

Flight-Path Angle Control via Neuro-Adaptive Backstepping

Manu Sharma* and David G. Ward†

Barron Associates, Inc.
1160 Pepsi Place Suite 300
Charlottesville, VA 22902

Abstract

A method of controlling flight-path angle via neuro-adaptive backstepping is presented. Backstepping is used to circumvent the matching condition, which can restrict feedback linearizing methods. Additionally, an on-line multilayer neural network is used to provide robustness to aerodynamic uncertainties in the plant model. The network also simplifies the backstepping design because it obviates the need to construct a regressor, and does not require an a-priori basis. Numerical simulation results show the viability of this approach.

Introduction

Various forms of nonlinear control have emerged as enabling technologies for control of advanced flight vehicles.¹⁻³ Offering both increases in performance as well as reduced development times by dealing with the complete dynamics of the vehicle rather than point designs, nonlinear control tools provide a great deal of design flexibility. Feedback linearization, in its various forms, is perhaps the most commonly employed nonlinear control method in this arena. However, the majority of the literature on feedback linearizing controllers treats some combination of the aircraft's angular rates (p, q, r), aerodynamic angles (α, β), or Euler angles (ϕ, θ, ψ) as the command variables. Using time-scale separation arguments, the plant dynamics are partitioned into slow states and fast states, with the fast states used as virtual controls for the slow.⁴ Another restriction on feedback linearization is the matching condition, which requires that all parametric plant uncertainties appear in the same equation of a state-space representation as the control.^{5,6} This is of significance in flight control where the plant aerodynamics always contain some degree of uncertainty.

A nonlinear control method that offers relief from these issues is backstepping: a Lyapunov based method applicable to systems in various cascaded forms. Backstepping makes use of a recursive procedure that breaks down the control problem for the full system into a sequence of designs for lower order systems.^{6,7} Unmatched terms are then dealt with at each stage of the recursive process. Although traditional backstepping can be applied only to plants with unmatched *nonlinearities*, adaptive backstepping deals with unmatched parametric *uncertainties*. Backstepping also relaxes time-scale separation requirements by including transients in the virtual controls explicitly in the control formulation. Other strengths of backstepping lie in both providing the designer with control Lyapunov functions as the design progresses, as well as in allowing the designer to discriminate between which nonlinearities cancel, and which plant dynamics to exploit.⁷ This is an advantage over feedback linearization which attempts to cancel all plant nonlinearities indiscriminately.

A comprehensive study of the early work in backstepping and adaptive backstepping is given by Krstic et al.⁵ Extensions of adaptive backstepping to use neural networks (NNs) using indirect-adaptive formulations are given by Polycarpou.^{8,9} Lee and Tomizuka construct a similar method that uses fuzzy logic,¹⁰ and other extensions of adaptive backstepping using NNs include.^{11,12}

One criticism of some approaches to adaptive backstepping is that the number of terms involved in computing the control balloon with each backward step because terms that must be computed via differentiation are introduced with each step. To alleviate this burden, Kwan and Lewis lump some of the known, but difficult to determine, terms with the uncertainties and rely on a two-layer NN that is *linear-in-the-network*-parameters while being *non-linear-in-the-system*-parameters to compensate for them.^{13,14} This greatly simplifies the analysis required to construct a backstepping controller, and does not require that the uncertainties be linearly

*Research Scientist, Member AIAA. sharma@bainet.com

†Senior Research Scientist, Member AIAA.
ward@bainet.com

Copyright © 2002 by Barron Associates, Inc. Published by the American Institute of Aeronautics and Astronautics, Inc. with permission.

parameterizable as does traditional adaptive backstepping. Furthermore, this method circumvents the need to compute regressors and especially their derivatives, thereby greatly simplifying the control computation for each back-step. Sharma and Calise extend this work to use a class of *nonlinear-in-the-parameters* neural network.^{15,16} An advantage of the nonlinear network is that it does not require the designer to provide basis functions a priori; although the designer still needs to provide inputs to the NN that parameterize the uncertainties. In addition, a single nonlinear network with as many outputs as needed can be used in place of individual linear networks, making the nonlinear network easily scalable in application.

Backstepping is particularly attractive for flight control where aerodynamic parameters are often in error, and various control variables contain unmatched uncertainties. There have, however, been only limited applications of backstepping concepts to flight control. Furthermore, the majority of the literature deals with the control of aerodynamic angles (α, β) and/or Euler angles (ϕ, θ, ψ), which are only one integrator removed from the body angular rates.^{17–20} Reference 20 is the only work known that also addresses backstepping control of flight-path angle (γ), but it does so without adaptation.

This paper extends that in Ref 21 by utilizing the adaptive backstepping theory developed in Ref 15,16 to provide robustness to parametric uncertainties. This also provides an extension to Ref 19, which presents a neuro-adaptive backstepping approach for flight control of (α, β, ϕ) but does not address the unmatched uncertainties in lift and sideforce (i.e. uncertainties in α and β dynamics).

This paper is organized as follows. The flight-path angle dynamics are first presented. For simplicity, only the pitch-plane motion of the aircraft is considered. Next, the dynamics are cast into a strict feedback form for backstepping. There are two options: one of which uses the pitch Euler angle as a virtual control, and the other uses angle of attack. The former was originally developed in Ref 21. Both options are considered here, and are shown to be equivalent for flight in the pitch-plane. Subsequently, a general procedure for adaptive backstepping is summarized, and is followed by an overview of the multilayered neural used in this work. Finally, numerical results are presented from a nonlinear 3-DOF simulation of a UAV.

