**Nonlinear Adaptive Flight Control With A Backstepping Design Approach**

**Marc L. Steinberg and Anthony B. Page**

Naval Air Warfare Center Aircraft Division  
22347 Cedar Point Road, Unit #6  
Paxtont River, Maryland 20670-1161

Approved for public release; distribution is unlimited.
Abstract
This paper examines the use of adaptive backstepping for multi-axis control of a high performance aircraft. The control law is demonstrated on a 6 Degree-of-Freedom simulation with nonlinear aerodynamic and engine models, actuator models with saturation, and turbulence. Simulation results are demonstrated for large pitch-roll maneuvers, and for maneuvers with failure of the right stabilator. There are substantial differences between the control law design and simulation models, which are used to demonstrate some robustness aspects of this control law. Actuator saturation is shown to be a considerable problem for this type of controller. However, the flexibility of the backstepping design provides opportunities for improvement. In particular, the Lyapunov function is modified so that the growth of integrated error and the rate of change of parameter growth are both reduced when the surface commands are growing at a rate that will likely saturate the actuators. In addition, the deadzone technique from robust linear adaptive control is applied to improve robustness to turbulence.

Introduction
In the early 1990's, adaptive backstepping was developed by Kanellakopoulos et al\(^1\) as a way of designing stable adaptive control laws for a broad range of nonlinear systems. Adaptive backstepping is an approach that combines Lyapunov stability theory with the substantial advances made in nonlinear differential-geometric control theory\(^2\) in the recent past. The basic concept behind backstepping is to use some states as virtual controls to control other states. However, this initial approach required overparameterization, and yielded high order controllers that were not very practical for most flight control problems. More recent work by Krstic et al\(^3-4\) developed a tuning function approach to eliminate overparameterization. The tuning function approach adds computational complexity to the controller, but yields a controller of the same order as the number of unknown parameters. The primary benefit of this type of controller would seem to be that it allows a wide array of nonlinearities to be incorporated in the controller design, and has proven nominal stability and convergence of error. The types of nonlinearities included in the control law design could be either nonlinearities in the system model or nonlinearities chosen to meet the complex design criteria associated with flight control. In the past, such nonlinearities have not been very successful in flight control\(^5\), but the powerful theoretical tools associated with backstepping may make such designs more feasible. Another potential advantage of backstepping is that it converges very quickly (when it does converge) because it does not have the lags associated with parameter identification for conventional adaptive control approaches. As a result, it may be effective in dealing with damage and failures. However, there are concerns about the robustness of backstepping designs. Even more problematic is that basic backstepping designs tend to generate very large effector commands. This is a serious problem for flight control, due to the importance of actuator saturations in aircraft. Luckily, the flexibility of backstepping design seems to provide opportunities to mitigate this problem. Freeman & Kokotovic\(^6\) and Zhao & Kanellakopoulos\(^7\), for example, provide some interesting approaches to reduce the magnitude of control commands generated by a backstepping control law.

In recent years, backstepping has been applied to a number of research problems including a 1 Degree-of-Freedom wing rock problem\(^8\), electric motor control\(^9\),
Another interesting approach towards nonlinear direct adaptive flight control is that of Kim & Calise, Leitner et al, and McFarland. This approach uses neural networks, but provides theoretical proofs based on similar tools. Singh & Steinberg developed a backstepping adaptive control law with increased robustness through the incorporation of integrated error in the Lyapunov function, and applied it to a simple nonlinear aircraft simulation. The aircraft simulation was a constant velocity model with no disturbances or actuator models. The main contribution of this paper is to demonstrate this adaptive backstepping approach on a much more complex simulation model, and modify the previous design to deal with turbulence and actuator saturations. Dealing with actuator saturations, in particular, provides a good venue to demonstrate the flexibility of backstepping design. The Lyapunov function from ref. 20 is modified to reduce integrated error growth and the rate of change of parameter estimates when the actuator command is approaching saturation. Also, in ref. 20, the only difference between the control law design model and the simulation model was that the lift and drag effects of the control surfaces were not taken into account in the design model. In this paper, there are many more differences from the simulation model, so the results in this paper examine the robustness of this control law to a wide range of errors in the model. Finally, the control law in this paper makes use of the deadzone technique from linear robust adaptive control to improve robustness to turbulence.

