses in a significant way.

With these considerations (and with a minor reordering of the variables), the parameters that were identified in the simulations of this paper are the coefficients that yield the best fit of the following relationships

$$
\begin{aligned}
&y_{w,1} = a_n = \theta_{11}\bar{q}\alpha + \theta_{16}\bar{q} \\
&y_{w,2} = \dot{q} = \theta_{21}\bar{q}\alpha + \theta_{22}\frac{\bar{q}}{v}q + \theta_{26}\bar{q} + \theta_{27}\delta_e \\
&y_{w,3} = a_y = \theta_{33}\bar{q}\beta + \theta_{34}\frac{\bar{q}}{v}p + \theta_{35}\frac{\bar{q}}{v}r + \theta_{36}\bar{q} + \theta_{38}\delta_a + \theta_{39}\delta_r \\
&y_{w,4} = \dot{p} = \theta_{43}\bar{q}\beta + \theta_{44}\frac{\bar{q}}{v}p + \theta_{45}\frac{\bar{q}}{v}r + \theta_{46}\bar{q} + \theta_{47}\delta_e + \theta_{48}\delta_a + \theta_{49}\delta_r \\
&y_{w,5} = \dot{r} = \theta_{53}\bar{q}\beta + \theta_{54}\frac{\bar{q}}{v}p + \theta_{55}\frac{\bar{q}}{v}r + \theta_{56}\bar{q} + \theta_{57}\delta_e + \theta_{58}\delta_a + \theta_{59}\delta_r
\end{aligned}
\tag{11}
$$

where a_n and a_y are expressed in g's. The derivation of the state-space model from (11) is direct, except for the $\dot{\alpha}$ and $\dot{\beta}$ equations. Assuming a linearization around zero state, the following equations

$$\dot{\alpha} = \frac{180}{\pi}\left(\frac{(-a_n+1)g}{v}\right) + q, \quad \dot{\beta} = \frac{180}{\pi}\left(\frac{a_y g}{v}\right) - r \tag{12}$$

lead to a state-space model, where g is the acceleration of gravity (in ft/s^2). When the output vector is chosen to be $y = (q, p, r)$, the parameters of the $\dot{\alpha}$ and $\dot{\beta}$ equations of the state-space model are *not* needed for the computation of the model reference control law, so that this transformation is not needed, but the parameters of the first equation of (11) are nevertheless useful for the nonlinear autopilot to be discussed later.

Stabilized Recursive Least-Squares Algorithm

Given the linear parameterization obtained for identification, a recursive least-squares algorithm may be used to find the vector $\theta(n)$ that best fits the linear relationship (10), given measurements of $y_w(k)$ and $w(k)$ for $k = 1, ..., n$. The algorithm may then be applied to determine the parameters of the 5 equations of (11). We consider the optimization criterion

$$J(\theta(n)) = \sum_{k=1}^{n} \lambda^{n-k} (y_w(k) - \theta^T(n)w(k))^2 + \alpha_w \left\|\theta(n) - \theta(n-1)\right\|^2 \tag{13}$$

The criterion contains the weighted sum of the squares of the errors over the window of n data points. The weighting factor λ, called the *forgetting factor*, emphasizes the importance of the latest measurements. If $\lambda = 1$, all the data points are weighted equally. If λ is close to 0, only the last data points contribute to the criterion J in a significant way. The choice of λ requires a trade-off between the long term memory of the algorithm and its ability to adapt rapidly after changes. The second term in the optimization criterion limits the deviation of the estimate from the previous estimate. It is necessary in order to guarantee the stability of the algorithm during periods of insufficient excitation, *e.g.,* steady flight. Without this term, the algorithm (which is the so-called least-squares algorithm with forgetting factor) may be sensitive to noise.

The batch form of the solution to (13) has been referred to in the literature as *the modified sequential least-squares algorithm*. It was used for parameter identification of the VISTA F-16 in [1]. Here, we use the recursive formulation of [17] with the update law

$$\begin{aligned} &P^{-1}(n) = \lambda P^{-1}(n-1) + w(n)w^{T}(n) + \alpha_w(1-\lambda)I \\ &\theta(n) = \theta(n-1) + P(n)w(n)(y(n) - w^{T}(n)\theta(n-1) + \alpha_w \lambda P(n)(\theta(n-1) - \theta(n-2)) \end{aligned} \qquad (14)$$

where $P(n)$ is an $n_p \times n_p$ matrix, with $n_p = \dim(w) = \dim(\theta)$. Initially, $P^{-1}(0) = \alpha_w I$ and $\theta(0)$ is arbitrary. The matrix $P(n)$ is called the covariance matrix. If $\alpha_w = 0$ and $w(n) = 0$, (14) shows that $P^{-1}(n)$ converges to zero, which means that the covariance matrix $P(n)$ goes to infinity (this undesirable property also arises if $w(n)$ is zero in some directions, *i.e.,* if w is not persistently exciting). With $\alpha_w \neq 0$, the covariance matrix update law is stabilized, and one can show that both $P(n)$ and $P^{-1}(n)$ are bounded as long as $w(n)$ is bounded.

In the exact form of the algorithm (14), the inversion of the matrix $P^{-1}(n)$ is required in order to use $P(n)$ in the update law for $\theta(n)$. Computation of this inverse may be avoided by applying the so-called matrix inversion lemma ([19], p. 50) and an approximation of the update law, as discussed in [17]. Specifically, the first equation of (14) is replaced by

$$P(n)=\frac{1}{\lambda}P(n-1)-\frac{1}{\lambda}P(n-1)C(n)\left(\lambda I+C^{T}(n)P(n-1)C(n)\right)^{-1}C(n)^{T}P(n-1) \tag{15}$$

where the $n_p \times 2$ matrix $C(n)$ is given by

$$C(n)=\begin{pmatrix} w(n) & \sqrt{n_p\alpha_w(1-\lambda)}\cdot e(n)\end{pmatrix} \tag{16}$$

and the sequence of $n_p \times 1$ vectors $e(n)$ is given by

$$e(1)=\begin{pmatrix}1\\0\\\vdots\\0\end{pmatrix}, e(2)=\begin{pmatrix}0\\1\\\vdots\\0\end{pmatrix}, \ldots, e(n_p)=\begin{pmatrix}0\\0\\\vdots\\1\end{pmatrix}, e(n_p+1)=\begin{pmatrix}1\\0\\\vdots\\0\end{pmatrix}, \cdots \tag{17}$$

The update law is initialized with $P(0)=(1/\alpha_w)I$. Only a 2×2 matrix inverse is required. The covariance matrix is known to be symmetric, and it is helpful to compute the upper triangular part of the matrix only, and to update the lower triangular part by symmetry (or vice-versa).