System Definition

Backstepping takes advantage of the idea that certain states of a system can be used as virtual controls

to stabilize a portion of the overall system. However, one of the restrictions of the backstepping approach is that the system must be in a “cascade” form such as strict and pure feedback forms.⁵ Begin, with the definition of flight path angle of an aircraft

$$\gamma = \sin^{-1} \left(\frac{\dot{h}}{V_t} \right), \quad (1)$$

where \dot{h} is the rate of change in altitude, and V_t is the aircraft’s airspeed. The time rate-of-change of γ is given by

$$\begin{aligned} \dot{\gamma} = \frac{1}{mV_t} [-D \sin \beta \sin \mu - Y \sin \mu \cos \beta + L \cos \mu \\ + F_T (\cos \mu \sin \alpha + \sin \mu \sin \beta \cos \alpha) - mg \cos \gamma] \end{aligned} \quad (2)$$

where L , D , and Y are lift, drag and side-force; α , β , and μ are angle-of-attack, angle-of-sideslip, and stability-axis roll angle; and m and g are the aircraft’s mass and gravity respectively.²²

For simplicity, the aircraft motion is restricted to the pitch-plane, i.e. $Y = \beta = \mu = 0$, which reduces (1-2) to

$$\begin{aligned} \gamma &= \theta - \alpha \\ \dot{\gamma} &= \dot{\theta} - \dot{\alpha}. \end{aligned} \quad (3)$$

The pitch-plane α and θ dynamics are given by^{22,23}

$$\dot{\alpha} = \frac{1}{mV_T} [-F_T \sin \alpha - L + mg \cos \gamma] + q \quad (4)$$

$$\dot{\theta} = q. \quad (5)$$

The thrust term, $F_t \sin \alpha$, in (4) may be neglected as it is generally much smaller than lift. Furthermore, let the lift term be expressed as a sum of two terms

$$L = L_o + L_\alpha \alpha, \quad (6)$$

where $L_\alpha = L_\alpha(\alpha, M, h, V_t)$ represents the lift curve slope, and $L_o = L_o(\alpha, M, h, V_t, q)$ represents all of the other contributions to lift, such as Mach number, pitch rate, etc. The lift contribution of control surfaces is intentionally neglected, as is commonly done, since it is generally much smaller than the moment contribution. Furthermore, its inclusion would prevent the transforming the γ dynamics into strict-feedback form. Although the L_α term is a linear approximation of the lift contribution of α , there is no loss of generality since L_α represents a *locally* linear approximation of the lift curve slope. Thus, the

simplified aircraft dynamics become

$$\dot{\gamma} = -\frac{g}{V_T} \cos \gamma + L'_o + L'_\alpha \alpha \quad (7a)$$

$$\dot{\alpha} = \frac{g}{V_T} \cos \gamma - L'_o - L'_\alpha \alpha + q \quad (7b)$$

$$\dot{\theta} = q \quad (7c)$$

$$\dot{q} = M_o + M_\delta \delta \quad (7d)$$

where $L'_o = \frac{L_o}{mV_T}$, $L'_\alpha = \frac{L_\alpha}{mV_T}$, and δ represents control surface deflections, M_δ represents the control pitching moment, and $M_o = M_o(\alpha, M, h, V_t, q)$ represents moment contributions from all other sources such as α , Mach number, etc. M_o is often approximated by $M_o = M_\alpha \alpha + M_q q$.

Strict-Feedback Form

Inspection of the dynamics in (7) reveals that the dynamics of the triple (γ, α, q) is in strict feedback form. In addition, the dynamics for (γ, θ, q) can also be expressed in strict-feedback form with α in (7a) expressed in terms of γ and θ as defined in (3).

$$\dot{\gamma} = -\frac{g}{V_T} \cos \gamma - L'_\alpha \gamma + L'_o + L'_\alpha \theta$$

Since these two methods are equivalent, they are both studied in this paper; the latter was first developed in Ref 21.

Option I: α as an Intermediate Control Variable

First consider the use of α as the intermediate state to control γ . For notational simplification, let

$$x_1 \triangleq \frac{1}{L'_\alpha} \gamma \quad (8a)$$

$$x_2 \triangleq \alpha \quad (8b)$$

$$x_3 \triangleq q \quad (8c)$$

$$u \triangleq \delta, \quad (8d)$$

and

$$\dot{x}_1 = f_1(x_1) + x_2 \quad (9a)$$

$$\dot{x}_2 = f_2(x_1, x_2) + x_3 \quad (9b)$$

$$\dot{x}_3 = f_3(x_2, x_3) + g_3(x_2) u, \quad (9c)$$

where

$$f_1(x_1) = \frac{L_o}{L_\alpha} - \frac{mg}{L_\alpha} \cos(L'_\alpha x_1) \quad (10a)$$

$$f_2(x_1, x_2) = -L'_o + \frac{g}{V_T} \cos(L'_\alpha x_1) - L'_\alpha x_2 \quad (10b)$$

$$f_3(x_2, x_3) = M_\alpha x_2 + M_q x_3 \quad (10c)$$

$$g_3(x_2) = M_\delta. \quad (10d)$$

The aerodynamic parameters of an aircraft are rarely, if ever, known exactly but approximations to them are certainly available. Then, let \hat{f}_i and \hat{g} represent approximations of f_i and g respectively, and define $\delta'_i = \hat{f}_i - f_i$. With appropriate substitutions this yields

$$\dot{x}_1 = \hat{f}_1(x_1) - \delta'_1(x_1) + x_2 \quad (11a)$$

$$\dot{x}_2 = \hat{f}_2(x_1, x_2) - \delta'_2(x_1, x_2) + x_3 \quad (11b)$$

$$\dot{x}_3 = \hat{f}_3(x_2, x_3) - \delta'_1(x_2, x_3, u) + \hat{g}_3(x_2) u. \quad (11c)$$