Aircraft Simulation Model

The aircraft simulation being used is a high performance aircraft with 2 engines, 2 stabilators, 2 ailerons, 2 rudders, 2 leading edge flaps, and 2 trailing edge flaps. The simulation uses the standard equations of motion and kinematic relations.

\[
\begin{align*}
\dot{u} &= \frac{F_x + F_y}{m} - g \sin \Theta + rv - qw \\
\dot{v} &= \frac{F_x + F_y}{m} + g \cos \Theta \sin \Phi + pw - ru \\
\dot{w} &= \frac{F_z + F_y}{m} + g \cos \Theta \cos \Phi + qu - pv \\
\dot{p}(I_{xx}I_{zz} - I_{xz}^2) &= I_{xx}(l_n + l_r - qr(I_{xx} - I_{rr}) + qpI_{xx}) \\
&= I_{xx}(n_x + n_r + qp(I_{xx} - I_{rr}) - qrI_{xx}) \\
\dot{q}(I_{yy}) &= m_x + m_r + (r^2 - p^2)(I_{xx}) + pr(I_{xx} - I_{xy}) \\
\dot{r}(I_{yy}I_{zz} - I_{xz}^2) &= I_{xx}(n_x + n_r + qp(I_{xx} - I_{rr}) - qrI_{xx}) \\
&= I_{xx}(l_n + l_r - qr(I_{xx} - I_{rr}) + qpI_{xx}) \\
\Phi &= \Theta \cos \Phi + \Psi \cos \Theta \sin \Phi \\
Q &= \Theta \cos \Phi + \Psi \cos \Theta \sin \Phi \\
R &= \Psi \cos \Theta \cos \Phi - \Theta \sin \Phi
\end{align*}
\]

The components of the aerodynamic forces \((F_x, F_y, F_z)\) and moments \((l_n, m_n, n_n)\) are calculated from table look-ups. Gross thrust, \(T\), is calculated from the following equation:

\[
T = [1 + a_1 \alpha + a_2 \alpha^2] F_T(h, M, P_{LT}) [kP_{LT} + c]
\]

where \(a_1, a_2, c, k\) are constants, \(F_T\) is calculated from a table look-up, and \(P_{LT}\) is lagged throttle position. The throttle model is a first order linear system with a variable time constant and variable rate limit based on the value of \(P_{LT}\). The actuator models are 2nd order linear systems with rate and position limits. The turbulence model is the standard Dryden Gust model from MIL-STD-1797A.

Control Law Design Model

For purposes of design, the full simulation model would yield a control law that was far too complex to be practically implemented. Also, it was felt to be important to deliberately have some major differences between the design and simulation model in order to examine the robustness of the control law. As a result, the following model was used...
Some of the key simplifications made in this model are constant velocity (a separate auto-throttle will attempt to maintain this), no lift and drag effects of the control surfaces, and none of the higher frequency dynamics (e.g., actuators). Also, the stabilators and rudders will only be used collectively, and the effects of flaps, which are scheduled with Mach and angle-of-attack, are ignored.

**Backstepping Control Law Design**

This design roughly follows that of ref. 20 and for more detail the reader should consult that reference. However, there are several differences from that controller. First of all, the Lyapunov function is modified so that integrated error growth and the rate of change of unknown parameters are reduced when the actuators seem likely to saturate. Secondly, several scalar constants are turned into constant matrices. Third, all controller parameters are frozen if the actuators saturate, and the actuator error continues to increase. Finally, a deadzone is used to improve the robustness of the system to disturbances. The latter 2 changes have not yet been justified theoretically for the backstepping design, but have been for linear adaptive control designs[21].