Nonlinear Autopilot

The nonlinear autopilot is the outer loop of the reconfigurable nonlinear autopilot. It consists of four separate controllers for altitude, heading, sideslip, and speed. The controllers attempt to achieve linear responses through compensation of the nonlinear dynamic effects and through couplings between the controllers when appropriate.

Altitude Controller

The altitude controller consists of several loops built around the reconfigurable control law discussed in section 2. The structure of the control system is shown in Fig. 2, where every loop is designed assuming that the next inner loop achieves perfect tracking of the commands. The assumption is obviously not satisfied, but this type of approach has been found to work well in practice. Its advantage is that it is possible to impose limits on inner loop variables, such as the angle of attack and the rate of climb. In the figure, the blocks C_1, C_2, C_3 refer to transformations that are linear in the main input variable, but nonlinear in the other variables. These other variables are θ_{11}, θ_{16} (the parameters identified for a_n by the inner reconfigurable control loop), a_n (the signal provided by a normal accelerometer located at the center of gravity), v (the airspeed), θ (the pitch angle in deg), and ϕ (the roll angle in deg).

The inner loop of the altitude controller regulates the angle of attack. From flight dynamics ([20], p. 607), one has that, for small α and β,

$$\dot{\alpha} = -\frac{180}{\pi} \cdot \frac{g}{v} \left(a_n - \cos\theta \cos\phi \right) + q \tag{1}$$

(recall that α is expressed in deg and q in deg/s). Thus, we choose

$$\begin{aligned} q_{com} &= \frac{180}{\pi} \cdot \frac{g}{v} (a_n - \cos\theta \cos\phi) + \dot{\alpha}_{com} \\ \dot{\alpha}_{com} &= g_\alpha \left(\alpha_{com} - \alpha \right) \end{aligned} \tag{2}$$

where α_{com} is the commanded angle of attack. Assuming that q=q_{com}, $\dot{\alpha} = \dot{\alpha}_{com}$ results, and the response of the angle of attack to the command is that of a first-order system with pole at $-g_\alpha$.

The second loop controls the rate of climb, given the command $\dot{h}_{com}$. Recall that the parameter identification algorithm obtained parameters such that

$$a_n = \theta_{11} \bar{q} \alpha + \theta_{16} \bar{q} \tag{3}$$

In order to achieve a desired normal acceleration $a_{n,com}$, the following angle of attack command may therefore be chosen

$$\alpha_{com} = \frac{a_{n,com} - \theta_{16} \bar{q}}{\theta_{11} \bar{q}} \tag{4}$$

Next, from flight dynamics, it is known that the second derivative of altitude is related to the normal accelerometer signal through

$$\ddot{h} = g\left(\cos\theta\cos\phi\; a_n - 1\right) \tag{5}$$

Therefore, we let the variable $a_{n,com}$ be

$$\begin{aligned} a_{n_{com}} &= \frac{1}{\cos\theta\cos\phi}\left(1 + \frac{1}{g}\ddot{h}_{com}\right) \\ \ddot{h}_{com} &= g_{\dot{h}}\left(\dot{h}_{com} - \dot{h}\right) \end{aligned} \tag{6}$$

so that a first-order response will result for $\dot{h}$ under ideal conditions.

Finally, we choose

$$\dot{h}_{com} = g_h\left(h_{com} - h\right) \tag{7}$$

so that, if the angle of attack command is tracked perfectly, the altitude response will satisfy a second-order system description

$$\ddot{h} = g_{\dot{h}}\left(\dot{h}_{com} - \dot{h}\right) = g_{\dot{h}} g_h h_{com} - g_{\dot{h}} g_h h - g_{\dot{h}}\dot{h} \tag{8}$$

whose poles are determined by $s^2 + g_{\dot{h}} s + g_{\dot{h}} g_h = 0$, and may be placed arbitrarily. Although not shown on the figure, it is helpful to insert limits for the angle of attack command and for the altitude rate command (the values of the gains and of the limits used in the simulation are given later in the paper).

Heading Controller

The control law for heading is shown in Fig. 3. It uses the roll rate command in the same manner as a classical autopilot would use the aileron command. The inner loop regulates the roll angle ϕ and the outer loop tracks commands for the heading angle χ. From flight dynamics, one has approximately that

$$\begin{aligned} \dot{\phi} &= p \\ \dot{\chi} &= \frac{g}{v}\phi \end{aligned} \tag{9}$$

Assuming that $p = p_{com}$, the control law

$$\begin{aligned} p_{com} &= g_{\phi}\left(\phi_{com} - \phi\right) \\ \phi_{com} &= g_{\chi}\frac{v}{g}\left(\chi_{com} - \chi\right) \end{aligned} \tag{10}$$

yields

$$\ddot{\chi} = \frac{g}{v}g_{\phi}\left(\phi_{com} - \phi\right) = g_{\phi}g_{\chi}\chi_{com} - g_{\phi}g_{\chi}\chi - g_{\phi}\dot{\chi} \tag{11}$$

The closed-loop poles are given by the roots of $s^2 + g_{\phi}s + g_{\phi}g_{\chi} = 0$ and may be placed arbitrarily by choice of g_{ϕ} and g_{χ}. As for the heading controller, it is helpful to limit the maximum roll command ϕ_{com}. It is also helpful to ensure that the error e_{χ} remains in the range ± 180 degrees.

Sideslip Controller

The sideslip control law is similar to the angle of attack control law of Fig. 2 and is shown in Fig. 4. Here, the relevant equation ([20], p. 611) is

$$\dot{\beta} = \frac{180}{\pi}\cdot\frac{g}{v}(a_y + \cos\theta\sin\phi) + p\sin\alpha - r\cos\alpha \tag{12}$$

From this equation, the yaw rate command is chosen to be

$$\begin{aligned} r_{com} &= p\tan\alpha + \frac{180}{\pi}\cdot\frac{g}{v\cos\alpha}\left(a_y + \cos\theta\sin\phi\right) - \frac{\dot{\beta}_{com}}{\cos\alpha} \\ \dot{\beta}_{com} &= g_{\beta}\left(\beta_{com} - \beta\right) \end{aligned} \tag{13}$$

If $r = r_{com}$, the response of the sideslip controller is that of a first-order system with pole at $-g_{\beta}$. Note that, for $\dot{\beta}_{com} = 0$, the yaw rate command r_{com} is the yaw rate required to execute a coordinated turn.

In a dynamic inversion design, it is typical to control sideslip by specifying a matrix C so that the output signal is a blend of sideslip and yaw rate. Yaw rate must be blended to ensure feasibility of model matching with a first-order system. The design proposed here is an alternative approach. The angle of sideslip is commanded directly, and the scheme exploits the known nonlinear dynamics of the directional variables without linearization.