Equation (11) is in now in a form to which backstepping may be applied. The details of this are presented in next section

Option II: θ as an Intermediate Control Variable

Now consider the use of θ as an intermediate state to control γ . Returning to the γ dynamics, the relevant equations in (7) can be written as

$$\dot{\gamma} = -\frac{g}{V_T} \cos \gamma + L'_o + L'_\alpha (\theta - \gamma) \quad (12a)$$

$$\dot{\theta} = q \quad (12b)$$

$$\dot{q} = M_\alpha \alpha + M_q q + M_\delta \delta \quad (12c)$$

As before, redefine the states as

$$x_1 \triangleq \frac{1}{L'_\alpha} \gamma \quad (13a)$$

$$x_2 \triangleq \theta \quad (13b)$$

$$x_3 \triangleq q \quad (13c)$$

$$u \triangleq \delta. \quad (13d)$$

Then,

$$\dot{x}_1 = f_1(x_1) + x_2 \quad (14a)$$

$$\dot{x}_2 = f_2 + x_3 \quad (14b)$$

$$\dot{x}_3 = f_3(x_1, x_2, x_3) + g_3(x_1, x_2) u, \quad (14c)$$

where

$$f_1(x_1) = \frac{L_o}{L_\alpha} - \frac{mg}{L_\alpha} \cos(L'_\alpha x_1) - L'_\alpha x_1 \quad (15a)$$

$$f_2 = 0 \quad (15b)$$

$$f_3(x_1, x_2, x_3) = -M_\alpha L'_\alpha x_1 + M_\alpha x_2 + M_q x_3 \quad (15c)$$

$$g_3(x_1, x_2) = M_\delta. \quad (15d)$$

Again, including terms representing parametric uncertainty yields

$$\dot{x}_1 = \hat{f}_1(x_1) - \delta'_1(x_1) + x_2 \quad (16a)$$

$$\dot{x}_2 = x_3 \quad (16b)$$

$$\dot{x}_3 = \hat{f}_3(x_1, x_2, x_3) - \delta'_1(x_1, x_2, x_3, u) + \hat{g}_3(x_1, x_2) u. \quad (16c)$$

As in the previous section, (16) is in strict-feedback form.

Backstepping

With the plant dynamics cast into the proper form, this section presents the neuro-adaptive backstepping method in general terms so that it can be applied to either set of dynamics in the previous section. Begin by define the following error states

$$\zeta_1 \triangleq x_1 - x_{1c} \quad (17a)$$

$$\zeta_2 \triangleq x_2 - \bar{x}_2 \quad (17b)$$

$$\zeta_3 \triangleq x_3 - \bar{x}_3 \quad (17c)$$

where \bar{x}_2 and \bar{x}_3 are virtual commands to be constructed that will ensure that the command x_{1c} is tracked.

Step 1: Differentiating ζ_1 yields

$$\dot{\zeta}_1 = \hat{f}_1(x_1) - \delta'_1(x_1) - \dot{x}_{1c} + \zeta_2 + \bar{x}_2, \quad (18)$$

where \bar{x}_2 is now viewed as a virtual control for the ζ_1 dynamics. For consistency of notation, let $\delta'_1 \equiv \delta_1$. Then, to stabilize (18), let

$$\bar{x}_2 = -\hat{f}_1(x_1) - k_1\zeta_1 + \dot{x}_{1c} + \nu_{c1}, \quad (19)$$

where ν_{c1} is an adaptive control term to be specified later. Then substituting (19) into (18) yields

$$\dot{\zeta}_1 = -k_1\zeta_1 + \zeta_2 + \nu_{c1} - \delta_1(x_1). \quad (20)$$

Under ideal conditions, $\nu_{c1} = \delta_1$ and $x_2 \rightarrow \bar{x}_2$, so the error $\zeta_2 \rightarrow 0$, and the ζ_1 dynamics become asymptotically stable.

