

Research Article

A Unified Approach to Nonlinear Dynamic Inversion Control with Parameter Determination by Eigenvalue Assignment

Yu-Chi Wang, Donglong Sheu, and Chin-E Lin

Department of Aeronautics and Astronautics, National Cheng Kung University, Tainan 701, Taiwan

Correspondence should be addressed to Chin-E Lin; chinelin@mail.ncku.edu.tw

Received 8 September 2015; Revised 13 November 2015; Accepted 16 November 2015

Academic Editor: Rongwei Guo

Copyright © 2015 Yu-Chi Wang 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 a unified approach to nonlinear dynamic inversion control algorithm with the parameters for desired dynamics determined by using an eigenvalue assignment method, which may be applied in a very straightforward and convenient way. By using this method, it is not necessary to transform the nonlinear equations into linear equations by feedback linearization before beginning control designs. The applications of this method are not limited to affine nonlinear control systems or limited to minimum phase problems if the eigenvalues of error dynamics are carefully assigned so that the desired dynamics is stable. The control design by using this method is shown to be robust to modeling uncertainties. To validate the theory, the design of a UAV control system is presented as an example. Numerical simulations show the performance of the design to be quite remarkable.

1. Introduction

In the development of high-performance aircraft, control difficulties may be encountered over some parts of flight envelope. These difficulties arise from highly nonlinear aerodynamic properties [1] in some flight conditions. In order to solve these control difficulties, nonlinear controllers are required for high-performance aircraft.

Among many control methods, nonlinear dynamic inversion (NDI) is very popular and has been widely studied for flight control designs (e.g., [2, 3]). NDI-based control systems are usually divided into fast- and slow-loop control subsystems according to the multiple time-scale method [4]. In each subsystem, Lie derivatives [5] are used to transform the nonlinear equations into linear equations. Then, linear control design methods can be employed and the control inputs are obtained by converting the linear system control variables into the original coordinates. However, the control systems obtained by feedback linearization [6] may have nonminimum phase problems for affine or nonaffine nonlinear system [7] and robust issues in case of model mismatch. A typical nonminimum phase problem may be found in flight dynamics where the altitude-elevator transfer function usually has a right-half zero. The internal state control [8] is

often used to overcome these nonminimum phase problems. In addition, the fuzzy logic control [9] was also applied for solving these kinds of problems. Furthermore, to overcome the robust problems, μ -analysis [10] and H_∞ method [11] were applied. Specially, incremental NDI (INDI) [12] was used to increase the robustness to aerodynamic uncertainties by calculating the control surface deflection changes instead of giving inputs directly.

To circumvent some aforementioned robust problems, an adaptive nonlinear model inversion control [13] was introduced, in which the design concept is similar to the conventional NDI yet without linearizing the nonlinear system. The model inversion method replaces the motion rates with a P-form or PI-form desired dynamics to negate the original dynamics. The choice of parameters in the desired dynamics is based on the bandwidth of response and time scales. The effects of different types of desired dynamics on the resulted control system were discussed [14].

Although the aforementioned NDI approaches are successful in many flight control system designs [14–16] over a large part of the flight envelope, the systems of dynamics in general have to be separated into several subsystems according to the rates of response. There are many cases in which the fast rate and the slow rate might not be

distinguished so clearly however. Also, although the pole assignment method had been introduced to determine the parameters in the desired dynamics, very often the control system must first be transformed into a standard feedback control form from which the standard eigenvalue assignment method can be applied.

In consideration of the aforementioned problems existing in the current literature on control designs, a unified approach to nonlinear dynamic inversion control is proposed in this paper. The equations of motion will not be necessary to be separated into fast rate and slow rate groups, nor will they be limited to an affine system. Feedback linearizations will not be required to transform the nonlinear equations into linear equations. Nonminimum phase problems are solved by eigenvalue assignments for error dynamics. An iterative method for determining the parameters of the desired dynamics from the assigned eigenvalues of error dynamics is proposed. Analysis of robustness to model uncertainties or disturbances is conducted. This method will be ready for design without simplifying the system of equations based on physical insights once the governing dynamic equations are established and the state variables to be tracked are selected. The theory is to be developed in detail in the following sections. A UAV is introduced and its control system is designed with the developed method. Numerical simulations are conducted to validate this method.

2. A Unified Approach to Nonlinear Dynamic Inversion Control

2.1. Nonlinear Dynamic Inversion Control. In general, dynamic equations of motion with control inputs can be expressed by

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad (1)$$

where $\mathbf{x} \in R^n$ is the state vector, $\mathbf{u} \in R^m$ ($m < n$) the control vector, and \mathbf{f} the nonlinear function representing the model of dynamics with controls. By extending the concept of dynamic inversion (DI) [4], the control vector \mathbf{u} can be assumed to be computed from

$$\dot{\mathbf{x}}_d = \mathbf{f}(\mathbf{x}_d, \mathbf{u}), \quad (2)$$

where $\mathbf{x}_d \in R^n$ is a desired state vector with its rate of change being designated.

In this paper, the desired dynamics is designated as a set of stable first-order differential equations:

$$\dot{\mathbf{x}}_d = \mathbf{\Omega}(\mathbf{x}_d - \mathbf{x}), \quad (3)$$

where $\mathbf{\Omega} \in R^{n \times n}$ represents a constant matrix with n independent parameters which can be chosen. Substituting (3) into (2) yields

$$\mathbf{\Omega}(\mathbf{x}_d - \mathbf{x}) = \mathbf{f}(\mathbf{x}_d, \mathbf{u}) \quad (4)$$

which constitutes a set of n algebraic equations. Since $m < n$, obviously, \mathbf{u} cannot satisfy (4) if all elements of \mathbf{x}_d are to be designated. It means that only part of \mathbf{x}_d can be designated.

So let \mathbf{x}_d be divided into two groups, say $\mathbf{x}_c \in R^m$ and $\mathbf{x}_r \in R^{n-m}$, where \mathbf{x}_c contains the state variables which are to be controlled or designated and \mathbf{x}_r the residual ones. Both \mathbf{u} and \mathbf{x}_r constitute n unknown variables which are to be determined from (4). To solve a set of nonlinear algebraic equations, the Newton-Raphson iteration method can be employed as follows:

$$\begin{bmatrix} \mathbf{u} \\ \mathbf{x}_r \end{bmatrix}_{k+1} = \begin{bmatrix} \mathbf{u} \\ \mathbf{x}_r \end{bmatrix}_k - \left[\frac{\partial \mathbf{f}_{\text{err}}}{\partial \mathbf{u}} \quad \frac{\partial \mathbf{f}_{\text{err}}}{\partial \mathbf{x}_r} \right]^{-1} \mathbf{f}_{\text{err}}, \quad (5)$$

where $\mathbf{f}_{\text{err}} \triangleq \mathbf{f}(\mathbf{x}_d, \mathbf{u}) - \mathbf{\Omega}(\mathbf{x}_d - \mathbf{x})$ and $k = 0, 1, 2, \dots$ represents the iteration number.

2.2. Parameter Determination. Now, a question arises whether the state vector \mathbf{x} will asymptotically follow the desired vector \mathbf{x}_d if \mathbf{u} is determined from (4) and substituted in (1). In order to answer it, let (1) and (4) be examined more carefully as follows.

Subtracting (4) from (1) yields

$$\dot{\mathbf{x}} - \mathbf{\Omega}(\mathbf{x}_d - \mathbf{x}) = \mathbf{f}(\mathbf{x}, \mathbf{u}) - \mathbf{f}(\mathbf{x}_d, \mathbf{u}). \quad (6)$$

If \mathbf{x} is very close to \mathbf{x}_d , then the above equation can be linearized as follows:

$$\dot{\mathbf{x}} - \mathbf{\Omega}(\mathbf{x}_d - \mathbf{x}) \approx \mathbf{f}_x(\mathbf{x}_d, \mathbf{u})(\mathbf{x} - \mathbf{x}_d). \quad (7)$$

With the concept that the desired variables \mathbf{x}_d are near constant, say $\dot{\mathbf{x}}_d \approx \mathbf{0}$, (7) can be rewritten as

$$\dot{\mathbf{x}} - \dot{\mathbf{x}}_d \approx [-\mathbf{\Omega} + \mathbf{f}_x(\mathbf{x}_d, \mathbf{u})](\mathbf{x} - \mathbf{x}_d). \quad (8)$$

Defining an error vector $\mathbf{e} = \mathbf{x} - \mathbf{x}_d$ and replacing the approximate sign with the equal sign lead (8) to

$$\dot{\mathbf{e}} = [-\mathbf{\Omega} + \mathbf{f}_x(\mathbf{x}_d, \mathbf{u})]\mathbf{e}. \quad (9)$$

Equation (9) is a set of error dynamics in which the error vector \mathbf{e} will vanish eventually if all the real parts of the eigenvalues of $[-\mathbf{\Omega} + \mathbf{f}_x(\mathbf{x}_d, \mathbf{u})]$ are negative. This is possible if $\mathbf{\Omega}$ is chosen appropriately. It means that the state vector \mathbf{x} can approach the desired vector \mathbf{x}_d once \mathbf{e} approaches $\mathbf{0}$. Recall that \mathbf{x}_d contains \mathbf{x}_c , a state vector to be tracked.