Speed Control & Total Energy Compensation

A proportional/integral compensator may be used for speed regulation. Although tuning of the controller may be performed through parameter identification using the longitudinal accelerometer signal, and although the design is suited for such real-time adaptation, a fixed controller was judged adequate for the simulations reported in this paper.

The speed autopilot assumes that the following transfer function represents (approximately) the response from τ_{com}, the throttle command (specified as a variable ranging from 0 to 1), to v the speed of the aircraft

$$\frac{v(s)}{\tau_{com}(s)} = \frac{k_v}{s(s+a_v)} \tag{14}$$

where k_v / a_v is the amount of steady-state longitudinal acceleration per unit of throttle command, and $1/a_v$ is the time constant of the engine thrust response. The following proportional/integral controller is selected

$$\tau_{com}(s) = g_{FV} v_{com}(s) - g_{PV} v(s) + \frac{g_{IV}}{s}\left(v_{com}(s) - v(s)\right) \tag{15}$$

so that the closed-loop transfer function is

$$\frac{v_{com}(s)}{v(s)} = \frac{k_v g_{FV} s + k_v g_{IV}}{s^3 + a_v s^2 + k_v g_{PV} s + k_v g_{IV}} \tag{16}$$

Closed-loop poles, as well as the zero of the closed-loop transfer function, may be placed arbitrarily by proper choice of the gains. A possible design consists in placing the closed-loop poles at $s = -a/3$ and the zero at $s = -a/4$. This design corresponds to the so-called symmetric optimum [21] (which maximizes the phase margin) and is such that the zero almost cancels one of the poles. The desired poles and zeros are obtained for

$$g_{PV} = \frac{a_v^2}{3k_v}, \quad g_{IV} = \frac{a_v^3}{27k_v}, \quad g_{FV} = \frac{4a_v^2}{27k_v} \tag{17}$$

Although fixed gains were used in the simulations, real-time tuning of the gains using estimates of k_v and a_v would be straightforward using (17).

Total energy compensation may also be included in the controller, in order to improve the speed tracking during climb or descent. The concept of total energy compensation is described in [22] (and references therein), and is based on the definition of the total energy

$$E_{total} = PE + KE = mgh + \frac{1}{2}mv^2 \tag{18}$$

where PE is the potential energy and KE is the kinetic energy of the aircraft. Assuming that the aircraft is in steady flight with thrust T, and neglecting any variation in drag, the variation of the total energy due to an increment of thrust ΔT satisfies

$$\dot{E} = mg\dot{h} + mv\dot{v} = v\Delta T \tag{19}$$

Therefore, in order to maintain speed despite changes of altitude command, the following change in thrust should be applied

$$\Delta T = \frac{mg}{v}\dot{h} \tag{20}$$

Neglecting the first-order dynamics of the thrust response, (14) indicates that

$$\Delta T = \frac{mk_v}{a_v}\Delta\tau \tag{21}$$

where $\Delta\tau$ is the increment of throttle command needed to produce an increment of thrust ΔT. Defining τ_{TECS} to be the increment of throttle command needed to maintain the total energy in the presence of an altitude rate command, total energy compensation may be implemented as a simple cross-feed from the altitude controller to the velocity controller with

$$\tau_{TECS} = \left(\frac{a_v}{k_v}\right)\left(\frac{g}{v}\right)\dot{h}_{com} \triangleq g_{TECS}\dot{h}_{com} \tag{22}$$

where $\dot{h}_{com}$ is given by (7). The TECS compensation is added to (15), yielding the overall speed controller shown in Fig. 5.

F-16 Simulation Model & Implementation Details

The reconfigurable autopilot was evaluated using a nonlinear F-16 simulation that included: (a) 6 degree-of-freedom aircraft model with nonlinear aerodynamic tables, (b) first-order actuator models with position/rate saturation, (c) atmospheric model and engine thrust model. The aerodynamic data, the atmospheric model and the engine thrust model were obtained from [23] (pp. 584-592). The actuator models had poles at 20 rad/s, position limits $|\delta_e| \le 25\,\mathrm{deg}$, $|\delta_a| \le 21.5\,\mathrm{deg}$, $|\delta_r| \le 30\,\mathrm{deg}$, and rate limits $|\dot{\delta}_e| \le 60\,\mathrm{deg}/s$, $|\dot{\delta}_a| \le 80\,\mathrm{deg}/s$, $|\dot{\delta}_r| \le 120\,\mathrm{deg}/s$. Note that the actuator positions were not measured by the control algorithm, so that their dynamics constituted unmodelled dynamics. The possibility of a floating left elevator failure was included in the simulation. In that case, the left elevator was moved to the local angle of attack, which was computed as a function of the angle of the attack ([20], p. 34). The initial flight condition in the simulations was at 1,000ft altitude, 500ft/s speed. The sampling rate for the simulation and for the control law was 100Hz.

Parameters were chosen for the autopilot, based on what was considered reasonable, and were not optimized. Limits were similarly selected, and were not necessarily required or encountered in the simulations. Altogether, the following parameters were implemented in the autopilot.

- <u>Model reference controller:</u> $k = 4$ (leading to closed-loop poles of the inner reconfigurable control loop at -4 rad/s). Command limiting was applied to satisfy the position and rate limits by scaling the commands, and then the increments [18]. The code tested whether $\left|\det(CB)\right| \le 10^{-6} \max_{i,j} \left|CB_{ij}\right|$, to avoid singularity of the estimated matrix CB, but this situation was never encountered.
- <u>Recursive least-squares algorithm:</u> $\lambda = 0.97$, $\alpha = 10$. Commanded actuator positions were used for estimation.
- <u>Altitude controller:</u> $g_h = 0.2$, $g_{\dot{h}} = 0.6$ (leading to closed-loop poles at $-0.3 \pm j0.17$ rad/s), $g_\alpha = 1$. Limits on α_{com}: -10 to -30 deg. Limits on $\dot{h}_{com}$: $\pm 0.3v$. Lower limit on θ_{11}: 0.01. The measurement of the altitude rate was implemented by the first-order approximation for α, β, θ small

$$\dot{h} = v\left(\theta - \alpha\cos(\phi) - \beta\sin(\phi)\right) \tag{23}$$

- Heading controller: $g_\chi = 0.25$, $g_\phi = 1$(leading to 2 closed-loop poles at -0.5 rad/s). Limits on ϕ_{com}: ± 45 degrees. In the algorithm, the measurement of the heading angle was replaced by the first-order approximation for α, β, θ small

  $$\chi = \psi - \alpha \sin(\phi) + \beta \cos(\phi) \tag{24}$$

  The true heading angle, however, is shown in the plots of the simulation section.