Step 2: Differentiating ζ_2 yields

$$\dot{\zeta}_2 = \hat{f}_2(x_1, x_2) - \delta_2(x_1, x_2, \zeta_1) + \zeta_3 + \bar{x}_3, \quad (21)$$

where $\delta_2(x_1, x_2, \zeta_1) \triangleq \delta'_2(x_1, x_2) + \dot{\bar{x}}_2$. Let

$$\bar{x}_3 = -\hat{f}_2(x_1, x_2) - \zeta_1 - k_2\zeta_2 + \nu_{c2} \quad (22)$$

so that

$$\dot{\zeta}_2 = -\zeta_1 - k_2\zeta_2 + \zeta_3 + \nu_{c2} - \delta_2(x_1, x_2, \zeta_1). \quad (23)$$

The purpose of introducing ζ_1 in (23) is to compensate for the coupling between the ζ_1 and ζ_2 dynamics. The sign of the ζ_1 in (22) is intentionally chosen as negative to set up a skew-symmetric matrix representing the complete error dynamics. This skew-symmetric structure is a key feature of backstepping controllers, and results in the cancellation of the coupling terms during Lyapunov stability analysis.⁵

Step 3: This last step is very similar to the previous ones except that rather than the virtual control, the actual control signal is constructed. Differentiating ζ_3 yields

$$\dot{\zeta}_3 = \hat{f}_3(x_1, x_2, x_3) - \delta_3(x_1, x_2, x_3, \zeta_1, \zeta_2, u) + \hat{g}_3(x_1, x_2)u, \quad (24)$$

where $\delta_3(x_1, x_2, x_3, \zeta_1, \zeta_2, u) \triangleq \delta'_3(x_1, x_2, x_3, u) + \dot{\bar{x}}_3$. Let

$$u = \hat{g}_3^{-1}(x_1, x_2) \left[-\hat{f}_3(x_2, x_3) - \zeta_2 - k_3\zeta_3 + \nu_{c3} \right] \quad (25)$$

so that

$$\dot{\zeta}_3 = -\zeta_2 - k_3\zeta_3 + \nu_{c3} - \delta_3(x_2, x_3, \zeta_1, \zeta_2, u). \quad (26)$$

Although the term \hat{g} must be chosen such that it is never zero, this condition is easily enforced since it is not a time-varying term, but one chosen by the designer a-priori. In most aircraft control applications, the control power remains non-zero and retains its sign in all but extreme cases.

Equations (20,23,26) can be now expressed in state space form

$$\dot{z} = \bar{A}z + \nu_c - \Delta, \quad (27)$$

where

$$z \triangleq [\zeta_1, \zeta_2, \zeta_3]^T$$

$$\bar{A} = \begin{bmatrix} -k_1 & 1 & 0 \\ -1 & -k_2 & 1 \\ 0 & -1 & -k_3 \end{bmatrix} \quad (28)$$

$$\Delta = [\delta_1, \delta_2, \delta_3]^T \quad (29)$$

$$\nu_c \triangleq [\nu_{c1}, \nu_{c2}, \nu_{c3}]^T. \quad (30)$$

The gains $k_{1,2,3} > 0$ to ensure stability, but they also need to be tuned per application to obtain reasonable performance. If the corrective signal ν_c cancels Δ , then the error dynamics in (27) is stable. The complete control policy is

$$\bar{x}_2 = -\hat{f}_1(x_1) - k_1\zeta_1 + \dot{x}_{1c} + \nu_{c1} \quad (31a)$$

$$\bar{x}_3 = -\hat{f}_2(x_1, x_2) - \zeta_1 - k_2\zeta_2 + \nu_{c2} \quad (31b)$$

$$u = \hat{g}_3^{-1}(x_1, x_2) \left[-\hat{f}_3(x_1, x_2, x_3) - \zeta_2 - k_3\zeta_3 + \nu_{c3} \right]. \quad (31c)$$

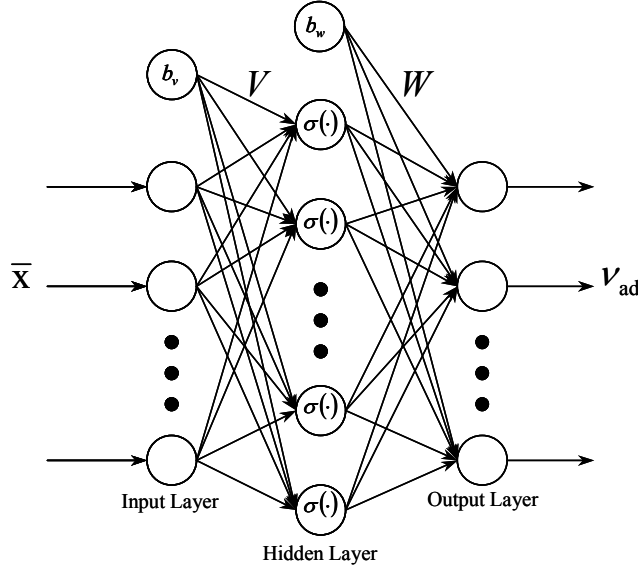


Fig. 1 Multilayered Neural Network

Neural Network

Although, any appropriate adaptive element can be used to generate the corrective signal ν_c , a multilayer neural network (MNN) as shown in Fig. 1 is used herein. These types of network have been shown to be universal approximators,²⁴ and do not require the designer to provide a basis (although the designer must still provided inputs to the NN that parameterize the uncertainty). The input-output map of this sigmoidal MNN is given by

$$y = \hat{\omega}^T \sigma \left(\hat{V}^T \bar{x} \right), \quad (32)$$

where \hat{V} contains the input-to-hidden layer weights, $\hat{\omega}$ the hidden-to-output layer weights, \bar{x} is the network inputs, and $\sigma(\cdot)$ is the activation function

$$\sigma(s) = \frac{1}{1 + e^{-\alpha s}}. \quad (33)$$

Using the approximation property of the MNN, it is assumed that each unknown function, δ_i , in (29) is continuous on a compact set \mathcal{D}_i containing the origin. Thus, it can be approximated by the MNN as