Notice that the eigenvalue λ in (9) can be determined by

$$|\lambda \mathbf{I} + \mathbf{\Omega} - \mathbf{f}_x(\mathbf{x}_d, \mathbf{u})| = 0, \quad (10)$$

where \mathbf{I} is an identity matrix. For simplicity, if $\mathbf{\Omega}$ is diagonal and all elements are the same, say, $\mathbf{\Omega} = \sigma \mathbf{I}$, then (10) can be rewritten as

$$|(\lambda + \sigma) \mathbf{I} - \mathbf{f}_x(\mathbf{x}_d, \mathbf{u})| = 0. \quad (11)$$

In order to make \mathbf{e} bounded, σ can be so chosen that

$$\text{real}(\lambda_i + \sigma) < 0 \quad (12)$$

for all λ_i ($i = 1, 2, \dots, n$). Although this method is simple, the resulting σ may be unnecessarily large.

For more general cases, $\bar{\Omega}$ contains a set of parameters, σ_k , ($k = 1, 2, 3, \dots, n$), which may be chosen to determine the eigenvalues and eigenvectors of $-\bar{\Omega} + \mathbf{f}_x(\mathbf{x}_d, \mathbf{u})$. Recall that (3) must be a stable model and, therefore, the simplest way of constructing $\bar{\Omega}$ is to let all its elements vanish except those at the diagonal, which are assumed to be $\Omega_{kk} = \sigma_k > 0$. In fact, some off-diagonal elements can also be allowed to exist. For example, let $\Omega_{kk} = \Omega_{ll} = \sigma_k > 0$, and $\Omega_{kl}^2 = \Omega_{lk}^2 = \sigma_l^2 < \sigma_k^2$. It is trivial to prove that the latter case is also a stable model.

Now, assume that the i th ($i = 1, 2, 3, \dots, n$) eigenvalue and eigenvector are $\bar{\lambda}_i$ and $\bar{\mathbf{e}}_i$, respectively. Accordingly,

$$[-\bar{\Omega} + \mathbf{f}_x(\mathbf{x}_d, \mathbf{u})] \bar{\mathbf{e}}_i = \bar{\lambda}_i \bar{\mathbf{e}}_i. \quad (13)$$

In general, the eigenvector $\bar{\mathbf{e}}_i$ can be normalized so that

$$\bar{\mathbf{e}}_i^T \bar{\mathbf{e}}_i = 1. \quad (14)$$

However, if the eigenvalue λ_i is desired rather than $\bar{\lambda}_i$, then it is necessary to adjust the parameters in $\bar{\Omega}$. Assume that an increment $(\partial\bar{\Omega}/\partial\sigma_k)\Delta\sigma_k$ to $\bar{\Omega}$ is required. Then

$$\left[-\bar{\Omega} - \frac{\partial\bar{\Omega}}{\partial\sigma_k}\Delta\sigma_k + \mathbf{f}_x(\mathbf{x}_d, \mathbf{u})\right] \mathbf{e}_i = \lambda_i \mathbf{e}_i, \quad (15)$$

where λ_i and \mathbf{e}_i are assumed to be approximated to $\bar{\lambda}_i + (\partial\lambda_i/\partial\sigma_k)\Delta\sigma_k$ and $\bar{\mathbf{e}}_i + (\partial\mathbf{e}_i/\partial\sigma_k)\Delta\sigma_k$, respectively, and \mathbf{e}_i is also normalized so that

$$\left(\bar{\mathbf{e}}_i + \frac{\partial\mathbf{e}_i}{\partial\sigma_k}\Delta\sigma_k\right)^T \left(\bar{\mathbf{e}}_i + \frac{\partial\mathbf{e}_i}{\partial\sigma_k}\Delta\sigma_k\right) = 1. \quad (16)$$

Accordingly, the derivative of (13) with respect to σ_k results in

$$\left[-\frac{\partial\bar{\Omega}}{\partial\sigma_k}\right] \bar{\mathbf{e}}_i + [-\bar{\Omega} + \mathbf{f}_x(\mathbf{x}_d, \mathbf{u})] \frac{\partial\bar{\mathbf{e}}_i}{\partial\sigma_k} = \frac{\partial\lambda_i}{\partial\sigma_k} \bar{\mathbf{e}}_i + \bar{\lambda}_i \frac{\partial\bar{\mathbf{e}}_i}{\partial\sigma_k} \quad (17)$$

and the derivative of (14) becomes

$$\bar{\mathbf{e}}_i^T \frac{\partial\bar{\mathbf{e}}_i}{\partial\sigma_k} = 0. \quad (18)$$

Equations (17) and (18) can be rearranged to

$$\begin{bmatrix} \bar{\mathbf{e}}_i & \bar{\lambda}_i \mathbf{I} + \bar{\Omega} - \mathbf{f}_x(\mathbf{x}_d, \mathbf{u}) \\ 0 & \bar{\mathbf{e}}_i^T \end{bmatrix} \begin{Bmatrix} \frac{\partial\lambda_i}{\partial\sigma_k} \\ \frac{\partial\bar{\mathbf{e}}_i}{\partial\sigma_k} \end{Bmatrix} = \begin{Bmatrix} -\frac{\partial\bar{\Omega}}{\partial\sigma_k} \bar{\mathbf{e}}_i \\ 0 \end{Bmatrix} \quad (19)$$

from which $\partial\lambda_i/\partial\sigma_k$ can be determined. Note that $\partial\lambda_i/\partial\sigma_k$ represents the variation of λ_i with respect to σ_k . With all $\partial\lambda_i/\partial\sigma_k$, ($k = 1, 2, 3, \dots, n$), the i th revised eigenvalue should be

$$\lambda_i = \bar{\lambda}_i + \sum \frac{\partial\lambda_i}{\partial\sigma_k} \Delta\sigma_k, \quad (i = 1, 2, 3, \dots, n). \quad (20)$$

Recall that, in this equation, $\bar{\lambda}_i$ represents the i th eigenvalue of $[-\bar{\Omega} + \mathbf{f}_x(\mathbf{x}_d, \mathbf{u})]$, where $\bar{\Omega}$ contains the parameters $\bar{\sigma}_k$,

($k = 1, 2, 3, \dots, n$). Now since λ_i is the i th desired eigenvalue, (20) in fact becomes a set of n simultaneous equations from which $\Delta\sigma_k$ can be solved. With $\bar{\sigma}_k$ being the initial guess, revisions for parameters can be made by

$$\sigma_k = \bar{\sigma}_k + \Delta\sigma_k. \quad (21)$$

Since λ_i , ($i = 1, 2, 3, \dots, n$) are nonlinear function of σ_k , ($k = 1, 2, 3, \dots, n$), iterative computations of (9) and (19)–(21) are required in order to obtain a set of convergent σ_k . To make it clear, the iteration procedures are summarized as follows.

- (1) Let the desired eigenvalues be λ_i ($i = 1, 2, 3, \dots, n$) which are all distinct.
- (2) Guess a set of parameters $\bar{\sigma}_k$ ($k = 1, 2, 3, \dots, n$).
- (3) Determine the eigenvalues and eigenvectors of (9). Denote the i th eigenvalue and eigenvector as $\bar{\lambda}_i$ and $\bar{\mathbf{e}}_i$, respectively.
- (4) With each $\bar{\sigma}_k$ ($k = 1, 2, 3, \dots, n$), determine $\partial\lambda_i/\partial\sigma_k$ ($i = 1, 2, 3, \dots, n$) from (19).
- (5) Determine $\Delta\sigma_k$ ($k = 1, 2, 3, \dots, n$) from (20). If $|\Delta\sigma_k|$ is less than a preset small value, then stop; else continue with the next step.
- (6) Determine σ_k from (21).
- (7) Replace $\bar{\sigma}_k$ with σ_k and go to step 3.

At this point, it must be mentioned that the iterations will converge only if the initial guesses are very close to the true answers. It is also emphasized that the desired eigenvalues λ_i must be chosen to lie on the left-half s -plane and the resulting parameters in $\bar{\Omega}$ must satisfy the stable model requirements in (3).