- Sideslip controller: $g_\beta = 1$ (leading to a closed-loop pole at -1 rad/s).

- Speed controller: the following estimates were obtained for the engine response: $a_v = 1$, $k_v = 24$. Note that the thrust model ([23], pp. 586-587) implemented in the simulation has a time constant that varies with throttle setting. The gains were computed to be $g_{VF} = 0.0062$, $g_{VP} = 0.014$, $g_{VI} = 0.0015$ (leading to 3 closed-loop poles at -1/3 rad/s, and a zero at -1/4 rad/s). Anti-windup was implemented by stopping integration when the throttle limit was reached.

In the last simulation, measurement noise was added to test the robustness of the algorithm. The additive noise was generated using a gaussian random number generator, with the following standard deviations. $\alpha, \beta, \theta, \phi, \psi$: 0.1 deg. q, p, r: 0.1 deg/s. a_y, a_n: 0.01g. $\dot{p}$: $3 \deg / s^2$. $\dot{q}, \dot{r}$: $1.5 \deg / s^2$. $\overline{q}$: 0.2psf. v: 0.2 ft/s. h: 5ft. The numbers were selected by inspection of data collected during the VISTA F-16 experiments of [1]. The numbers are not meant to produce an exact representation of the measurement noise to be expected, but rather sufficiently representative numbers to test the robustness of the system. In the last simulation, a Dryden turbulence model was also implemented. The standard deviation of the wind gusts was 10 ft/s (moderate turbulence). In that case, the angle of attack and sideslip used by the controller were assumed to be provided by airspeed sensors.

Simulation Results

Responses to Heading Commands -- Turn Coordination

Fig. 6 shows the results of a first simulation demonstrating the turn coordination of the nonlinear autopilot. The aircraft was commanded to fly at an altitude of 1000 ft and a speed of 500 ft/s. Changes of heading command between 0 and 45 degrees were applied at regular intervals. The sideslip command was zero. The top/left plot of Fig. 6 shows the heading command (dashed) and the heading angle response (solid). In the transient portion of the heading response, the bank angle reaches a (limited) value of 45 degrees, resulting in a linearly increasing heading angle. The top/right response shows the response in altitude. The variations of altitude are small, considering the steep bank angles associated with the maneuver.

The bottom/left plot of Fig. 6 shows the angle of attack (solid) and the angle of sideslip (dashed). Note that the control law forces significant increases in angle of attack, in order to compensate for the loss of lift and maintain altitude. At the same time, low angles of sideslip are maintained. In other words, the control law automatically performs turn coordination, applying carefully balanced commands in all three axes.

The bottom/right plot of Fig. 6 shows the right elevator angle (solid) and the left elevator angle (dashed). Mid-way through the simulation, a floating elevator failure was emulated by moving the left tail surface to the local angle of attack. As can be seen from the bottom/right plot, the right elevator position shifted to restore trim, and larger elevator deflections were applied to compensate for the loss of effectiveness in the pitch axis. The inner reconfigurable control loop successfully estimated the parameters of the system, so that the effect of the failure has a barely observable impact on the other plots of Fig. 6.

Responses to Altitude Commands -- Total Energy Compensation

Fig. 7 shows the results of a simulation aimed at observing the response to changes in altitude command and the benefits of the total energy compensation. The top/left plot shows the response of altitude to two consecutive increases in altitude command of 2,000ft. Mid-way through the simulation, a floating left elevator failure was applied, and again no visible impact is observed on the responses.

The top/right plot of Fig. 7 shows the velocity response. Here and on the other plots, the dashed line shows the response obtained without total energy compensation, and the solid line shows the response with the compensation. The velocity response is significantly improved with the total energy compensation. The bottom/left plot of Fig. 7 shows the engine power applied in both cases. As expected, the total energy compensation results in an earlier application of the thrust, yielding improved velocity tracking. The total energy compensation also eliminates the undershoot in the engine power response arising when the altitude is reached.

The bottom/right plot shows the response of one of the two parameters estimated by the inner reconfigurable loop and used by the nonlinear autopilot. The plot shows $\theta_{11}\overline{q}$, which is the amount of normal acceleration (in g's) per degree of angle of attack. The plot shows that the parameter decreases slightly as the altitude increases. Note that the estimation results are consistent whether total energy compensation is used or not .

Responses to Altitude Commands at Low and High Altitudes

Fig. 8 shows the responses of the altitude autopilot to increments of 2,000ft at low and high altitudes. Specifically, steps were applied from 1,000ft to 3,000ft, and from 25,000ft to 27,000ft.

No failure was applied in this simulation and the speed reference stayed at 500ft/s. The left plot of Fig. 8 shows the increment of altitude in both cases, in order to compare the two responses. The solid line is the response at 1,000ft and the dashed line is the response at 25,000ft. There is very little difference between the two responses. However, the plot on the right, which shows the elevator deflections, indicates that the control signals required are significantly different. The autopilot is able to automatically adapt to changes associated with a wide difference in flight condition.

Responses to Velocity Commands -- Front and Back Sides of the Power Curve

Fig. 9 shows the result of a simulation (without failure) where the speed was decreased from 500ft/s to 250ft/s. This simulation was inspired from [24], and moves the aircraft from the front side of the power curve to the back side. The speed command is the dashed line of the top/left plot, and the speed response is the solid line. The altitude responses are shown on the top/right plot. The command was maintained at 1,000ft, and the response stayed close to the set-point, although a small altitude error appeared at the end of the run. Integral compensation could be added in the altitude controller to reduce the error, but was not applied because the error was generally found to be small.

The bottom/left plot of Fig. 9 shows the engine power which decreases first, then increases, as operation moves from the front side to the back side of the power curve. The bottom/right plot of Fig. 9 shows the angle of attack, which rises sharply when the back side of the power curve is reached. Note that the engine power setting was determined through the integrator of the velocity autopilot, while the angle of attack was determined through the elevator trim determined by the inner reconfigurable control loop.

Responses to Multiple Commands with Measurement Noise and Turbulence

Fig. 10 shows the response of the autopilot to multiple commands in the presence of noise and turbulence. The altitude command starts at 1,000ft and rises to 10,000ft after 20 seconds. The climb rate was arbitrarily limited in the autopilot to 30% of the speed. Since the speed is 500ft/s, it takes about 60 seconds for the aircraft to reach the desired altitude, as shown on the top/left plot. The heading angle command rises from 0 to 45 degrees after 30 seconds, and the response is shown on the top/right plot. The speed command rises from 500ft/s to 800ft/s after 60 seconds, and the response is shown on the bottom/left plot.