$$\delta_i(x_1, \dots, x_i) = \omega_i^{*T} \sigma(V^{*T} \bar{x}) + \varepsilon_i, \quad (34)$$

where the ω_i^* and V^* are ideal network weights, and the ε_i are corresponding functional reconstruction errors. The actual values of the ideal weights are not needed in implementation. The output weights ω_i^* can be stacked into a matrix

$$W^* \triangleq [\omega_1^{*T}, \omega_2^{*T}, \omega_3^{*T}], \quad (35)$$

so that they can be viewed as the weights of the same, but larger, NN. Then a single MNN with 3 outputs can be used, with $\hat{\omega}_i$ corresponding to the columns of the larger weight matrix, \hat{W} .

$$\hat{W} \triangleq [\hat{\omega}_1^T, \hat{\omega}_2^T, \hat{\omega}_3^T] \quad (36)$$

The adaptive control term ν_c , consists of two components: ν_{ad} is the output of the MNN, and ν_r is a robustifying term. With the following substitutions,

$$\nu_c \triangleq \nu_{ad} + \nu_r \quad (37)$$

$$\nu_{ad} = \hat{W}^T \sigma \left(\hat{V}^T \bar{x} \right) \quad (38)$$

$$\nu_r \triangleq [\nu_{r1}, \nu_{r2}, \nu_{r3}]^T, \quad (39)$$

where ν_{r_i} is to be determined later, (27) can be rewritten as

$$\dot{z} = \bar{A}z + \hat{W}^T \hat{\sigma} - W^{*T} \sigma^* + \nu_r - \varepsilon, \quad (40)$$

where

$$\hat{\sigma} \triangleq \sigma \left(\hat{V}^T \bar{x} \right) \quad (41)$$

$$\sigma^* \triangleq \sigma \left(V^{*T} \bar{x} \right) \quad (42)$$

$$\varepsilon \triangleq [\varepsilon_1, \varepsilon_2, \varepsilon_3]^T. \quad (43)$$

Then from a Taylor-series expansion on the $W^{*T} \sigma^*$ term, (40) can be rewritten as

$$\dot{z} = \bar{A}z - \left\{ \hat{W}^T \left[\hat{\sigma} - \hat{\sigma}' \hat{V}^T \bar{x} \right] + \hat{W}^T \hat{\sigma}' \hat{V}^T \bar{x} - \nu_r + \varpi + \varepsilon \right\} \quad (44)$$

where

$$\varpi \triangleq W^{*T} \mathcal{O}^2 \left(\hat{V}^T \bar{x} \right) + \hat{W}^T \hat{\sigma}' V^{*T} \bar{x}. \quad (45)$$

Theorem 1 *If the following assumptions hold,*

1. All command signals are bounded, $x_{1c} \in C^n$.
2. The matrix \bar{A} in (44) is Hurwitz, i.e. $k_i > 0$, $1 \leq i \leq n$.
3. The control coefficient \hat{g}_3 in (31) has the properties: $\text{sgn}(\hat{g}_3) = \text{sgn}(\hat{g})$ and $|\hat{g}_3| \geq \frac{1}{2} |g|$
4. The neural network input is chosen as $\bar{x} = \begin{bmatrix} z^T & x_{1c} & \dot{x}_{1c} & \nu_{ad} & \|\hat{Z}\| & 1 \end{bmatrix}$.
5. The bound on the ideal weights is known, $\|Z^*\|_F \leq Z$.

and if the following neural network weight update rule is used

$$\dot{\hat{V}} = -\left[\bar{x}z^T\hat{W}^T\hat{\sigma}' + \mu\|z\|\hat{V}\right]\Gamma_V \quad (46a)$$

$$\dot{\hat{W}} = -\left[(\hat{\sigma} - \hat{\sigma}'\hat{V}^T\bar{x})z^T + \mu\|z\|\hat{W}\right]\Gamma_W \quad (46b)$$

where

$$\Gamma_V > 0, \Gamma_W > 0, \mu > 0 \quad (47)$$

$$\nu_r = -\left[a_o + a_1\left(\|\hat{Z}\| + \bar{Z}\right)\right]z \quad (48)$$

with $a_o, a_1 > 0$, then all signals in the system comprised of (44) and (46) remain bounded.

Proof. Please see Refs 3,16,25. *Note: Condition 3 arises from a requirement that the map $\nu_{c_3} \mapsto \delta_3$ be a contraction.*²⁶ ■

Numerical Results

As a demonstration of the control methods developed in the previous sections, they were applied to a nonlinear 3-DOF model of a flying-wing UAV. This type of airframe is particularly challenging as it is statically unstable at low angles of attack and possesses a restricted set of control effectors that provide less yaw authority than the traditional set. Furthermore aerodynamic coefficients from a single flight condition were used in the control design, and the neural network is used to compensate for errors that arise as the aircraft departs from this condition. All neural network weights are initialized at zero.

Figures 2 - 11 summarize the results from a simulation started at 5,000 ft. and 500 ft/sec. The goal of this simulation was to follow steps in γ while maintaining airspeed. A simple linear controller (PI) was used to control airspeed for the purpose of this experiment. The figures show both algorithms developed provide good command tracking and that there is little difference between them. In addition, the MNN adapts to parametric errors in the plant dynamics. It is interesting to note in Figs 10 and 11 that the MNN outputs of two approaches are slightly different despite having identical MNN parameters and learning rates.