2.3. Robust Analysis. The nonlinear dynamic inversion control design developed so far is based on a nominal dynamical model. In the real world, however, there always exist some modeling uncertainties which cannot be determined in advance. In order to check if the control design is robust, let the actual dynamical model be represented by

$$\dot{\mathbf{x}} = \mathbf{f}'(\mathbf{x}, \mathbf{u}). \quad (22)$$

With this assumption, now (6) can be modified to

$$\dot{\mathbf{x}} - \bar{\Omega}(\mathbf{x}_d - \mathbf{x}) = \mathbf{f}'(\mathbf{x}, \mathbf{u}) - \mathbf{f}(\mathbf{x}_d, \mathbf{u}) \quad (23)$$

which can be rewritten as follows:

$$\begin{aligned} \dot{\mathbf{x}} + \bar{\Omega}\mathbf{x} = & \bar{\Omega}\mathbf{x}_d + [\mathbf{f}_x(\mathbf{x}_d, \mathbf{u})(\mathbf{x} - \mathbf{x}_d) + \Delta\mathbf{f}_n(\mathbf{x}, \mathbf{x}_d, \mathbf{u}) \\ & + \mathbf{f}'(\mathbf{x}, \mathbf{u}) - \mathbf{f}(\mathbf{x}, \mathbf{u})], \end{aligned} \quad (24)$$

where $\Delta \mathbf{f}_n(\mathbf{x}, \mathbf{x}_d, \mathbf{u})$ denotes the nonlinear part of $\mathbf{f}(\mathbf{x}, \mathbf{u}) - \mathbf{f}(\mathbf{x}_d, \mathbf{u})$. The solution of the above equation can be represented as follows:

$$\begin{aligned} \mathbf{x}(t) = & e^{-\Omega t} \mathbf{x}_0 + (\mathbf{I} - e^{-\Omega t}) \mathbf{x}_d \\ & + \int_0^t e^{-\Omega(t-\tau)} \left\{ \mathbf{f}_x(\mathbf{x}_d, \mathbf{u}(\tau)) (\mathbf{x}(\tau) - \mathbf{x}_d) \right. \\ & + \Delta \mathbf{f}_n(\mathbf{x}(\tau), \mathbf{x}_d, \mathbf{u}(\tau)) + \mathbf{f}'(\mathbf{x}(\tau), \mathbf{u}(\tau)) \\ & \left. - \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau)) \right\} d\tau. \end{aligned} \quad (25)$$

Note that from (9), the eigenvalues of $-\Omega + \mathbf{f}_x(\mathbf{x}_d, \mathbf{u})$ are all in the left-half s -plane since Ω has been determined with that assumption. Also, at this point, it is not unreasonable to assume that both $\Delta \mathbf{f}_n(\mathbf{x}(\tau), \mathbf{x}_d, \mathbf{u}(\tau))$ and the modeling differences $|\mathbf{f}'(\mathbf{x}, \mathbf{u}) - \mathbf{f}(\mathbf{x}, \mathbf{u})|$ are bounded. Therefore, if the model in (3) is stable enough, the integral part in (25) will vanish along with $e^{-\Omega t}$. Accordingly, as the time t gets large enough, the state variables \mathbf{x} will approach the desired values \mathbf{x}_d as can be found from (25) even if there are some modeling uncertainties or disturbances.

3. Nonlinear Dynamic Inversion Flight Control System Design for a UAV: An Example

3.1. Flight Dynamics Equations of Motion. To illustrate the theory, a design of flight control system with the method developed is presented. Before the flight control design proceeds, a set of flight dynamics equations of motion must be formulated. Note that all aerodynamic forces and moments result from the relative motions between aircraft and the air. The aircraft is assumed to have a ground velocity:

$$\mathbf{V} = u\mathbf{i} + v\mathbf{j} + w\mathbf{k} = \dot{X}_E\mathbf{I} + \dot{Y}_E\mathbf{J} - \dot{h}\mathbf{K}, \quad (26)$$

where $(\mathbf{i}, \mathbf{j}, \mathbf{k})$ are unit vectors in the aircraft body moving frame and $(\mathbf{I}, \mathbf{J}, \mathbf{K})$ the unit vectors in a fixed flat earth frame. The air is assumed to have a velocity:

$$\mathbf{V}_w = u_g\mathbf{i} + v_g\mathbf{j} + w_g\mathbf{k} = V_{w_x}\mathbf{I} + V_{w_y}\mathbf{J} - V_{w_z}\mathbf{K} \quad (27)$$

which is also known as the wind velocity. Then, the velocity of the aircraft relative to the air can be represented by

$$\mathbf{V}_a = \mathbf{V} - \mathbf{V}_w = (u - u_g)\mathbf{i} + (v - v_g)\mathbf{j} + (w - w_g)\mathbf{k} \quad (28)$$

from which, the aircraft total speed relative to the air, the angle of attack, and the sideslip angle can be determined, respectively, by the following equations:

$$\begin{aligned} V_a = & \sqrt{(u - u_g)^2 + (v - v_g)^2 + (w - w_g)^2} \\ \alpha = & \tan^{-1} \frac{w - w_g}{u - u_g} \\ \beta = & \tan^{-1} \frac{v - v_g}{\sqrt{(u - u_g)^2 + (w - w_g)^2}}. \end{aligned} \quad (29)$$

With the assumptions of fixed flat earth and winds being present, the motions of aircraft with six degrees of freedom can be represented by a set of nonlinear first-order differential equations as follows:

$$\begin{aligned} \dot{X}_E & \triangleq f_X \\ & = V_a (C_{11} \cos \alpha \cos \beta + C_{21} \sin \beta + C_{31} \sin \alpha \cos \beta) \\ & \quad + V_{w_x} \end{aligned} \quad (30)$$

$$\begin{aligned} \dot{Y}_E & \triangleq f_Y \\ & = V_a (C_{12} \cos \alpha \cos \beta + C_{22} \sin \beta + C_{32} \sin \alpha \cos \beta) \\ & \quad + V_{w_y}, \end{aligned} \quad (31)$$

$$\begin{aligned} \dot{h} & \triangleq f_h \\ & = -V_a (C_{13} \cos \alpha \cos \beta + C_{23} \sin \beta + C_{33} \sin \alpha \cos \beta) \\ & \quad + V_h \end{aligned} \quad (32)$$

$$\dot{\psi} \triangleq f_\psi = q \sec \theta \sin \phi + r \sec \theta \cos \phi \quad (33)$$

$$\dot{\theta} \triangleq f_\theta = q \cos \phi - r \sin \phi \quad (34)$$

$$\dot{\phi} \triangleq f_\phi = p + q \tan \theta \sin \phi + r \tan \theta \cos \phi \quad (35)$$

$$\begin{aligned} \dot{V}_a & \triangleq f_V \\ & = \frac{F_x}{m} \cos \alpha \cos \beta + \frac{F_y}{m} \sin \beta + \frac{F_z}{m} \sin \alpha \cos \beta + f_{V_w} \end{aligned} \quad (36)$$

$$\begin{aligned} \dot{\alpha} & \triangleq f_\alpha \\ & = -p \cos \alpha \tan \beta + q - r \sin \alpha \tan \beta \\ & \quad - \frac{F_x}{mV_a} \sin \alpha \sec \beta + \frac{F_z}{mV_a} \cos \alpha \sec \beta + f_{\alpha_w} \end{aligned} \quad (37)$$

$$\begin{aligned} \dot{\beta} & \triangleq f_\beta \\ & = p \sin \alpha - r \cos \alpha - \frac{F_x}{mV_a} \cos \alpha \sin \beta + \frac{F_y}{mV_a} \cos \beta \\ & \quad - \frac{F_z}{mV_a} \sin \alpha \sin \beta + f_{\beta_w} \end{aligned} \quad (38)$$

$$\begin{aligned} \dot{p} & \triangleq f_p \\ & = \frac{I_{xz}(-I_{xx} + I_{yy} - I_{zz})}{I_{xx}I_{zz} - I_{xz}I_{xz}} pq \\ & \quad + \frac{I_{yy}I_{zz} - I_{zz}I_{zz} - I_{xz}I_{xz}}{I_{xx}I_{zz} - I_{xz}I_{xz}} qr + \frac{I_{zz}}{I_{xx}I_{zz} - I_{xz}I_{xz}} M_x \\ & \quad - \frac{I_{xz}}{I_{xx}I_{zz} - I_{xz}I_{xz}} M_z \end{aligned} \quad (39)$$

$$\dot{q} \triangleq f_q = \frac{I_{xz}}{I_{yy}} (p^2 - r^2) + \frac{I_{zz} - I_{xx}}{I_{yy}} pr + \frac{1}{I_{yy}} M_y \quad (40)$$

$$\begin{aligned}
\dot{r} &\triangleq f_r \\
&= \frac{I_{xx}I_{xx} - I_{xx}I_{yy} + I_{xz}I_{xz}}{I_{xx}I_{zz} - I_{xz}I_{xz}} pq \\
&\quad + \frac{I_{xz}(I_{xx} - I_{yy} + I_{zz})}{I_{xx}I_{zz} - I_{xz}I_{xz}} qr - \frac{I_{xz}}{I_{xx}I_{zz} - I_{xz}I_{xz}} M_x \\
&\quad + \frac{I_{xx}}{I_{xx}I_{zz} - I_{xz}I_{xz}} M_z,
\end{aligned} \tag{41}$$

where (X_E, Y_E) represents the position in a fixed flat earth frame, h the altitude, C_{ij} the elements of direction cosine matrix for transferring a fixed flat earth frame to the aircraft body moving frame, α the angle of attack, β the sideslip angle, ψ the heading angle, θ the pitch angle, ϕ the bank angle, p the roll rate, q the pitch rate, and r the yaw rate. Also, F_x , F_y , and F_z represent, respectively, three components of the total force in an aircraft body moving frame. The three force components are composed of the thrust T , the lift L , the drag D , the side force Y , and the weight mg (m is the aircraft mass and g the gravity acceleration) by the following equations:

$$\begin{aligned}
F_x &= T \cos \theta_T + L \sin \alpha - D \cos \alpha - mg \sin \theta \\
F_y &= Y + mg \sin \phi \cos \theta
\end{aligned} \tag{42}$$

$$F_z = -T \sin \theta_T - L \cos \alpha - D \sin \alpha + mg \cos \phi \cos \theta,$$

where θ_T is the angle between the thrust and the longitudinal axis. Moreover, M_x , M_y , and M_z represent the roll moment, the pitch moment, and the yaw moment, respectively, about the center of gravity, and I_{xx} , I_{yy} , I_{zz} , and I_{xz} are the components of the moment-of-inertia tensor. Furthermore, the wind disturbances on \dot{V}_a , $\dot{\alpha}$, and $\dot{\beta}$ are, respectively, represented by f_{V_w} , f_{α_w} , and f_{β_w} which are expressed as follows:

$$\begin{aligned}
f_{V_w} &= \frac{F_{x_w}}{m} \cos \alpha \cos \beta + \frac{F_{y_w}}{m} \sin \beta + \frac{F_{z_w}}{m} \sin \alpha \cos \beta \\
f_{\alpha_w} &= -\frac{F_{x_w}}{mV_a} \sin \alpha \sec \beta + \frac{F_{z_w}}{mV_a} \cos \alpha \sec \beta \\
f_{\beta_w} &= -\frac{F_{x_w}}{mV_a} \cos \alpha \sin \beta + \frac{F_{y_w}}{mV_a} \cos \beta \\
&\quad - \frac{F_{z_w}}{mV_a} \sin \alpha \sin \beta,
\end{aligned} \tag{43}$$

where

$$\begin{aligned}
F_{x_w} &= m(rv_g - qw_g - \dot{u}_g) \\
F_{y_w} &= m(-ru_g + pw_g - \dot{v}_g) \\
F_{z_w} &= m(qu_g - pv_g - \dot{w}_g).
\end{aligned} \tag{44}$$

In (44), F_{x_w} , F_{y_w} , and F_{z_w} are the three components of the force exerted by winds. At this point, it is worthy to mention that although there is no explicit term relating wind

TABLE 1: The geometric data, weight, and moment of inertia of the UAV.

	Parameter	Value	Units
Reference wing area	S	75	ft ²
Wing span	b	15	ft
Mean aerodynamic chord	\bar{c}	5.66	ft
Weight	W	2562.5	lb
x -axis inertia	I_{xx}	296.75	slug/ft ²
y -axis inertia	I_{yy}	1744.1875	slug/ft ²
z -axis inertia	I_{zz}	1971.875	slug/ft ²
x - z product of inertia	I_{xz}	30.6875	slug/ft ²

disturbances to \dot{p} , \dot{q} , and \dot{r} in (39)–(41), winds do have effects on V_a , α , and β through which M_x , M_y , and M_z are affected. Also, winds not only have explicit effects on \dot{V}_a , $\dot{\alpha}$, and $\dot{\beta}$ in (36)–(38), but also have implicit effects on them through L , Y , and D which obviously depend on V_a , α , and β .

To validate the method developed in this paper, a UAV as shown in Figure 1 is introduced. The parameters used for analysis are listed in Table 1.

The aerodynamic forces and moments of the UAV are computed by the following equations:

$$\begin{aligned}
L &= \bar{q}S \left[C_L(M, \alpha, \delta_e) + C_{L_q}(M, \alpha) \frac{\bar{c}q}{2V_a} \right] \\
D &= \bar{q}SC_D(M, \alpha, \delta_e) \\
Y &= \bar{q}S \left[C_{Y_\beta}(M, \alpha) \beta + C_{Y_p}(M, \alpha) \frac{bp}{2V_a} \right. \\
&\quad + C_{Y_r}(M, \alpha) \frac{br}{2V_a} + \Delta C_Y(M, \alpha, \delta_a) \\
&\quad \left. + \Delta C_Y(M, \alpha, \delta_r) \right] \\
M_x &= \bar{q}Sb \left[C_{\ell_\beta}(M, \alpha) \beta + C_{\ell_p}(M, \alpha) \frac{bp}{2V_a} \right. \\
&\quad + C_{\ell_r}(M, \alpha) \frac{br}{2V_a} + \Delta C_\ell(M, \alpha, \delta_a) \\
&\quad \left. + \Delta C_\ell(M, \alpha, \delta_r) \right] \\
M_y &= \bar{q}S\bar{c} \left[C_m(M, \alpha, \delta_e) + C_{m_q}(M, \alpha) \frac{\bar{c}q}{2V_a} \right] \\
M_z &= \bar{q}Sb \left[C_{n_\beta}(M, \alpha) \beta + C_{n_p}(M, \alpha) \frac{bp}{2V_a} \right. \\
&\quad + C_{n_r}(M, \alpha) \frac{br}{2V_a} + \Delta C_n(M, \alpha, \delta_a) \\
&\quad \left. + \Delta C_n(M, \alpha, \delta_r) \right],
\end{aligned} \tag{45}$$

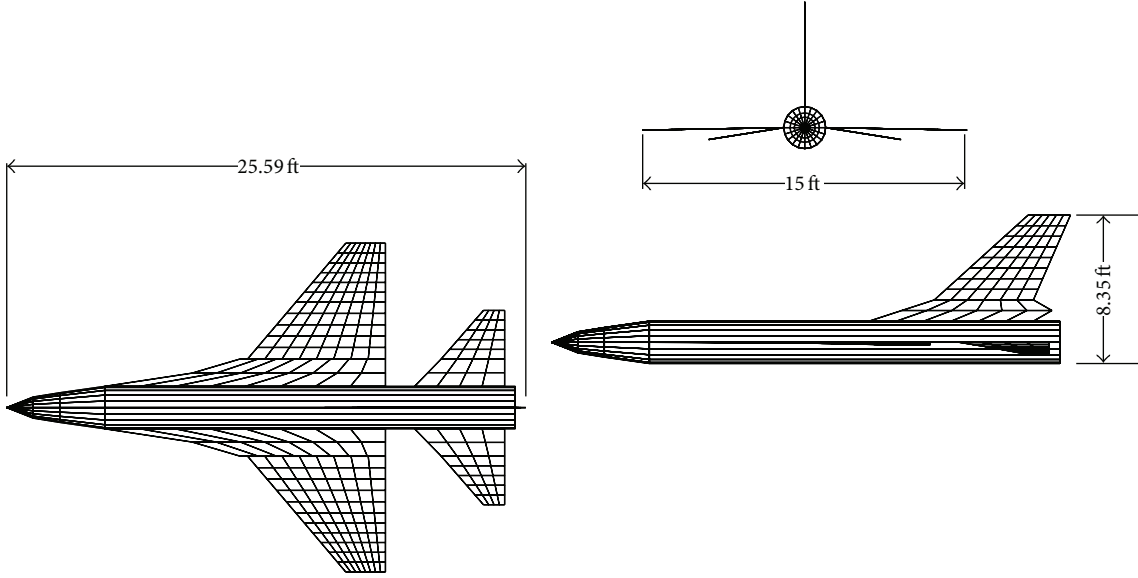


FIGURE 1: The configuration of the UAV.

where \bar{q} is the dynamic pressure and M the Mach number. The aerodynamic coefficients and stability derivatives are determined by a computer code dubbed "VORSTAB" [17]. The computation results are in the form of discrete data which are then interpolated as continuous functions in the computer program. These functions are represented by $C_L(M, \alpha, \delta_e)$, $C_{L_q}(M, \alpha)$, and so forth, in (45). The nonlinear database includes α ranging from -20° to 40° , δ_e from -24° to 24° , δ_a from -25° to 25° , and δ_r from -30° to 30° . Also, the flight conditions include altitude h ranging from sea level to 40,000 ft and Mach number M from 0.1 to 0.9 and from 1.1 to 1.9. The thrust model of the UAV is assumed as

$$\dot{T} = 0.1(T_c - T), \quad (46)$$

where T_c is the thrust command. The thrust is assumed to have a limitation, say $0 \leq T \leq 1.1W$.

3.2. Flight Control Design. In flight control design, only nominal flight dynamics models are considered since modeling uncertainties or wind disturbances, and so forth, cannot be determined in advance. Hence, all terms related to the wind disturbances in the flight dynamics equations of motion are neglected in this stage. For this flight dynamics model, the parameters involved in the control analysis are further elaborated as follows.

- (1) In the flight dynamics equations of motion, only h , θ , ϕ , V , α , β , p , q , and r are coupled. Therefore $\mathbf{x} \triangleq \{h, \theta, \phi, V, \alpha, \beta, p, q, r\}^T$, $\mathbf{f} = \{f_h, f_\theta, f_\phi, f_V, f_\alpha, f_\beta, f_p, f_q, f_r\}^T$, and $n = 9$.
- (2) The control input vector $\mathbf{u} \triangleq \{\delta_e, \delta_a, \delta_r, T_c\}^T$ and $m = 4$. According to the theory derived in the above, only 4 state variables can be controlled or designated. In this paper, the state vector to be controlled

$\mathbf{x}_c \triangleq \{h_c, V_c, \phi_c, \beta_c\}^T$ is chosen and the residual vector is $\mathbf{x}_r \triangleq \{\theta_r, \alpha_r, p_r, q_r, r_r\}^T$.