A floating left elevator failure was simulated, starting at 50 seconds. The elevator deflections are shown on the bottom/right plot of Fig. 10. The solid line is the right elevator, whose response is noisy due to the measurement noise (no filters were applied to the measured signals). The response, however, shows a similar adjustment for trim as previous responses, indicating successful reconfiguration. The dashed line is the response of the left elevator, which moves to the local angle of attack after 50 seconds, and is noisy due to the turbulence.

Conclusions

The reconfigurable autopilot proposed in this paper uses a reconfigurable flight control system as an inner loop, and a nonlinear compensator as an outer loop. The inner core provides tracking of pitch, roll, and yaw rate commands, using a model reference controller and a stabilized recursive least-squares algorithm. The outer loop is based on a linear design, plus some compensation for known nonlinear couplings, and adaptation to two parameters related to the lift curve. Overall, minimal prior information is required, and takes the form of desired closed-loop dynamics and limits to be specified by the user. Adjustment to flight conditions and failures occurs automatically through real-time adaptation.

The reconfigurable autopilot may be used as a controller for UAV's, or as an autopilot for piloted aircraft. Due to the modularity of the design, other reconfigurable control algorithms may be considered for the inner core. The two parameters of the lift curve may be estimated easily using the least-squares algorithm of this paper. The autopilot was found to perform well in simulations with a detailed F-16 model. A variety of trajectories were considered to demonstrate the automatic tuning through various flight conditions, the turn coordination, the reconfiguration after a left elevator failure, the ability to move across the power curve, and the tolerance to measurement noise and turbulence.

Tracking of velocity vectors in 3D space may be achieved directly by the algorithm. Trajectory tracking may be based on the proposed algorithm combined with an additional outer loop. An interesting problem is the design of optimal trajectories satisfying the limits of the aircraft. Another interesting objective would be for the autopilot to determine its own limits in real-time, for use in the trajectory generation.

Acknowledgements

Effort sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant F49620-98-1-0013. The views and conclusions contained in the paper are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research or the U.S. Government.

References

[1] Ward, D., Monaco, J., & Bodson, M., “Development and Flight Testing of a Parameter Identification Algorithm for Reconfigurable Control,” *Journal of Guidance, Control, and Dynamics*, vol. 2, no. 6, 1998, pp. 948-956.

[2] Brinker, J., & Wise, K., “Flight Testing of Reconfigurable Control Law on the X-36 Tailless Aircraft,” *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 5, 2001, pp. 903-909.

[3] Eberhardt, R., & Ward, D., “Indirect Adaptive Flight Control of a Tailless Fighter Aircraft,” *Proc. of the AIAA Guidance, Control, and Dynamics Conference*, Portland, OR, 1999, pp. 466-476.

[4] Doman, D., Ngo, A., Leggett, D., Saliers, M., & Pachter, M., “Development of a Hybrid Direct-Indirect Adaptive Control System for the X-33,” *Proc. of the AIAA Guidance, Control, and Dynamics Conference*, Denver, CO, 2000.

[5] Boskovic, J., & Mehra, R., “A Hybrid Fault-Tolerant Scheme for Flight Control Applications,” *Proc. of the AIAA Guidance, Navigation, and Control Conference*, Montreal, Canada, 2001.

[6] Buffington, J., Chandler, P., & Pachter, M., “Integration of On-line System Identification and Optimization-based Control Allocation,” *Proc. of the AIAA Guidance, Navigation, and Control Conference*, Boston, MA, 1998, pp. 1746-1756.

[7] Burken, J., Lu, P., Wu, Z., & Bahm, C., “Two Reconfigurable Flight-Control Design Methods: Robust Servomechanism and Control Allocation,” *Journal of Guidance, Control, and Dynamics,* vol. 24, no. 3, 2001, pp. 482-493.

[8] Calise, A., Lee, S., & Sharma, M., “Development of a Reconfigurable Flight Control Law for Tailless Aircraft,” *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 5, 2001, pp. 896-902.

[9] Elgersma, M., Enns, D., Shald, S., & Voulgaris, P., “Parameter Identification for Systems with Redundant Actuators,” *Proc. of the AIAA Guidance, Control, and Dynamics Conference*, Boston, MA, 1998, pp. 109-117.

[10] Siwakosit, W., & Hess, R., “Multi-Input/Multi-Output Reconfigurable Flight Control Design,” *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 6, 2001, pp. 1079-1088.

[11] Steinberg, M., “Comparison of Intelligent, Adaptive, and Nonlinear Flight Control Laws,” *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 4, 2001, pp. 693-699.

[12] Bodson, M., & Groszkiewicz, J., “Multivariable Adaptive Algorithms for Reconfigurable Flight Control,” *IEEE Trans. on Control Systems Technology,* vol. 5, no. 2, 1997, pp. 217-229.

[13] Napolitano, M., Naylor, S., Neppach, C., & Casdorph, V., “On-Line Learning Nonlinear Direct Neurocontrollers for Restructurable Control Systems,” *Journal of Guidance, Control, and Dynamics*, vol. 18, no. 1, 1995, pp. 170-176.

[14] Chandler, P., Pachter, M., & Rasmussen, S., “UAV Cooperative Control,” *Proc. of the American Control Conference*, Arlington, VA, 2001, pp. 50-55.

[15] McLain, T., & Beard, R., “Trajectory Planning for Coordinated Rendezvous of Unmanned Air Vehicles,” *Proc. of the AIAA Guidance, Control, and Dynamics Conference*, Denver, CO, 2000.

[16] McLean, D., *Automatic Flight Control Systems*, Prentice-Hall, New York, 1990.

[17] Bodson, M., “An Adaptive Algorithm with Information-Dependent Data Forgetting,” *Proc. of the American Control Conference*, Seattle, WA, 1995, pp. 3485-3489.

[18] Bodson, M., & Pohlchuck, W., “Command Limiting in Reconfigurable Flight Control,” *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 4, 1998, pp. 639-646.

[19] Astrom, K.J., & Wittenmark, B., *Adaptive Control*, 2nd edition, Addison-Wesley, Reading, MA, 1995.

[20] Blakelock, J., *Automatic Control of Aircraft and Missiles*, 2nd ed., Wiley & Sons, New York, 1991.

[21] Preitl, S., & Precup, R.-E., "An Extension of Tuning Relations after Symmetrical Optimum Method for PI and PID Controllers," *Automatica*, vol. 35, 1999, pp. 1731-1736.

[22] Voth, C. & Ly, U.-L., "Design of a Total Energy Control Autopilot Using Constrained Parameter Optimization," *Journal of Guidance, Control, and Dynamics*, vol. 14, no. 5, 1991, pp. 927-935.