Conclusions

A neuro-adaptive backstepping method for the control of flight-path angle has been developed. Numerical results validate the viability of this approach. Although the results presented herein do not differentiate between the two approaches, it is expected that the α option will be superior when considering out-of-plane motion. The reason for this

is that the relationship $\gamma = \theta - \alpha$ no longer holds when the aircraft is banked. Thus in the general case, an explicit θ term cannot be isolated in the $\dot{\gamma}$ equation. However, the $\dot{\gamma}$ equation still contains a lift term which depends strongly on α . Therefore, it is expected that α will be the intermediate variable of choice for application to the complete 6-DOF aircraft model.

Acknowledgments

This research was sponsored by the US Air Force Research Laboratory under contract F33615-01-C-3108.

References

- ¹Ward, D. G., Monaco, J. F., and Bodson, M., "Development and flight testing of a parameter identification algorithm for reconfigurable control," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 6, 1998, pp. 948–956.
- ²Brinker, J. S. and Wise, K. A., "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.
- ³Calise, A. J., Lee, S., and Sharma, M., "Development of a Reconfigurable Flight Control Law for the X-36 Tailless Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 5, 2001, pp. 896–902.
- ⁴Brinker, J. S. and Wise, K. A., "Stability and Flying Qualities Robustness of a Dynamic Inversion Aircraft Control Law," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 6, 1996, pp. 1270–1277.
- ⁵Krstic, M., Kanellakopoulos, I., and Kokotovic, P., *Nonlinear and Adaptive Control Design*, John Wiley & Sons, Inc., New York, 1995.
- ⁶Khalil, H. K., *Nonlinear Systems*, Prentice-Hall, Inc., New Jersey, 2nd ed., 1996.
- ⁷Kokotovic, P. V., "The Joy of Feedback: Nonlinear and Adaptive," *IEEE Control Systems*, June 1992, pp. 7–17.
- ⁸Polycarpou, M. M., "Stable Adaptive Neural Control Scheme for Nonlinear Systems," *IEEE Transactions on Automatic Control*, Vol. 41, No. 3, 1996, pp. 447–451.
- ⁹Polycarpou, M. M. and Mears, M. J., "Stable Adaptive Tracking of Uncertain Systems Using Nonlinearly Parameterized On-Line Approximators," *International Journal of Control*, Vol. 70, No. 3, 1998, pp. 363–384.
- ¹⁰Lee, H. and Tomizuka, M., "Robust Adaptive Control Using a Universal Approximator for SISO Nonlinear Systems," *IEEE Transactions on Fuzzy Systems*, Vol. 8, No. 1, Feb. 2000, pp. 95–106.
- ¹¹Zhang, T., Ge, S. S., and Hang, C. C., "Adaptive Neural Network Control for Strict-Feedback Nonlinear Systems Using Backstepping Design," *Automatica*, Vol. 36, No. 12, Dec. 2000, pp. 1835–1846.
- ¹²Zhang, Y., Peng, P. Y., and Jiang, Z. P., "Stable Neural Controller Design for Unknown Nonlinear Systems Using Backstepping," *IEEE Transactions on Neural Networks*, Vol. 11, No. 6, Nov. 2000, pp. 1347–1360.
- ¹³Kwan, C. M. and Lewis, F. L., "Robust Backstepping Control of Induction Motors Using Neural Networks," *IEEE Transactions on Neural Networks*, Vol. 11, No. 5, Sept. 2000, pp. 1178–1187.
- ¹⁴Kwan, C. M. and Lewis, F. L., "Robust Backstepping Control of Nonlinear Systems Using Neural Networks," *IEEE*

Transactions on Systems, Man, and Cybernetics, Vol. 30, No. 6, Nov. 2000, pp. 753–766.

¹⁵Sharma, M. and Calise, A. J., “Adaptive Backstepping Control for a Class of Nonlinear Systems via Multilayered Neural Networks,” *Proceedings of the American Control Conference*, May 2002.

¹⁶Sharma, M., *A Neuro-Adaptive Autopilot Design for Guided Munitions*, Ph.D. Thesis, Georgia Institute of Technology, Atlanta, 2001.

¹⁷Singh, S. N. and Steinberg, M., “Adaptive Control of Feedback Linearizable Nonlinear Systems with Application to Flight Control,” *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 4, 1996, pp. 871–877.

¹⁸Steinberg, M. L., “Comparison of Intelligent, Adaptive, and Nonlinear Flight Control Laws,” *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 4, 2001, pp. 693–699.

¹⁹Lee, T. and Kim, Y., “Nonlinear Adaptive Flight Control Using Backstepping and Neural Networks Controller,” *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 4, 2001, pp. 675–682.

²⁰Härkegård, O., *Flight Control Design using Backstepping*, Ph.D. Thesis, University of Linköping, Linköping, 2001.

²¹Härkegård, O. and Glad, T., “A backstepping design for flight path angle control,” *Proceedings of the 39th IEEE Conference on Decision and Control*, Dec 2000, pp. 3570–3575.

²²“Multivariable Control Design Guidelines,” Tech. Rep. WL-TR-96-3099, Honeywell and Lockheed Martin, Wright Patterson AFB, OH, May 1996.

