JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 30, No. 2, March–April 2007

# Nonlinear Flight Control Design Using Constrained Adaptive Backstepping

L. Sonneveldt,* Q. P. Chu,† and J. A. Mulder‡
*Delft University of Technology, 2600 GB Delft, The Netherlands*

DOI: 10.2514/1.25834

**The design of an adaptive backstepping flight control law for the F-16/MATV (multi-axis thrust vectoring) aircraft is discussed. The control law tracks reference trajectories with the angle of attack $\alpha$, the stability-axes roll rate $p_s$, and the total velocity $V_T$. Furthermore, the sideslip angle $\beta$ has to be kept at zero. B-spline neural networks are used inside the parameter update laws of the backstepping control law to approximate the uncertain aerodynamic forces and moments. Command filters are used to implement the constraints on the control surfaces and the virtual control states. The stability of the parameter estimation process during periods of saturation is guaranteed by using a modified tracking error definition, in which the effect of the saturation has been filtered out. The controller and its performance are evaluated on a nonlinear, six-degrees-of-freedom dynamic model of an F-16/MATV aircraft in a number of simulation scenarios.**

## Nomenclature

| | | |
|---|---|---|
| $C_*$ | = | aerodynamic coefficient of * |
| $c_i$ | = | inertia parameters |
| $F_T$ | = | thrust force, N |
| $g_1, g_2, g_3$ | = | gravity components, m/s$^2$ |
| $I_x$ | = | roll moment of inertia, kg $\cdot$ m$^2$ |
| $I_y$ | = | pitch moment of inertia, kg $\cdot$ m$^2$ |
| $I_z$ | = | yaw moment of inertia, kg $\cdot$ m$^2$ |
| $I_{xz}$ | = | product moment of inertia, kg $\cdot$ m$^2$ |
| $\bar{L}, \bar{M}, \bar{N}$ | = | rolling, pitching, and yawing moments, N $\cdot$ m |
| $m$ | = | mass, kg |
| $p, q, r$ | = | body-axis roll, pitch and yaw rates, rad/s |
| $p_s, q_s, r_s$ | = | stability-axis roll, pitch and yaw rates, rad/s |
| $p_{\text{static}}$ | = | static pressure, Pa |
| $\bar{q}$ | = | dynamic pressure, Pa |
| $q_0, q_1, q_2, q_3$ | = | quaternion components |
| $S$ | = | reference wing surface area, m$^2$ |
| $T_{s/b}$ | = | rotation matrix body-fixed to stability-axes reference frame |
| $T_{w/b}$ | = | rotation matrix body-fixed to wind-axes reference frame |
| $V_T$ | = | total velocity, m/s |
| $\bar{X}, \bar{Y}, \bar{Z}$ | = | axial, lateral, and normal force components, N |
| $x_T, z_T$ | = | thrust point $x$-axis and $z$-axis locations, m |
| $z_i$ | = | tracking error |
| $\bar{z}_i$ | = | modified tracking error |
| $\alpha$ | = | angle of attack, rad |
| $\alpha_i$ | = | virtual control law |
| $\beta$ | = | angle of sideslip, rad |
| $\Gamma_*$ | = | update gain * |
| $\delta_e, \delta_a, \delta_r$ | = | elevator, aileron, and rudder deflections, rad |
| $\delta_{\text{LEF}}$ | = | leading edge flap deflection, rad |
| $\delta_{\tau_z}, \delta_{\tau_y}$ | = | horizontal and vertical engine nozzle deflections, rad |
| $\chi$ | = | constraint effect filter |

*Subscripts*

| | | |
|---|---|---|
| $r$ | = | reference |
| 0 | = | nominal |

*Superscripts*

| | | |
|---|---|---|
| ref | = | reference |
| $\hat{}$ | = | estimate |
| $\tilde{}$ | = | error |

Presented as Paper 6413 at the AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, Colorado, 21–24 August 2006; received 13 June 2006; revision received 9 October 2006; accepted for publication 3 November 2006. 