[23] Stevens, B., & Lewis, F., *Aircraft Control and Simulation*, Wiley-Interscience, New York, NY, 1992.

[24] Schindeler, N., & Pachter, M., "Phugoid Dynamics Control," *Proc. of the AIAA Guidance, Navigation, and Control Conference*, Montreal, Canada, 2001.

Appendix

The recursive identification algorithm was initialized with parameters corresponding to the unfailed aircraft at the nominal flight condition. The parameters were obtained by off-line identification using test signals, rather than by linearization of the nonlinear model, and are approximate. Small parameters were set to zero. The state-space matrices given below correspond to the initial parameters. With the "d" term, the trim values of the states and controls are zero, except α=2.11 degrees and $\delta_E = -0.759$ degrees.

$$
\begin{gathered}
A = \begin{pmatrix} -1.0913 & 1.0000 & 0. & 0. & 0. \\ 0.7289 & -0.9833 & 0. & 0. & 0. \\ 0. & 0. & -0.3029 & 0.0006 & -0.9923 \\ 0. & 0. & -29.6271 & -3.4107 & 0.9950 \\ 0. & 0. & 7.5970 & -0.1088 & -0.4777 \end{pmatrix}, \\
B = \begin{pmatrix} 0. & 0. & 0. \\ -9.5405 & 0. & 0. \\ 0. & 0.0273 & 0.0428 \\ 0. & -39.3939 & 7.2914 \\ 0. & -2.6000 & -3.2625 \end{pmatrix}, \quad d = \begin{pmatrix} 2.3026 \\ -8.7792 \\ 0 \\ 0 \\ 0 \end{pmatrix}
\end{gathered}
\tag{25}
$$