Following [20], the aircraft equations of motion can be put in the form

\[
\begin{pmatrix}
\dot{p} \\
\dot{q} \\
\dot{r} \\
\alpha \\
\beta \\
\phi
\end{pmatrix} =
\begin{pmatrix}
l_p \beta + l_q \beta + l_r (l_\beta p + l_r) \Delta \alpha + l_r p - i \varphi_r \\
\bar{m}_q \Delta \alpha + \bar{m}_q \varphi + l_p r - m_p \beta + m_q / \bar{V} \cos \theta \cos \phi - \cos \theta_0 \\
\varphi_n \beta + \varphi_n \varphi + \varphi_p \varphi + \varphi_p \varphi - i \varphi_n q \\
y_\beta \beta + \varphi \sin \alpha \cos \alpha + \Delta \alpha - r \cos \alpha_0 + (\bar{g}_0 / \bar{V}) \cos \theta \sin \phi \\
\varphi + \varphi \tan \theta \sin \phi + \varphi \tan \theta \cos \phi \\
p + q \tan \theta \sin \phi + r \tan \theta \cos \phi
\end{pmatrix}
\]

We will next define the error, \( e \), and a function \( s \) that combines error and integrated error

\[
e = y - y_c
\]

\[
s = e + K_0 x_i
\]

where \( K_0 \) is a positive definite symmetric matrix and \( y_c \) is the output of a command generator that is a linear, stable 3rd order system. The function \( f_c \) is a scalar bounded, twice differentiable, positive function with bounded derivatives that normally equals 1, but will be decreased to reduce the growth of integrated error when the actuators are approaching saturation. Choice of \( f_c \) will be discussed in greater detail below. Taking the derivative of \( s \) yields

\[
\dot{s} = \Phi(x_i)w_1 + B(x_i)(\omega_d + \hat{\omega}) + \nu
\]

\[
\omega = \omega_d + \hat{\omega}
\]

\[
\nu = \Phi_0 - \dot{y}_c + K_0 f_c e
\]

In this equation, \( \omega_d \) represents the desired value of the virtual controls. However, since these are states and not directly controllable effectors, there will be an error \( \hat{\omega} \).

We will next choose \( \omega_d \) through the use of the following Lyapunov function

\[
U_I = \left( s^T f_c s + \dot{w}_1^T L_1 \dot{w}_1 \right) / 2
\]

\[
\dot{w}_1 = w_i - \dot{w}_i
\]
where $L_i$ is a positive definite diagonal matrix and $\hat{w}_i$ is the estimate of $w_i$.

Taking the derivative of $U_1$ yields

$$U_1 = s^T f_c \left[ \Phi(x_i)w_i + B(x_i)(\hat{\omega} + \omega_d) + \nu(x_i, \dot{y}_c, f_c) \right]$$

$$+ \hat{w}_i^T L_4 \hat{w}_i + f_c s^T s$$

$$\omega_d$$ can then be chosen as

$$\omega_d = B^{-1}(x_i) [-K_{11}s - \Phi^T(x_i)\hat{w}_1 - \nu(x_i, \dot{y}_c, f_c)]$$

where $K_{11}$ is a positive definite diagonal matrix.

As a result of that choice of $\omega_d$

$$U_1 = -(K_{11}f_c - f_c)s^T s + \hat{w}_i^T \left[ f_c \Phi^T s + L_4 \hat{w}_1 \right]$$

$$+ f_c \hat{\omega}^T B^T(x_i) s$$

If we put $\hat{\omega}_d$ in the form

$$\hat{\omega}_d = \psi_{0a} - \psi_{1d} w_1 + \psi_{2d} w_2$$

then,

$$\dot{\hat{\omega}} = \psi_{0a} + \psi_{1a} \hat{w}_1 + \psi_{2a} \hat{w}_2 + D(x, w_u)u$$

$$\psi_{0a} = \psi_0 - \psi_{0d} + \psi_{1d} \hat{w}_1 + \psi_{2d} \hat{w}_2$$

$$\psi_{1a} = \psi_{1d}$$

$$\psi_{2a} = \psi_{1d} - \psi_{2d}$$

We will next chose a 2nd Lyapunov function such that

$$U_2 = U_1 + (\hat{\omega}^T f_c \hat{\omega} + \hat{w}_i^T L_2 \hat{w}_i + \hat{w}_u^T L_3 \hat{w}_u + x^T_0 K_{33} x_0) / 2$$