*Ph.D. Student, Control and Simulation Division, Faculty of Aerospace Engineering, P.O. Box 5058; l.sonneveldt@tudelft.nl.

†Associate Professor, Control and Simulation Division, Faculty of Aerospace Engineering, P.O. Box 5058; q.p.chu@lr.tudelft.nl.

‡Professor, Chairman of the Control and Simulation Division, Faculty of Aerospace Engineering, P.O. Box 5058; j.a.mulder@lr.tudelft.nl.

## I. Introduction

OVER the last few decades and pushed by developments in aircraft technology, the performance requirements on modern fighter aircraft became more and more challenging throughout an ever increasing flight envelope. Extreme maneuverability is achieved by designing the aircraft with multiple redundant control actuators and allowing static instabilities in certain modes. A good example is the F-22 Raptor, which makes use of thrust-vectored control to increase maneuverability. In recent years, the requirement for fighter aircraft to remain controllable while damaged has led to renewed interest in adaptive and fault tolerant control methods. Taking into account all these requirements for modern fighter aircraft poses a huge challenge for flight control system designers.

Traditionally, aircraft control systems were designed using linearized aircraft models at multiple trimmed flight conditions throughout the flight envelope. For each of these operating points a corresponding linear controller is derived using the well-established linear-based control design methods [1]. One of the many gain scheduling methods [2–4] can next be applied to derive a single flight control law for the entire flight envelope. However, a problem of this approach is that good performance and robustness properties cannot be guaranteed for a highly nonlinear fighter aircraft. Nonlinear control methods have been developed to overcome the shortcomings of linear design approaches. The theoretically established feedback linearization approach is the best known and most widely used of these methods [5–8].

Feedback linearization [9,10] is a nonlinear design method that can explicitly handle systems with known nonlinearities. By using nonlinear feedback and exact state transformations rather than linear approximations the nonlinear system is transformed into a constant linear system. This linear system can in principle be controlled by just a single linear controller. However, to perform perfect feedback

linearization all nonlinearities have to be precisely known. This is generally not the case with modern fighter aircraft, because it is very difficult to precisely know and model their complex nonlinear aerodynamic characteristics. Empirical data are usually obtained from wind tunnel experiments and flight tests, augmented by computational fluid dynamics (CFD) results, and thus are not 100% accurate. The problem of model deficiencies can be dealt with by closing the control loop with a linear, robust controller, for example, an $H_\infty$ controller [11] or a model predictive controller [12]. However, even then desired performance cannot be expected in case of gross errors, due to large, sudden changes in the aircraft dynamics resulting from, for example, battle damage or actuator failure.

In these cases nonlinear adaptive control methods are called for. The problem of combining feedback linearization with an adaptive outer loop [13] is that it does not give any closed-loop stability or tracking error convergence guarantees. One nonlinear adaptive control method that does give these guarantees is the adaptive backstepping design approach, which has received much attention in recent years [14–22]. Adaptive backstepping is a recursive, Lyapunov-based, nonlinear design method, that makes use of online parameter estimation laws to deal with parametric uncertainties. Adaptive backstepping can be applied to nonlinear systems that can be transformed into a lower-triangular form. The basic idea behind backstepping is to use some states as virtual controls to control other states.

However, a key problem with the adaptive backstepping method is that mechanical limitations of the aircraft control surfaces cannot be explicitly handled. A command filtered approach for nonlinear systems is proposed in [23–25], which fits within the recursive adaptive backstepping design procedure. This approach makes use of command filters to implement any mechanical or operating constraints on the control variables and the virtual control states. The effect of these constraints on the input and states is filtered and removed from the parameter update laws to ensure a stable parameter estimation process even when these limitations are in effect. An additional advantage of this approach is that the two other main drawbacks of the adaptive backstepping methods are also eliminated, that is, the tedious analytic computation of virtual control signal derivatives and the restriction to nonlinear systems of lower-triangular form.