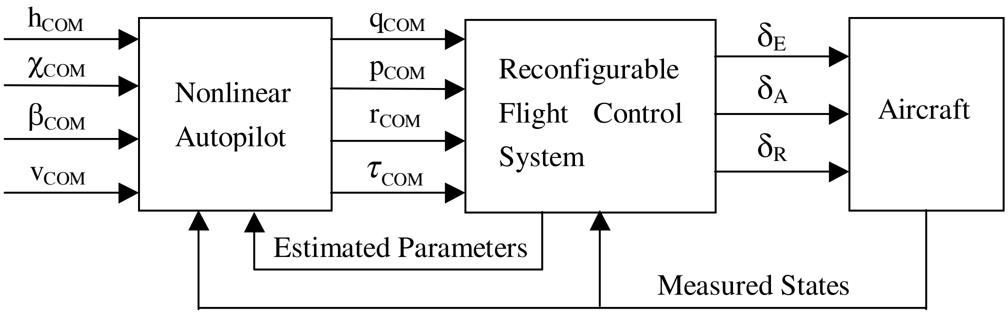

Figure 1: Reconfigurable nonlinear autopilot

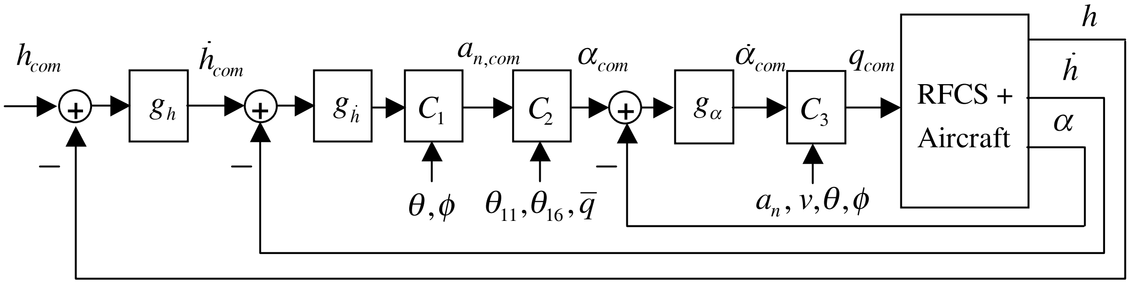

Figure 2: Altitude controller

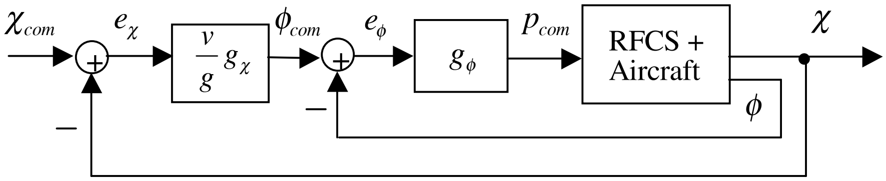

Figure 3: Heading controller

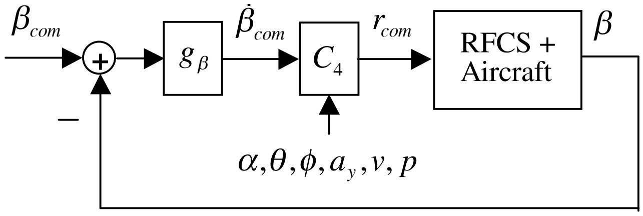

Figure 4: Sideslip controller

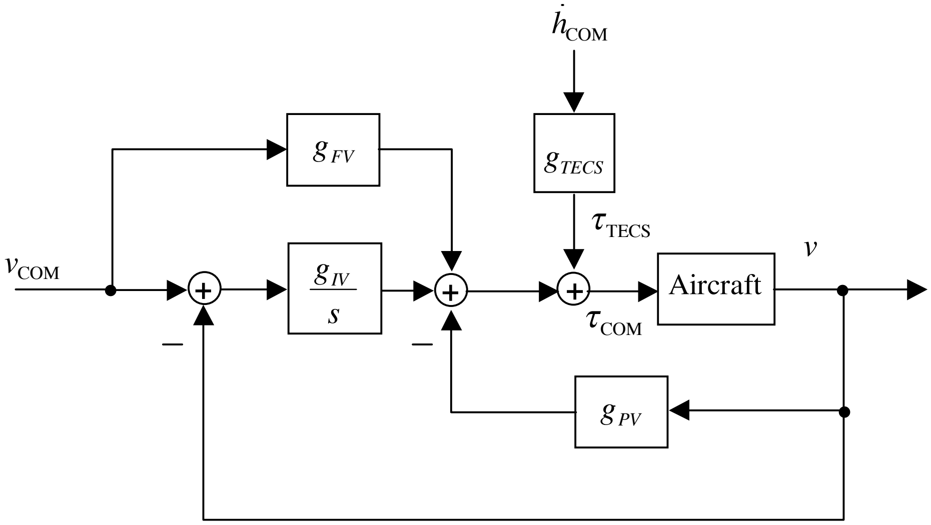

Figure 5: Speed controller

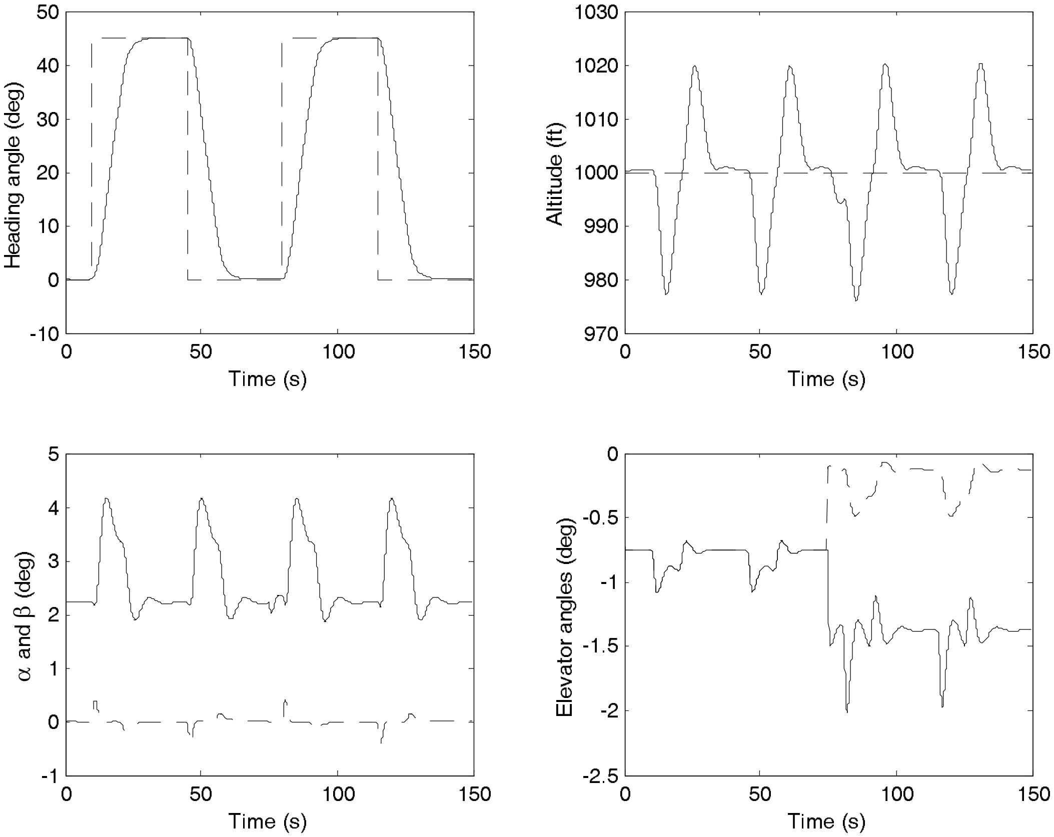

Figure 6: Responses to heading commands