²³Stevens, B. L. and Lewis, F. L., *Aircraft Control and Simulation*, John Wiley & Sons, Inc., 1st ed., 1992.

²⁴Hornik, K., “Multilayer Feedforward Networks are Universal Approximators,” *Neural Networks*, Vol. 2, 1989, pp. 359–366.

²⁵Lewis, F. L., Yesildirek, A., and Liu, K., “Multilayer neural-net robot controller with guaranteed tracking performance,” *IEEE Transactions on Neural networks*, Vol. 7, No. 2, March 1996, pp. 388–399.

²⁶Calise, A. J., Hovakimyan, N., and Idan, M., “Adaptive output feedback control of nonlinear systems using neural networks,” *Automatica*, Vol. 37, No. 8, 2001, pp. 1201–1211.

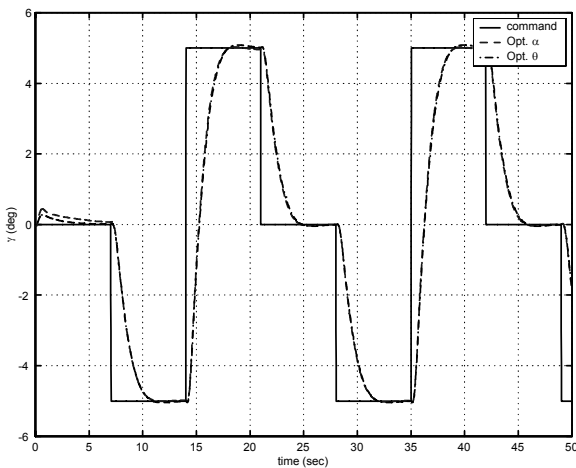


Fig. 2 Comparison of Flight-Path-Angle Tracking

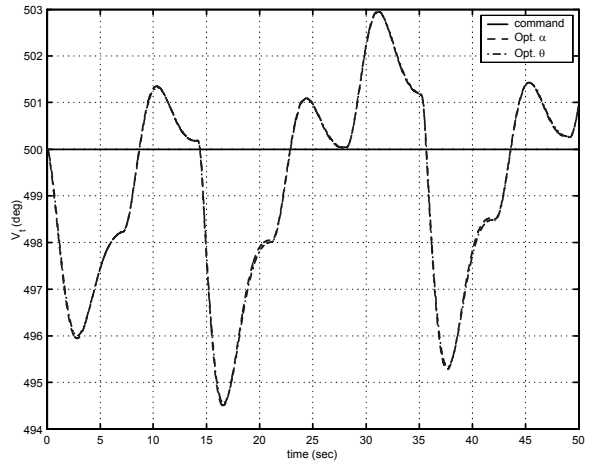