To be able to estimate the aircraft's nonlinear aerodynamic force and moment coefficients, neural networks can be used within the parameter update laws of the adaptive backstepping controller [26]. The neural networks are used to parameterize the nonlinear functions, so that the update laws can adapt the unknown parameters, that is, the network weights. The advantage of using update laws to adapt the network weights is that the adaptation process has the strong stability and convergence properties of the Lyapunov-based adaptive backstepping approach. Numerical and computational advantages are obtained by selecting radial basis function (RBF) neural networks with B-spline basis functions [27,28].

In this paper the command filtered adaptive backstepping approach [23–25] is used to design a control law for a nonlinear, six-degrees-of-freedom F-16 model with multi-axis thrust vectoring (MATV). The objectives of the control law are to track reference trajectories with $V_T$, $\alpha$, and $p_s$. Furthermore, regulation $\beta$ is provided. Flight control laws with similar control objectives are designed in [29,30]; in the latter work the roll angle is a controlled variable instead of the roll rate. The aerodynamic forces and moment coefficients of the nonlinear aircraft model are estimated online with parameter update laws that make use of B-spline neural networks to parameterize these nonlinear functions. The control law is evaluated on the high-fidelity F-16/MATV model in simulations under both nominal conditions and several damaged conditions, that is, large and sudden changes in aerodynamic coefficients and actuator failures.

The outline of this paper is as follows: First, the flight dynamics of the F-16/MATV model are briefly discussed in Sec. II. Section III is devoted to the theory behind the constrained adaptive backstepping approach. In Sec. IV the flight control law for the F-16/MATV model is derived. Section V discusses the update laws with the B-spline neural networks. The control law is tuned in Sec. VI and simulation examples are shown in Sec. VII. Finally, the conclusions can be found in Sec. VIII.

## II. F-16/MATV DYNAMICS

The control law design is based on the nonlinear model of an F-16 aircraft with MATV [31]. The nonlinear F16/MATV model was created using aircraft specific data, such as the aircraft geometry and the aerodynamic model, obtained from [32]. The aerodynamic data are valid up to Mach 0.6, angle of attack range of $-20 \leq \alpha \leq 90$ deg, and sideslip angle range $-30 \leq \beta \leq 30$ deg. The wind tunnel tests were conducted on sufficiently close points to capture the nonlinear behavior of the aerodynamic force and moment coefficients. Aerodynamic data for the intermediate points were linearly interpolated. The simple thrust vector control model is based on the one used in [33]. The relevant equations of motions used for the control design can be written as [32]

$$\begin{aligned}
\dot{V}_T &= \frac{1}{m}[-\bar{X}\cos\alpha\cos\beta + \bar{Y}\sin\beta + \bar{Z}\sin\alpha\cos\beta \\
&\quad + F_T\cos(\alpha+\delta_{\tau_y})\cos(\beta+\delta_{\tau_z}) + mg_1] \\
\dot{\alpha} &= q_s - p_s\tan\beta + \frac{1}{mV_T\cos\beta}[-\bar{X}\sin\alpha + \bar{Z}\cos\alpha \\
&\quad - F_T\sin(\alpha+\delta_{\tau_y}) + mg_3] \\
\dot{\beta} &= -r_s + \frac{1}{mV_T}[-\bar{X}\cos\alpha\sin\beta + \bar{Y}\cos\beta - \bar{Z}\sin\alpha\sin\beta \\
&\quad - F_T\cos(\alpha+\delta_{\tau_y})\sin(\beta+\delta_{\tau_z}) + mg_2]
\end{aligned} \tag{1}$$

$$\begin{bmatrix}\dot{q}_0\\ \dot{q}_1\\ \dot{q}_2\\ \dot{q}_3\end{bmatrix} = \frac{1}{2}\begin{bmatrix}0 & -p & -q & -r\\ p & 0 & r & -q\\ q & -r & 0 & p\\ r & q & -p & 0\end{bmatrix}\begin{bmatrix}q_0\\ q_1\\ q_2\\ q_3\end{bmatrix} \tag{2}$$