- (3) Determine the state variables and control variables for some trim conditions, on which the control design is based.
- (4) The model matrix in (3) is constructed as

$$\mathbf{\Omega} = \begin{bmatrix} \sigma_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & \sigma_4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_4 & 0 & \sigma_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_5 & 0 & 0 & \sigma_8 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_6 & 0 & 0 & \sigma_9 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_7 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_8 & 0 & 0 & \sigma_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_9 & 0 & 0 & \sigma_6 \end{bmatrix}, \quad (47)$$

where the arrangement for the pair (σ_2, σ_4) is enlightened from the phugoid mode in which θ and V are coupled. Similar arrangements are for the pairs (σ_5, σ_8) and (σ_6, σ_9) . In order to make the desired dynamics stable, the elements on the diagonal, $\sigma_1, \sigma_2, \sigma_3, \sigma_5, \sigma_6$, and σ_7 , must be positive. Also, those off-diagonal elements must satisfy the conditions, $\sigma_4^2 < \sigma_2^2$, $\sigma_8^2 < \sigma_5^2$, and $\sigma_9^2 < \sigma_6^2$. For convenience of later usage, these necessary conditions are dubbed "the stable model requirements."

In the design, the initial cruise flight conditions are $h = 600$ ft and $M = 0.25$. The trim conditions are $\alpha_{\text{trim}} = \theta_{\text{trim}} = 6.045^\circ$, $T_{\text{trim}} = 341.6$ lb, and $\delta_{e_{\text{trim}}} = -0.872^\circ$. Based on these data, the open-loop eigenvalues of $\mathbf{f}_x(\mathbf{x}_{\text{trim}}, \mathbf{u}_{\text{trim}})$ are determined and listed in Table 2.

TABLE 2: The open-loop eigenvalues of $\mathbf{f}_x(\mathbf{x}_{\text{trim}}, \mathbf{u}_{\text{trim}})$.

$\lambda_{1,2}$	$-1.124786 \pm 1.566875i$	λ_6	-3.129774
$\lambda_{3,4}$	$-0.008857 \pm 0.158058i$	$\lambda_{7,8}$	$-0.297956 \pm 0.410071i$
λ_5	-0.000133	λ_9	-0.023006

Longitudinal dynamics analysis reveals that $\lambda_{1,2}$ and $\lambda_{3,4}$ are closely associated with the short period mode and the phugoid mode, respectively, and lateral dynamics analysis shows that λ_6 , $\lambda_{7,8}$, and λ_9 are closely associated with the pure roll mode, the Dutch roll mode, and the spiral mode, respectively. Finally, the rest one, λ_5 , can be inferred to be associated with the altitude dynamics.

Intuitively, σ_i ($i = 1, 2, 3, \dots, 9$) $\in \sigma$ may just be arbitrarily assigned as long as they satisfy the stable model requirements. However, the resulted error dynamics in (9) may not just be stable. As will be shown in simulations, the desired eigenvalues must not only be able to make the system follow the desired dynamics, but also be able to generate a good set of σ which makes $e^{-\Omega t}$ decrease so quickly that the system is also robust if modeling uncertainties exist or wind disturbances are encountered. The choice of these eigenvalues is not only just based on the control analysis alone but must also be simultaneously based on simulations.

Note that, in this case, longitudinal dynamics analysis also reveals a right-half zero $z = 10.85$ in the h - δ_e transfer function. It can therefore be identified as a nonminimum phase problem. For this kind of problem, the desired eigenvalues must be very carefully assigned lest the resulting σ does not satisfy the stable model requirements. To determine σ , a two-way approach ($\sigma \rightleftharpoons \lambda$) is proposed as follows.

- (1) If the desired eigenvalues are equal to the open-loop eigenvalues as listed in Table 2, obviously, $\sigma = \mathbf{0}$. Choose small $\sigma (\neq \mathbf{0})$ which satisfies the stable model requirements.
- (2) Determine the eigenvalues λ_i ($i = 1, 2, 3, \dots, 9$) $\in \lambda$ of the error dynamics in (9).
- (3) If the error dynamics is not stable, modify λ so that it makes the error dynamics stable. Use the modified values as the desired eigenvalues.
- (4) Determine σ by following the 7 steps of iteration procedure described in Section 2.2.
- (5) If σ does not satisfy the stable model requirements, modify σ to make it satisfy. Go to step 2.
- (6) In each step of simulations, determine \mathbf{x}_r and \mathbf{u} from (4) by using the iteration method described in (5). In this step, only the nominal dynamics model is used.
- (7) Use \mathbf{u} in (1) for simulations. The equations of motion may include modeling uncertainties or wind disturbances.
- (8) If the simulation results are not satisfactory, modify σ and then go to step 2, or modify λ and then go to step 4.

TABLE 3: The designated eigenvalues.

$\lambda_{1,2}$	$-4.22180 \pm 1.57684i$	λ_7	-1.75509
$\lambda_{3,4}$	$-0.112728 \pm 0.0902199i$	λ_8	-0.920091
λ_5	-0.0204641	λ_9	-0.146833
λ_6	-4.73168		

TABLE 4: The resulting parameters for desired dynamics.

σ_1	0.0201234	σ_6	1.04305
σ_2	0.101328	σ_7	1.60654
σ_3	0.112353	σ_8	0.00193758
σ_4	-0.0000325585	σ_9	0.295443
σ_5	3.09966		

By using these 8 steps of procedure, the designated eigenvalues and the resulting parameters for desired dynamics are obtained and listed in Tables 3 and 4, respectively.

These parameters indeed satisfy the stable model requirements. Note that initially σ_5 is small and $\lambda_{1,2}$ and $\lambda_{3,4}$ are not so deviated from their counterparts of open-loop eigenvalues. However, the simulation results show that the performance in tracking h_c is not good enough. Enlarging σ_5 does improve the performance but makes $\lambda_{1,2}$ and $\lambda_{3,4}$ deviate their counterparts of open-loop eigenvalues a lot. Also note that the large deviations of λ_6 , $\lambda_{7,8}$, and λ_9 from their counterparts of open-loop eigenvalues are due to the iteration procedure in determining a set of σ_6 , σ_7 , σ_8 , and σ_9 for satisfying the stable model requirements.

3.3. Flight Simulations. In simulations, the initial conditions are $(X_E, Y_E) = (0, 0)$, $h = 600$ ft, $\psi = 0^\circ$, $\theta = \theta_{\text{trim}} = 6.04^\circ$, $\phi = 0^\circ$, $M = 0.25$, $\alpha = \alpha_{\text{trim}} = 6.04^\circ$, $\beta = 0^\circ$, $p = 0$, $q = 0$, and $r = 0$. The states to be tracked are $h_c = 2,000$ ft, $\psi_c = 90^\circ$, $M_c = 0.40$, and $\beta_c = 0^\circ$. Although the heading angle is to be tracked, for pilots, it would make more sense to regulate the bank angle rather than the heading angle by assuming

$$\phi_c = k_\phi (\psi_c - \psi), \quad (48)$$

where k_ϕ is a constant parameter. In this study, $k_\phi = 0.12$.

At each instant, h_c , V_c , ϕ_c , and β_c are given, the controls δ_e , δ_a , δ_r , and T_c along with the residual state variables α_r , θ_r , p_r , q_r , and r_r can be determined from (4) by using the iteration method described in (5). Note that, in the computation of the control, only nominal flight dynamics equations of motion are used since modeling uncertainties or wind disturbances are not known. Also note that, in using the Newton-Raphson method, the convergence can be guaranteed if the initial guess is sufficiently close to the solution. Since the solution changes only very little in each step of integration, numerical practices show that it takes only 4 iterations to converge to within 0.0001% of the correct value if the solution in the previous step of integration is taken as an initial guess. In the first step of integration, it may be necessary to take a few more iterations, say, 10 iterations, since the solution is different

from that in the trim conditions, which is usually taken as an initial guess.

For illustration, simulation block diagrams including \mathbf{x} , \mathbf{x}_d , \mathbf{x}_c , \mathbf{x}_r , and \mathbf{u} are shown in Figure 2.