where $L_1$, $L_2$, and $K_{33}$ are all positive definite diagonal matrices and $\hat{x}_u$ is given by

$$\dot{\hat{x}}_u = f_c \hat{\omega}$$

Taking the derivative of $U_2$ gives

$$\dot{U}_2 = -(K_{11}f_c - f_c)s^T s + \hat{w}_i^T \left[ f_c \Phi^T s + L_4 \hat{w}_1 + f_c \psi_{1d} \hat{\omega} \right]$$

$$+ \hat{w}_i^T L_4 \hat{w}_i + f_c \hat{\omega}^T B^T(x_i) s$$

$$+ \hat{w}_u^T L_3 \hat{w}_u + f_c \hat{\omega}^T \hat{\omega}$$

We will next chose the following control law

$$u = \hat{D}^{-1}(-B^T s -(\psi_0 - \psi_{0d} + \psi_{1d} \hat{w}_1)$$

$$+ (\psi_1 - \psi_{2d}) \hat{w}_2) - K_{22} \hat{\omega} - K_{33} x_u$$

$$\dot{\hat{w}}_1 = f_c L_4^{-1}(\Phi^T s + (\psi_{1d} \hat{\omega}))$$

$$\dot{\hat{w}}_2 = f_c L_5^{-1} \psi_{1d} \hat{\omega}$$

$$\hat{\omega}_u = f_c L_5^{-1} \psi_{1d} \hat{\omega}$$

where $\psi_{1d}$ is chosen so

$$[D(x, w_u) - D(x, \hat{w}_u)] = \Psi_{1d}(x, u) \hat{w}_u$$

As a result, since $K_{11}$ and $K_{22}$ are postive definite diagonal matrices and $f_c$ is always positive, then, if $f_c$ is chosen correctly

$$\dot{U}_2 = -(K_{11}f_c - f_c)s^T s - (K_{22}f_c - f_c) \hat{\omega}^T \hat{\omega} \leq 0$$

Following the approach of ref. 20, it can be shown that the for the nominal system, $s$, $\hat{\omega}$, and $e$ all tend to zero as $t \rightarrow \infty$.

$f_c$ is made up of 2 components. The first is a fuzzy logic component and the second is a 3rd order linear stable system, chosen such that $f_c$ meets the above requirements for the range of possible inputs from the fuzzy logic system $f_{in}$. Fuzzy logic was used because it is an easy to design way of building a function to transition from 1 to the minimum value of $f_{in}$. The fuzzy logic rules are as follows:

1) If actuator position is normal and actuator rate is normal, then $f_{in}$ is normal (i.e., 1)

2) If (actuator position is large or actuator rate is fast) and $s$ is increasing rapidly, then $f_{in}$ is medium

3) If (actuator position is very large or actuator rate is very fast) and $s$ is increasing, then $f_{in}$ is medium

4) If (actuator position is very large or actuator rate is very fast) and $s$ is increasing rapidly, then $f_{in}$ is small

5) If (actuator position is near saturation or actuator rate is near saturation) and (s is increasing or increasing rapidly), then $f_{in}$ is small

In addition, if the actuator saturates and filtered actuator error is increasing above a threshold value, then $f_c$ is immediately set to zero. This violates the above stability proof, but is necessary to prevent departures in some situations.
Simulation Results

All simulation results were generated by Matlab/Simulink v. 4.2.1 using the RK45 integration routine and the full nonlinear simulation model with actuator and engine models.

Fig. 1 demonstrates the performance of the control law making a combined 60 degree roll and 8 degree angle-of-attack change while attempting to maintain zero sideslip at Mach .5 and 45,000 ft. altitude with no turbulence. The solid lines are the commanded values and the dotted lines are the actual values. This is a particularly difficult flight condition for the controller because the low dynamic pressure can lead to large control magnitudes that could saturate the actuators. As you can see, the roll response is very good. The alpha response is not quite as good, but this is partly due to the fact that the auto-throttle was not perfect and the velocity was fluctuating during the maneuver. There is some slight saturation of the rudder actuator. However, this is much less than would occur with the basic control law of ref. 20 without the actuator saturation mitigation approach of this paper. Fig. 2 shows the desired and actual values of the virtual control inputs for this case. Fig. 3 shows what happens for this same scenario, with $f_c=1$ at all times. In this case, the actuators rapidly saturate, and the aircraft departs. Of course, with this initial control law, smaller gains could have been used. However, the use of smaller gains led to very poor tracking performance throughout the envelope.