$$\begin{bmatrix}\dot{p}_s\\ \dot{q}_s\\ \dot{r}_s\end{bmatrix} = T_{s/b}\begin{bmatrix}(c_1 r + c_2 p)q + c_3(\bar{L}+L_T) + c_4(\bar{N}+N_T)\\ c_5 pr - c_6(p^2 - r^2) + c_7(\bar{M}+M_T)\\ (c_8 p - c_2 r)q + c_4(\bar{L}+L_T) + c_9(\bar{N}+N_T)\end{bmatrix} + \begin{bmatrix}r_s\\ 0\\ -p_s\end{bmatrix}\dot{\alpha} \tag{3}$$

where

$$\begin{aligned}
L_T &= -F_T z_T \sin\delta_{\tau_z} \\
M_T &= F_T(x_T\sin\delta_{\tau_y}\cos\delta_{\tau_z} + z_T\cos\delta_{\tau_y}\cos\delta_{\tau_z}) \\
N_T &= F_T x_T \sin\delta_{\tau_z}
\end{aligned} \tag{4}$$

$$\begin{bmatrix}g_1\\ g_2\\ g_3\end{bmatrix} = T_{w/b}\begin{bmatrix}2(q_1q_3 - q_0q_2)g\\ 2(q_2q_3 + q_0q_1)g\\ (q_0^2 - q_1^2 - q_2^2 + q_3^2)g\end{bmatrix} \tag{5}$$

$$\Gamma c_1 = (I_y - I_z)I_z - I_{xz}^2 \qquad \Gamma c_4 = I_{xz} \qquad c_7 = \frac{1}{I_y}$$

$$\Gamma c_2 = (I_x - I_y + I_z)I_{xz} \qquad c_5 = \frac{I_z - I_x}{I_y}$$

$$\Gamma c_8 = I_x(I_x - I_y) + I_{xz}^2 \qquad \Gamma c_3 = I_z \qquad c_6 = \frac{I_{xz}}{I_y} \qquad \Gamma c_9 = I_x$$

with $\Gamma = I_x I_z - I_{xz}^2$.

**Table 1 The control input units and maximum values**

| Control | Units | Min. | Max. | Rate limit |
|---|---|---|---|---|
| Thrust | N | 1000 | 100,000 | ±40,000 N/s |
| Elevator | deg | −25 | 25 | ±60 deg/s |
| Ailerons | deg | −21.5 | 21.5 | ±80 deg/s |
| Rudder | deg | −30 | 30 | ±120 deg/s |
| Engine nozzle $xy$ frame | deg | −17 | 17 | ±45 deg/s |
| Engine nozzle $xz$ frame | deg | −17 | 17 | ±45 deg/s |
| Leading edge flap | deg | 0 | 25 | ±25 deg/s |

Quaternions rather than the usual Euler angles are used to express the attitude of the aircraft to avoid the classical singularities of Euler angles in extreme maneuvers.

The F-16/MATV model allows for control over thrust, elevator, ailerons, rudder, and the thrust vector in horizontal and vertical directions. It is assumed that thrust measurements are available. All control inputs are defined positive in the conventional way, that is, positive thrust causes an increase in acceleration along the body $x$ axis, a positive elevator deflection results in a decrease in pitch rate, a positive aileron deflection gives a decrease in roll rate, and a positive rudder deflection decreases the yaw rate. The engine nozzle deflections are defined positive downward and to the right. The F-16/MATV is equipped with automatic leading edge flaps, which are deflected according to a transfer function dependent on $\alpha$ and a bias depending on the ratio of dynamic pressure and static pressure (Mach number):

$$\delta_{\text{LEF}} = 1.38\frac{2s+7.25}{s+7.25}\alpha - 9.05\frac{\bar{q}}{p_{\text{static}}} + 1.45$$

The control surfaces of the F-16/MATV are driven by servocontrolled actuators to produce the deflections commanded by the flight control system $u$, which are the true control variables. The actuators of the control surfaces are modeled as first-order low-pass filters with fixed gain and saturation limits in range and deflection rate; see Table 1. The gains of the actuators are $1/0.136$ for the leading edge flaps and $1/0.0495$ for the other control surfaces. The thrust model used in the F-16/MATV is also modeled as a low-pass filter with gain 1. A more complex thrust model is available in [34], but was not implemented.