After the controls δ_e , δ_a , δ_r , and T_c are determined, (30)–(41) are used for simulations. In order to ascertain if the control system is robust or not, a fictitious wind model is put enroute as follows:

$$V_{w_x} = \begin{cases} -\frac{30 \times 6080}{3600} \sin\left(\frac{2\pi(X_E - (X_{wo} - X_{scale}))}{2X_{scale}}\right) & \text{if } |X_E - X_{wo}| < X_{scale}, \\ 0 & \text{else.} \end{cases} \quad (49)$$

$$V_{w_y} = \begin{cases} \frac{30 \times 6080}{3600} \sin\left(\frac{2\pi(Y_E - (Y_{wo} - Y_{scale}))}{2Y_{scale}}\right) & \text{if } |Y_E - Y_{wo}| < Y_{scale}, \\ 0 & \text{else.} \end{cases} \quad (50)$$

$$V_h = \begin{cases} \frac{10 \times 6080}{3600} \sin\left(\frac{2\pi(h - (H_{wo} - H_{scale}))}{2H_{scale}}\right) & \text{if } |h - H_{wo}| < H_{scale}, \\ 0 & \text{else.} \end{cases} \quad (51)$$

$$\dot{V}_{w_x} = \begin{cases} -\frac{30 \times 6080}{3600} \times \frac{2\pi\dot{X}_E}{2X_{scale}} \cos\left(\frac{2\pi(X_E - (X_{wo} - X_{scale}))}{2X_{scale}}\right) & \text{if } |X_E - X_{wo}| < X_{scale}, \\ 0 & \text{else.} \end{cases} \quad (52)$$

$$\dot{V}_{w_y} = \begin{cases} \frac{30 \times 6080}{3600} \times \frac{2\pi\dot{Y}_E}{2Y_{scale}} \cos\left(\frac{2\pi(Y_E - (Y_{wo} - Y_{scale}))}{2Y_{scale}}\right) & \text{if } |Y_E - Y_{wo}| < Y_{scale}, \\ 0 & \text{else.} \end{cases} \quad (53)$$

$$\dot{V}_h = \begin{cases} \frac{10 \times 6080}{3600} \times \frac{2\pi\dot{h}}{2H_{scale}} \cos\left(\frac{2\pi(h - (H_{wo} - H_{scale}))}{2H_{scale}}\right) & \text{if } |h - H_{wo}| < H_{scale}, \\ 0 & \text{else,} \end{cases} \quad (54)$$

where $X_{wo} = 29,727$ ft, $Y_{wo} = 10,180$ ft, and $H_{wo} = 1,000$ ft represent the center position of wind zone, and $X_{scale} = 5,000$ ft, $Y_{scale} = 5,000$ ft, and $H_{scale} = 1,000$ ft represent the maximum range of wind zone from its center. From (49)–(51), it is also known that the three maximum wind velocity components in the earth fixed frame are 30 knots, 30 knots, and 10 knots, respectively.

In this study, two cases of simulations are conducted; one is without wind disturbances and another with wind disturbances. The results are presented in Figures 3–6.

As shown in Figure 3(a), there is no much difference in ground trajectories whether wind disturbances are present or not. While flying, the UAV first suffers vertical wind disturbances in between $t = 15.10$ sec and 32.04 sec and then horizontal wind disturbances in between $t = 50.60$ sec and 90.00 sec as shown in Figures 3(b) and 3(c).

The state variables to be tracked are all plotted in Figures 4(a)–4(e). From Figure 4(a), it is found that h decreases initially when $h_c = 2,000$ ft is commanded. A careful study reveals that the elevator angle computed is $\delta_e = -0.733^\circ$, which is not enough to hold the UAV level as $\delta_{e_{trim}} = -0.872^\circ$

is required. The lowest altitude reached is 578 ft with $\sigma_5 = 3.09966$ in Table 4 being used. Increasing σ_5 will increase the lowest altitude reached but at the expense of increasing overshoot. Fortunately, after descending to 578 ft, the UAV begins to climb. During its climb, the UAV encounters an ascending wind and then a descending wind. But the influences on the altitude are almost negligible. In contrary, the horizontal wind seems to have more influences as it makes the altitude fluctuate. Whether wind disturbances are present or not, the UAV can always approach the commanded altitude asymptotically.

As shown in Figure 4(b), the Mach number does not seem to be so affected by the vertical wind as does by the horizontal wind. In this case, the horizontal wind causes the Mach number to fluctuate up to $\pm 10\%$. The sharp angles shown in the figure are due to the discontinuity of wind acceleration at the edges of wind zone as shown in Figure 3(c).

When the UAV is commanded to make a 90° turn, its heading angle gradually increases and asymptotically approaches the command as shown in Figure 4(c) whether winds are present or not. Figure 4(d) shows that the bank

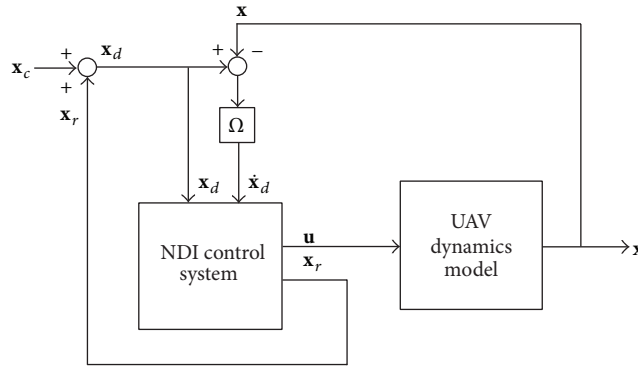


FIGURE 2: Simulation block diagrams for the NDI flight control system.

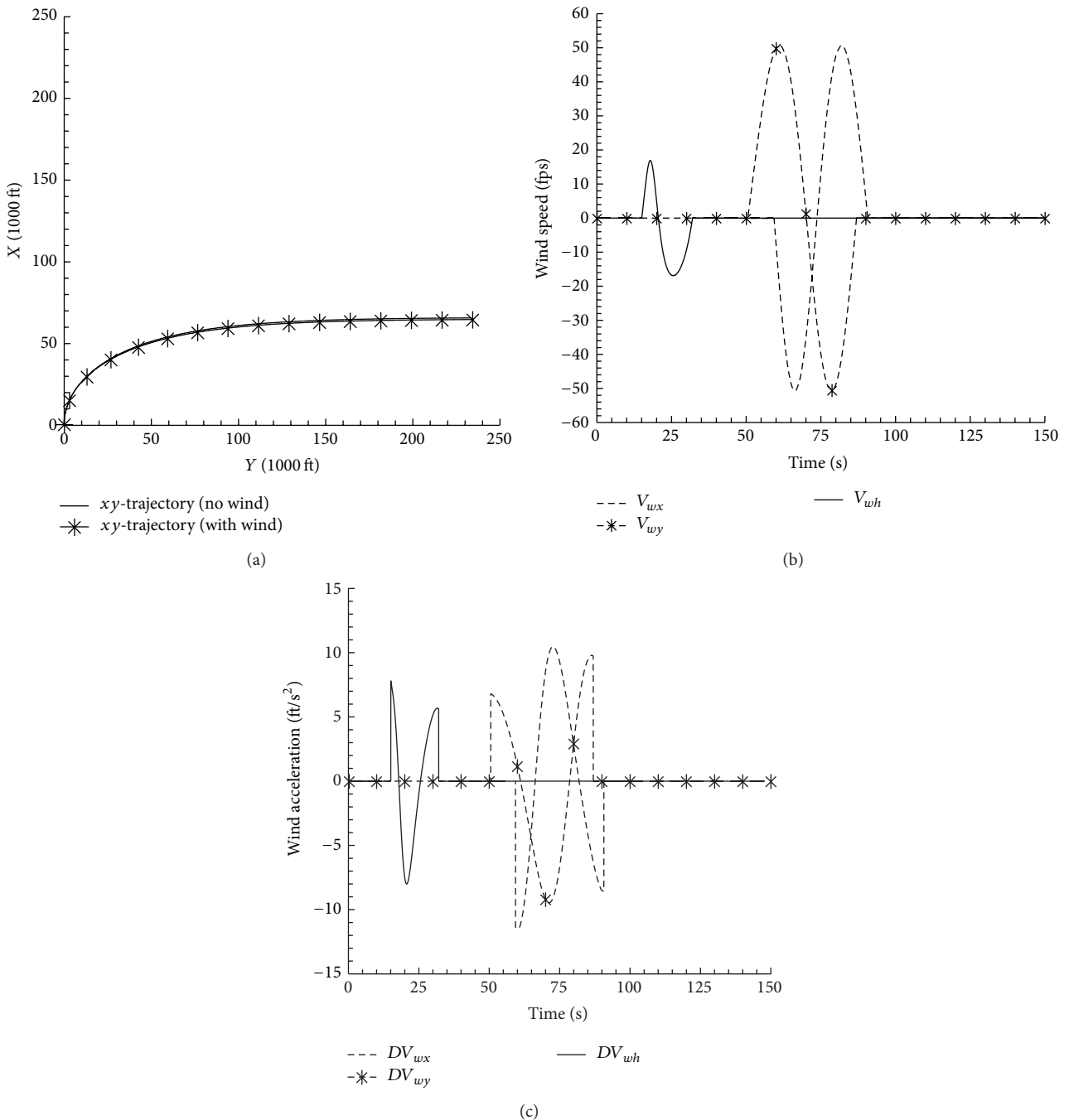


FIGURE 3: (a) Trajectory in X-Y plane, (b) wind speed, and (c) wind acceleration.