Figure 4 shows the same maneuver at the same flight condition with a failure of the right stabilator at 1.5 seconds. There is only fairly slight degradation of the roll and sideslip response and modest degradation of the alpha response. The reason for the good response can be seen in Fig. 5, which shows the change in estimated stab effectiveness. Due to the lack of filters, the parameter converges to a new reduced value very rapidly after the failure at 1.5 sec.

Tables 1 through 3 provide some error statistics for a 180 degree roll with an 8 degree angle-of-attack change followed by a return to the initial roll and angle-of-attack angles while attempting to maintain zero sideslip in moderate turbulence, as defined in MIL-STD-1797 [22]. The controller performs acceptably at all flight conditions. As mentioned earlier, the low dynamic pressure condition had the most difficulties with departures due to actuator saturation, but the high dynamic pressure condition has the worst tracking errors.

Despite the improvements made in mitigating the impact of actuator saturation through the $f_c$ function, the control law still cannot be used for very fast maneuvers. Fig. 6 shows an attempt to command a much faster roll. In this case, the actuators saturate despite the use of $f_c$ and cause the aircraft to depart.

Conclusions

This paper demonstrates an adaptive backstepping flight control law on a complex high performance aircraft simulation. The control law is demonstrated to have very fast convergence properties and to provide good performance in many situations, including the loss of a stabilator. It also demonstrated good robustness properties to a wide range of modeling errors. However, more work needs to be done to reduce the magnitudes of actuator commands, while still maintaining good performance. Achieving good performance generally required limiting the commands to the controller to prevent departure. Some of the potential ways of dealing with this could include better control allocation, improvements in the use of the $f_c$ function, and changes in the Lyapunov function to reduce the effective gain of the system or to modify the gain based on dynamic pressure. Luckily, backstepping design provides considerable flexibility to alter the Lyapunov function in such ways. However, it is usually not clear what effect many changes will have on the system and better theoretical tools and further simulation will be required.

References


Table 1 - Mach .7, 30K ft altitude

<table>
<thead>
<tr>
<th></th>
<th>Average Absolute Error</th>
<th>Maximum Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phi (deg.)</td>
<td>.74</td>
<td>1.9</td>
</tr>
<tr>
<td>Alpha (deg.)</td>
<td>.67</td>
<td>1.2</td>
</tr>
<tr>
<td>Beta (deg.)</td>
<td>.43</td>
<td>1.4</td>
</tr>
</tbody>
</table>

Table 2 - Mach .9, 5K ft altitude

<table>
<thead>
<tr>
<th></th>
<th>Average Absolute Error</th>
<th>Maximum Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phi (deg.)</td>
<td>1.8</td>
<td>4.7</td>
</tr>
<tr>
<td>Alpha (deg.)</td>
<td>0.93</td>
<td>2.1</td>
</tr>
<tr>
<td>Beta (deg.)</td>
<td>0.87</td>
<td>2.8</td>
</tr>
</tbody>
</table>
Table 3 - Mach .5, 45K altitude

<table>
<thead>
<tr>
<th></th>
<th>Average Absolute Error</th>
<th>Maximum Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phi (deg.)</td>
<td>0.71</td>
<td>2.6</td>
</tr>
<tr>
<td>Alpha (deg.)</td>
<td>0.64</td>
<td>2.2</td>
</tr>
<tr>
<td>Beta (deg.)</td>
<td>0.49</td>
<td>2.4</td>
</tr>
</tbody>
</table>

Figure 1 - No Failure Case with Actuator Saturation Mitigation Activated
Figure 2 - Desired and Actual Values of Virtual Controls for No Failure C
Figure 3 - No Failure Case without Actuator Saturation Mitigation
Figure 4 - Stabilator Failure Case

Figure 5 - Stabilator Effectiveness Estimate at failure
Figure 6 - Very Fast Roll Command