Fig. 3 Comparison of Airspeed Tracking

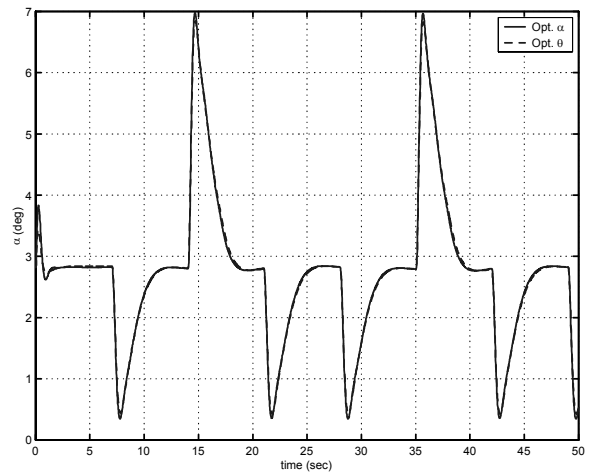


Fig. 4 Comparison of Angle-of-Attack

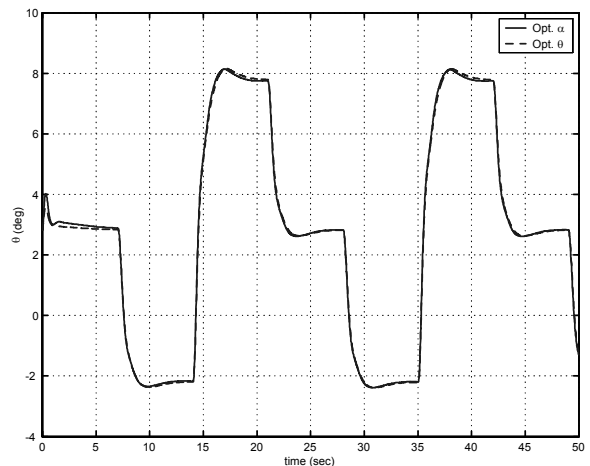


Fig. 5 Comparison of Pitch Attitude

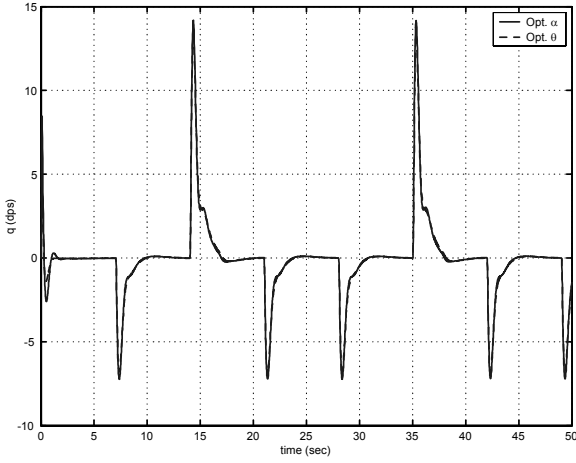


Fig. 6 Comparison of Pitch Rate

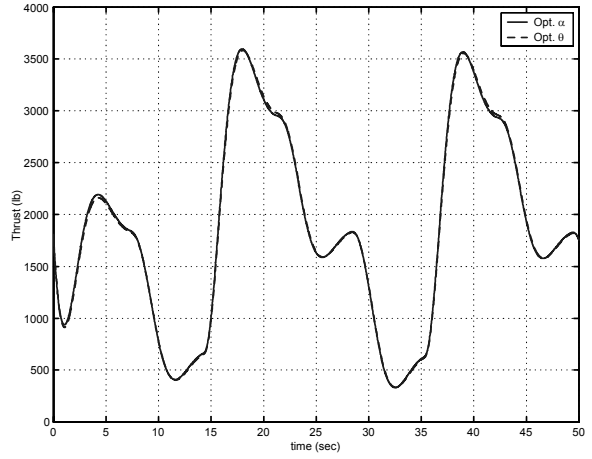


Fig. 9 Comparison of Engine Thrust

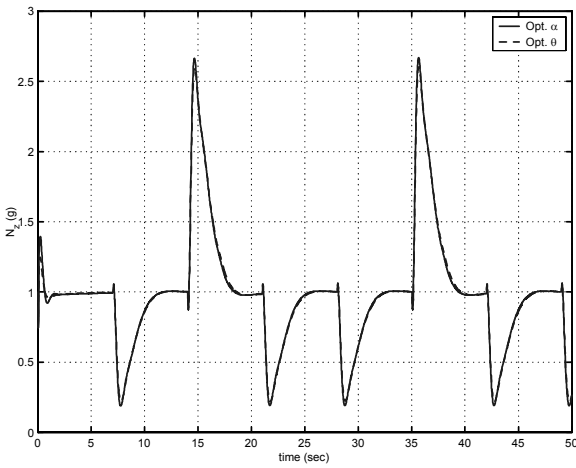


Fig. 7 Comparison of Normal Force

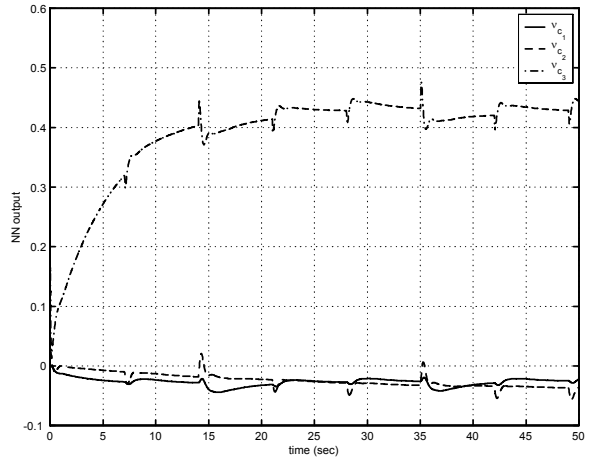


Fig. 10 Neural-Network Output - α as Intermediate Variable

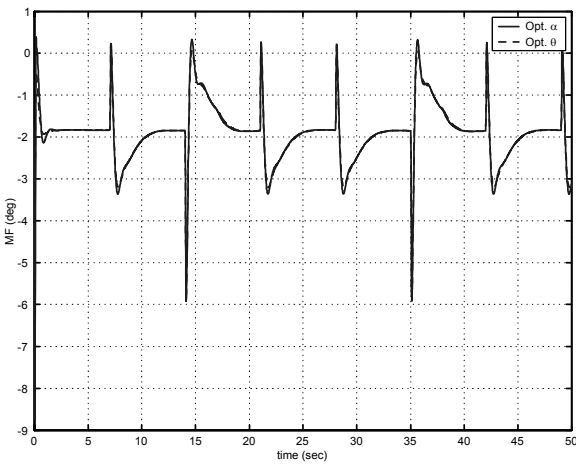


Fig. 8 Comparison of Mid-Board Flap Deflection

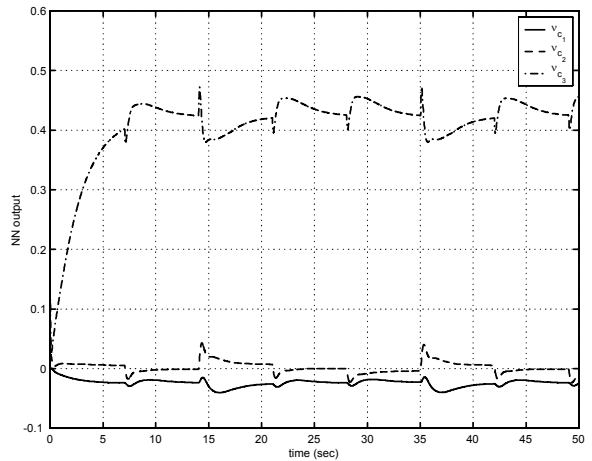


Fig. 11 Neural-Network Output - θ as Intermediate Variable