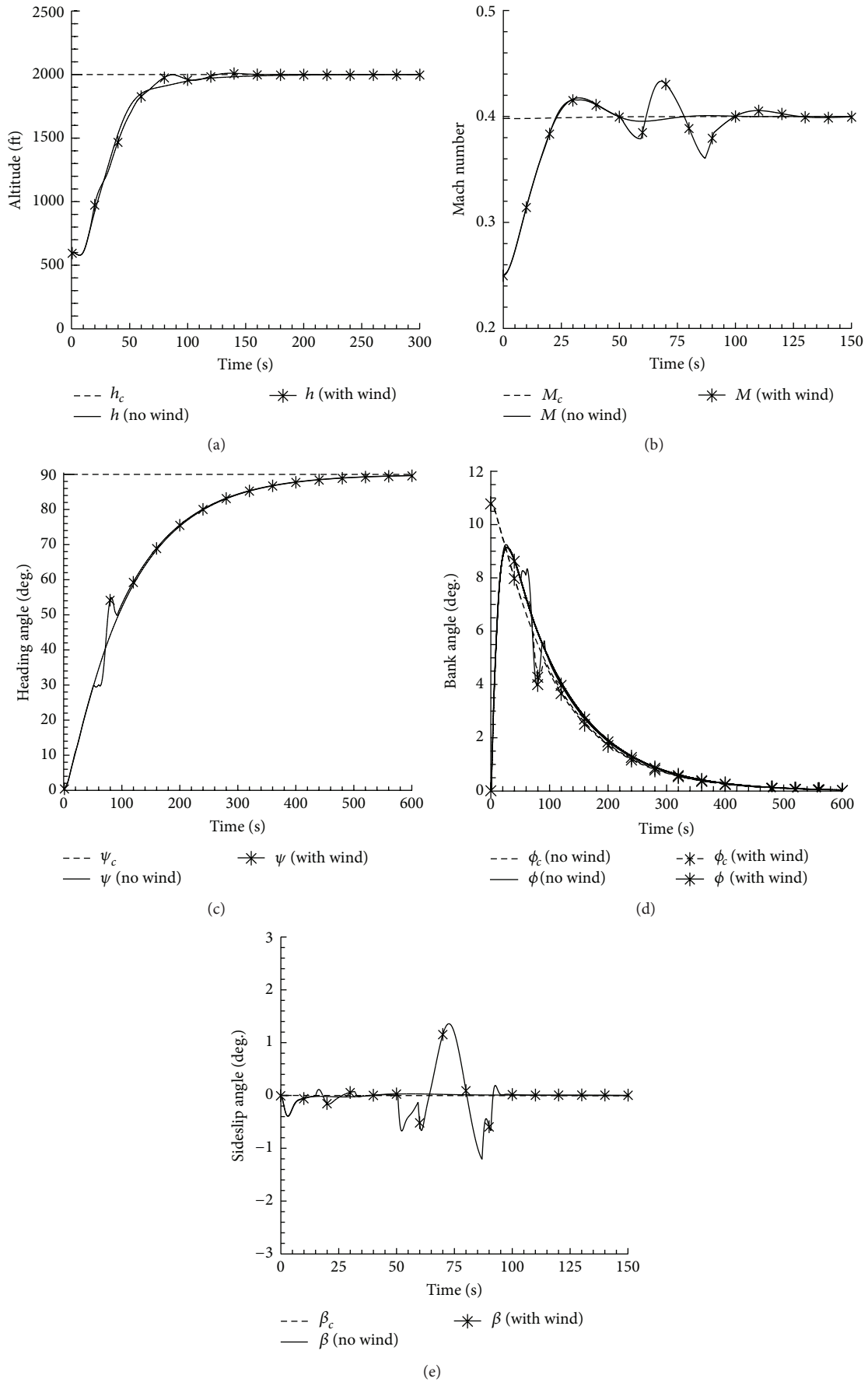


FIGURE 4: The responses of (a) altitude, (b) Mach number, (c) heading angle, (d) bank angle, and (e) sideslip angle.

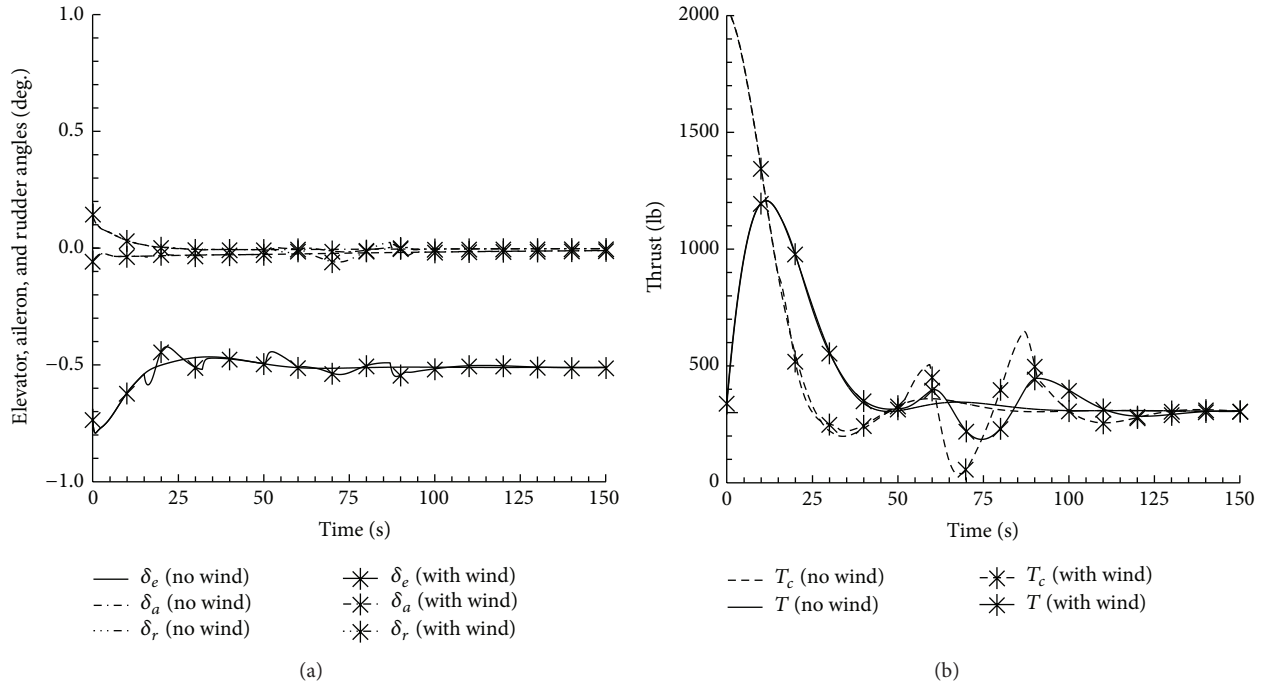


FIGURE 5: The responses of (a) control surface deflections and (b) thrust.

angle is closely associated with the heading angle as the bank angle is computed based on (48). Although the wind disturbances do not explicitly affect $\dot{\phi}$ in (35), they do implicitly affect it through α and β which in turn affect the aerodynamic moments. Fluctuations in the bank angle and the heading angle are remarkable when the UAV passes through the wind zone but are still tolerable.

Sideslip angle seems to be affected by winds more heavily as shown in Figure 4(e). Fortunately, the control system is robust enough as being able to keep it small.

As shown in Figure 5(a), all control surface deflection angles remain small. The elevator needs only to deflect slightly to rotate the UAV to climb. As mentioned early, $\delta_e = -0.733^\circ$ at the instant when h_c and M_c are given. When the UAV reaches the commanded state conditions, the elevator trim angle approaches -0.510° . Small deflection in aileron angle is enough to make the UAV bank turn and the rudder just keeps very small as does the sideslip angle. All control surface deflection angles are not remarkably affected as the UAV passes through the wind zone.

As shown in Figure 5(b), the thrust command increases very sharply as a demand to increase both the altitude and the Mach number simultaneously is given. Also, when the UAV passes the wind zone, the thrust command fluctuates very violently. Obviously, the thrust is very closely interacted with the Mach number. A low pass filter with time constant 10 sec alleviates the sharp and violent responses a lot for the actual thrust at the expense of delaying its reaction time.

In Figures 6(a)–6(e), responses of all residual state variables along with the computed commands for them are presented. In fact, these commands are generated in the inner system, not input from outer designations. It is interesting

to observe how the actual state variables track the commands.

In Figure 6(a), it is observed that, in comparison to $\alpha_{\text{trim}} = 6.04^\circ$, the computed command $\alpha_r = 4.85^\circ$ at the instant when h_c and M_c are given. Then the angle of attack α follows α_r closely without apparent short-period mode oscillations when the designated short-period mode eigenvalues are as high as $\lambda_{1,2} = -4.22180 \pm 1.57684i$. Winds do have remark effect on both α_r and α . Finally, both approach a new value, $\alpha_{\text{trim}} = 2.88^\circ$, for flight conditions at higher altitude and faster speed.

In Figures 6(b) and 6(c), it is observed that although both θ_r and q_r increase in response to the requirement for increasing the cruise altitude, θ and q decrease remarkably. As revealed in explaining h response, the elevator angle computed is not enough to hold $\dot{q} = 0$ and therefore $\dot{q} < 0$ which in turn makes $\dot{\theta} < 0$. The minimum pitch rate is $q = -0.717 \text{ deg./s}$ at $t = 0.78 \text{ sec}$. In this case, the vertical wind does have more remarkable effects on θ and q than the horizontal wind. Finally, both approach $\theta_{\text{trim}} = \alpha_{\text{trim}} = 2.88^\circ$ and $q_{\text{trim}} = 0$, respectively, for the new flight conditions.