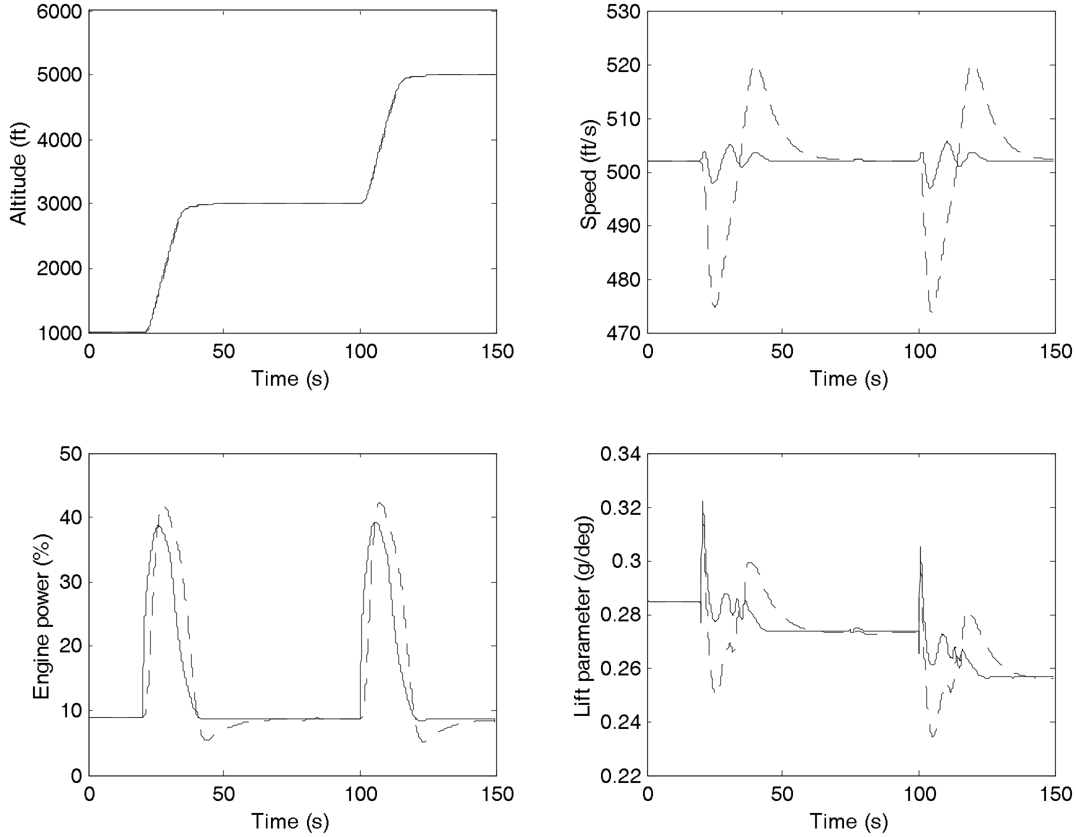

Figure 7: Responses to altitude commands

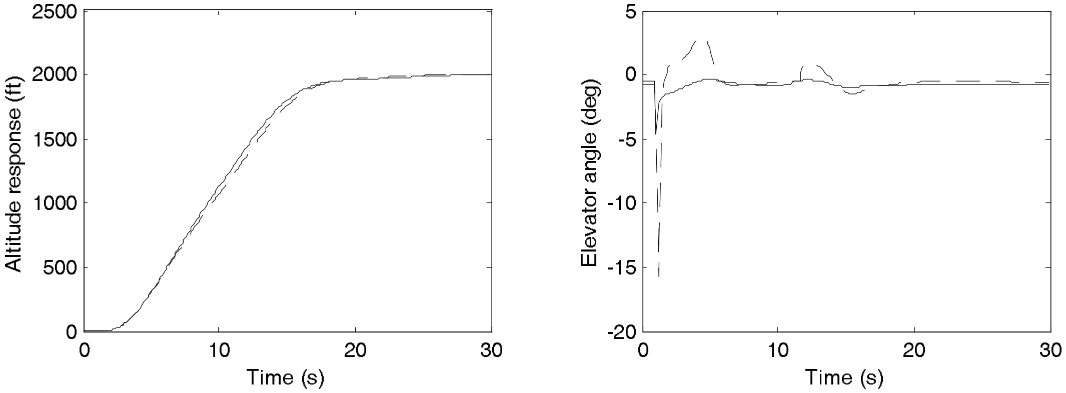

Figure 8: Responses to altitude commands at low and high altitudes

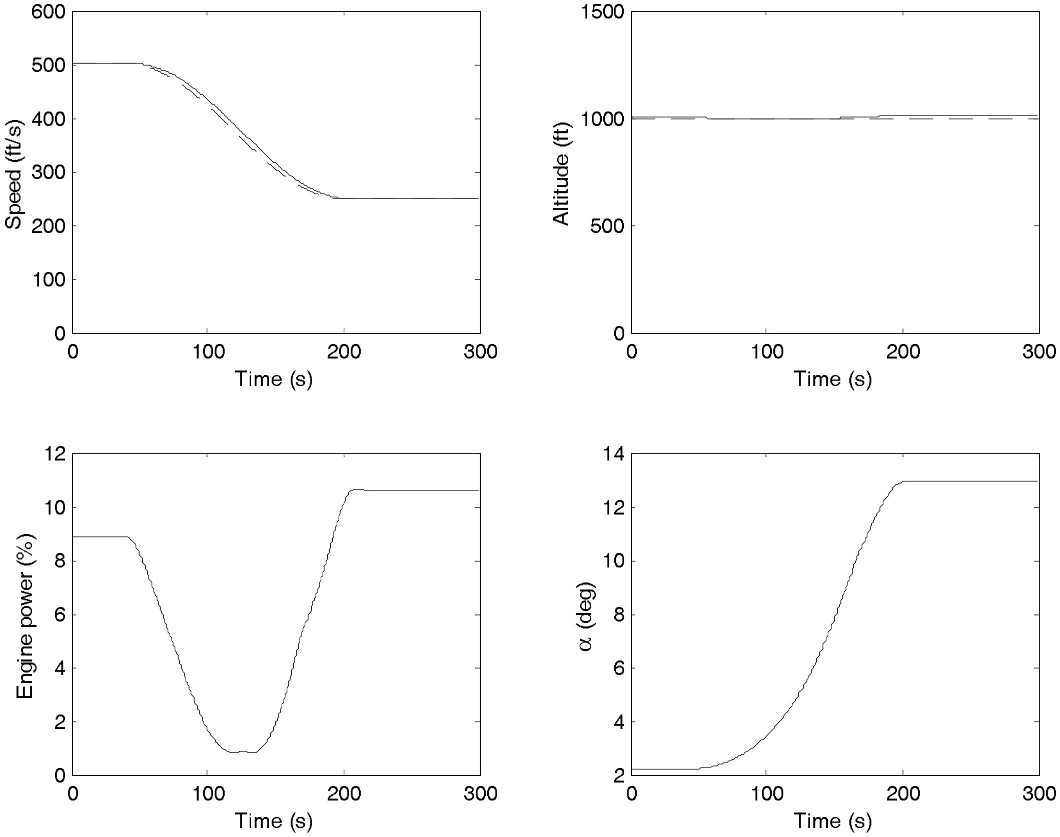

Figure 9: Responses to velocity commands across the power curve

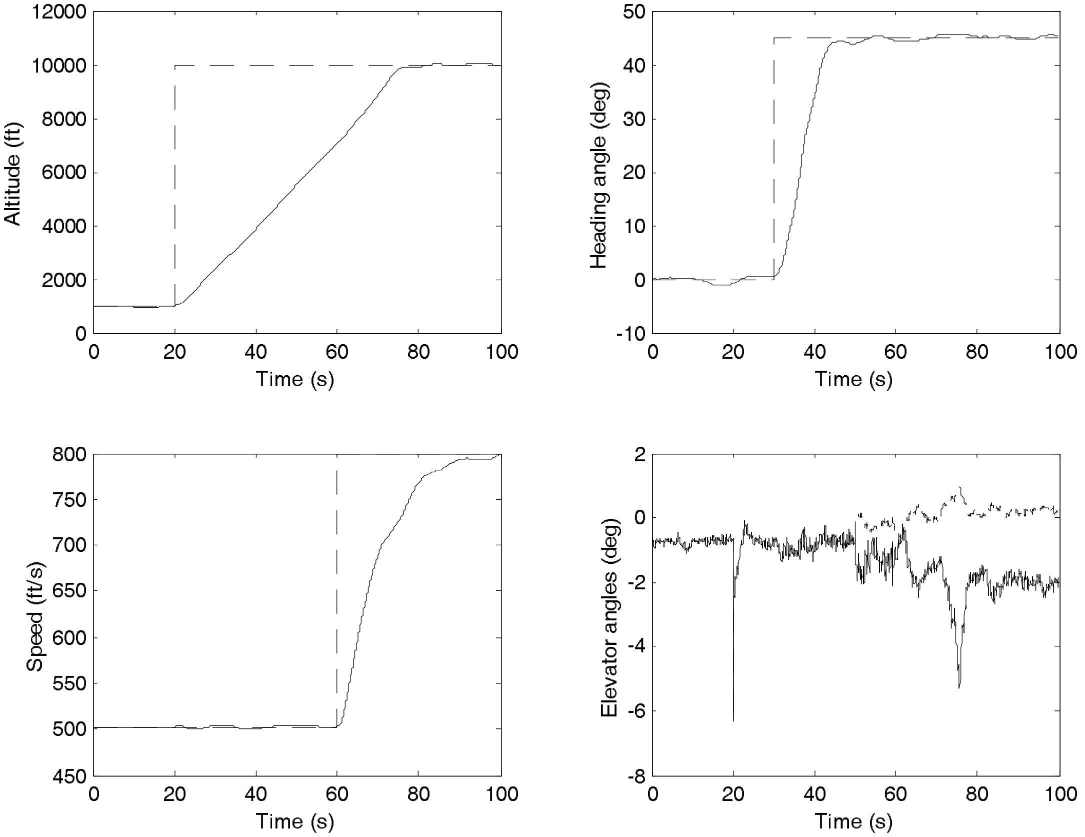

Figure 10: Responses to multiple commands with measurement noise and turbulence