It is very interesting to find that, in Figures 6(d) and 6(e), the shapes of p and r responses resemble, respectively, those of ϕ and β responses as shown in Figures 4(d) and 4(e). The effects of horizontal wind on both responses are more remarkable than those of vertical wind in this case. The decay of r seems to be very slow.

4. Conclusions

In this paper, a unified nonlinear dynamic inversion control system is successfully developed. With this method, the

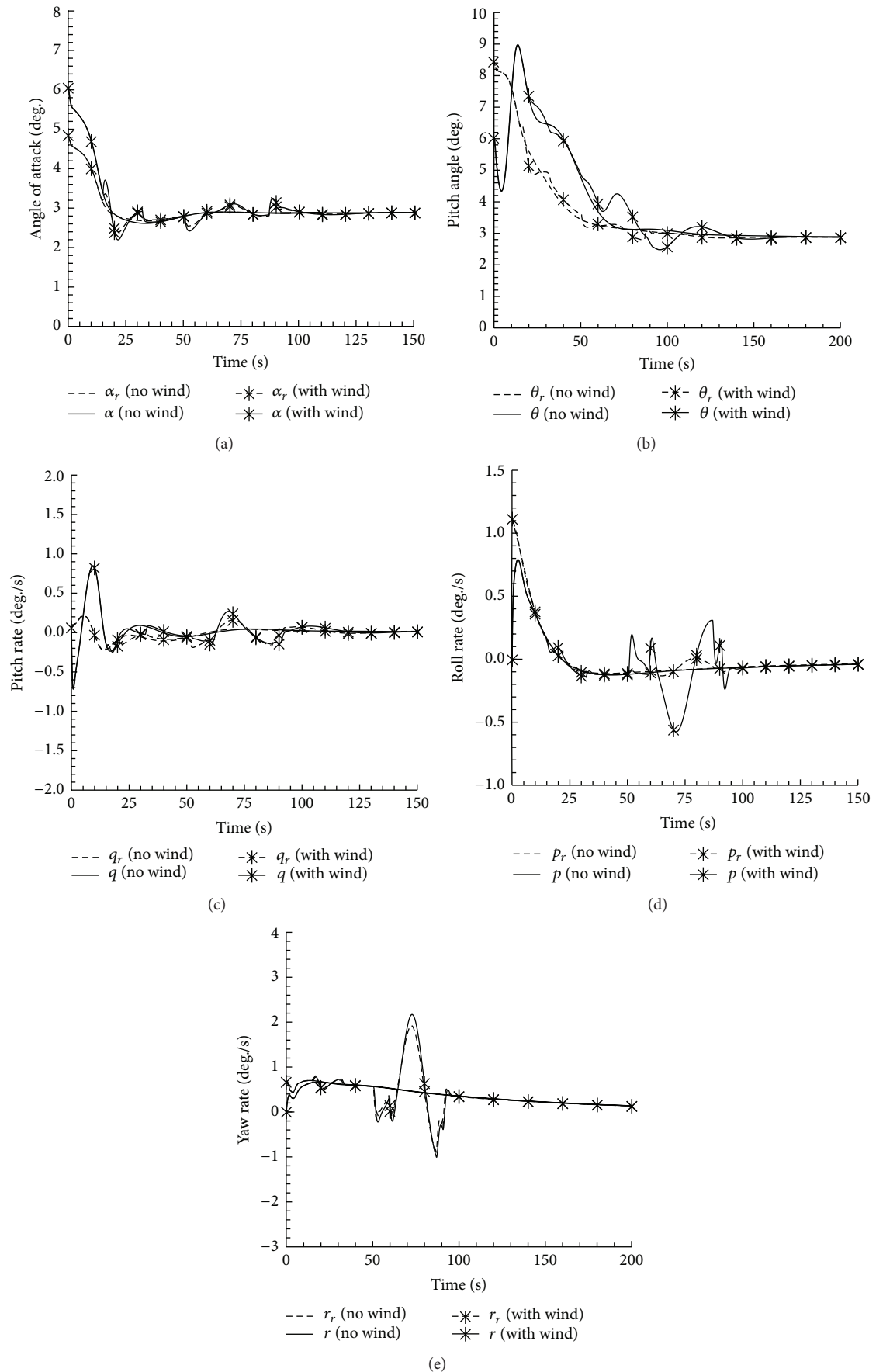


FIGURE 6: The responses of (a) angle of attack, (b) pitch angle, (c) pitch rate, (d) roll rate, and (e) yaw rate.

parameters for desired dynamics can be determined with a set of assigned eigenvalues for error dynamics. The control system has been proved to be robust to modeling uncertainties or wind disturbances. A nonaffine nonlinear flight dynamics system with right-half zero has been used as an example for the control design. In the design process, it is not necessary to use the feedback linearization to transform the nonlinear equations into linear equations. Numerical simulations of the control system show that the desired state variables can be successfully tracked whether winds are present or not.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors would like to express their gratitude to Chuan-Tau Edward Lan for his helps in computing the aerodynamic data of the UAV used as an example for the control design demonstrated in this paper.

References

- [1] M. R. Mendenhall, S. C. Perkins, M. Tomac, A. Rizzi, and R. K. Nangia, "Comparing and benchmarking engineering methods for the prediction of X-31 aerodynamics," *Aerospace Science and Technology*, vol. 20, no. 1, pp. 12–20, 2012.
- [2] M. L. Ireland, A. Vargas, and D. Anderson, "A comparison of closed-loop performance of multirotor configurations using non-linear dynamic inversion control," *Aerospace*, vol. 2, no. 2, pp. 325–352, 2015.
- [3] J. O. Pedro, A. Panday, and L. Dala, "A nonlinear dynamic inversion-based neurocontroller for unmanned combat aerial vehicles during aerial refuelling," *International Journal of Applied Mathematics and Computer Science*, vol. 23, no. 1, pp. 75–90, 2013.
- [4] N. Hovakimyan, E. Lavretsky, and A. Sasane, "Dynamic inversion for nonaffine-in-control systems via time-scale separation. Part I," *Journal of Dynamical and Control Systems*, vol. 13, no. 4, pp. 451–465, 2007.
- [5] Z. Xie, Y. Xia, and M. Fu, "Robust trajectory-tracking method for UAV using nonlinear dynamic inversion," in *Proceedings of the IEEE 5th International Conference on Cybernetics and Intelligent Systems (CIS '11)*, pp. 93–98, Qingdao, China, September 2011.
- [6] G. Gao, J. Wang, and X. Wang, "Adaptive fault-tolerant control for feedback linearizable systems with an aircraft application," *International Journal of Robust and Nonlinear Control*, vol. 25, no. 9, pp. 1301–1326, 2015.
- [7] H. N. Foghahaayee, M. B. Menhaj, and H. A. Talebi, "Weakly and strongly non-minimum phase systems: properties and limitations," *International Journal of Control*, pp. 1–16, 2015.
- [8] L. Fiorentini and A. Serrani, "Adaptive restricted trajectory tracking for a non-minimum phase hypersonic vehicle model," *Automatica*, vol. 48, no. 7, pp. 1248–1261, 2012.
- [9] A. R. Babaei, M. Mortazavi, and M. H. Moradi, "Fuzzy sliding mode autopilot design for nonminimum phase and nonlinear UAV," *Journal of Intelligent and Fuzzy Systems*, vol. 24, no. 3, pp. 499–509, 2013.
- [10] N. Dadkhah and B. Mettler, "Control system design and evaluation for robust autonomous rotorcraft guidance," *Control Engineering Practice*, vol. 21, no. 11, pp. 1488–1506, 2013.
- [11] Z. Lang and A. Wu, "Study on dual-loop controller of helicopter based on the robust hinfinite loop shaping method," *Applied Mechanics and Materials*, vol. 130–134, pp. 1182–1185, 2012.
- [12] S. Sieberling, Q. P. Chu, and J. A. Mulder, "Robust flight control using incremental nonlinear dynamic inversion and angular acceleration prediction," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 6, pp. 1732–1742, 2010.
- [13] A. Rahideh, A. H. Bajodah, and M. H. Shaheed, "Real time adaptive nonlinear model inversion control of a twin rotor MIMO system using neural networks," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 6, pp. 1289–1297, 2012.
- [14] J. Georgie and J. Valasek, "Evaluation of longitudinal desired dynamics for dynamic-inversion controlled generic reentry vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 5, pp. 811–819, 2003.
- [15] G. A. Smith and G. Meyer, "Aircraft automatic flight control system with model inversion," *Journal of Guidance, Control, and Dynamics*, vol. 10, no. 3, pp. 269–275, 1987.
- [16] I. Yang, D. Lee, and D. S. Han, "Designing a robust nonlinear dynamic inversion controller for spacecraft formation flying," *Mathematical Problems in Engineering*, vol. 2014, Article ID 471352, 12 pages, 2014.
- [17] C. T. E. Lan, "VORSTAB, a computer program for calculating lateral directional stability derivatives with vortex flow effect," NASA CR-172501, NASA, 1985.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