The maximum values and units for all control variables are given in Table 1. Note that the maximum total deflection of the engine nozzle is 17 deg, which means that $\delta_{\tau_y}$ and $\delta_{\tau_z}$ cannot reach the maximum deflection of 17 deg simultaneously. The rate and magnitude limits of the engine nozzle deflections were taken from [35].

## III. Constrained Adaptive Backstepping

### A. Basic Backstepping

Backstepping [15] is a systematic, Lyapunov-based method for nonlinear control design, which can be applied to nonlinear systems that can be transformed into lower-triangular form, such as the system of Eq. (6):

$$\dot{x}_1 = f(x_1) + g(x_1)x_2 \qquad \dot{x}_2 = u \tag{6}$$

The name "backstepping" refers to the recursive nature of the control law design procedure. Using the backstepping procedure, a control law is recursively constructed, along with a control Lyapunov function (CLF) to guarantee global stability. For the system Eq. (6), the aim of the design procedure is to bring the state vector $x_1$ to the origin. The first step is to consider $x_2$ as the *virtual control* of the scalar $x_1$ subsystem and to find a desired virtual control law $\alpha_1(x_1)$ that stabilizes this subsystem by using the control Lyapunov function $V_1(x_1)$:

$$V_1(x_1) = \tfrac{1}{2}x_1^2 \tag{7}$$

The time derivative of this CLF is negative definite

$$\dot{V}_1 = \frac{\partial V_1}{\partial x_1}(x_1)[f(x_1) + g(x_1)\alpha_1(x_1)] < 0, \qquad x_1 \neq 0 \tag{8}$$

if only the virtual control law

$$x_2 = \alpha_1(x_1) \tag{9}$$

could be satisfied. The key property of backstepping is that we can now "step back" through the system. If the error between $x_2$ and its desired value is defined as

$$z = x_2 - \alpha_1(x_1) \tag{10}$$

the system Eq. (6) can be rewritten in terms of this error state

$$\begin{aligned} \dot{x}_1 &= f(x_1) + g(x_1)[\alpha_1(x_1) + z] \\ \dot{z} &= u - \frac{\partial \alpha_1}{\partial x_1}(x_1)(f(x_1) + g(x_1)[\alpha_1(x_1) + z]) \end{aligned} \tag{11}$$

The control Lyapunov function Eq. (7) can now be expanded with a term penalizing the error state $z$

$$V_2(x_1, z) = V_1(x_1) + \tfrac{1}{2}z^2 \tag{12}$$

In simplified notation the time derivative of $V_2$ is equal to

$$\begin{aligned} \dot{V}_2 &= \frac{\partial V_1}{\partial x_1}[f + g(\alpha_1 + z)] + z\left(u - \frac{\partial \alpha_1}{\partial x_1}[f + g(\alpha_1 + z)]\right) \\ &= \frac{\partial V_1}{\partial x_1}(f + g\alpha_1) + z\left(\frac{\partial V_1}{\partial x_1}g + u - \frac{\partial \alpha_1}{\partial x_1}[f + g(\alpha_1 + z)]\right) \end{aligned} \tag{13}$$

which can be rendered negative definite with the control law

$$u = -cz + \frac{\partial \alpha}{\partial x_1}[f + g(\alpha_1 + z)] - \frac{\partial V_1}{\partial x_1}g, \qquad c > 0 \tag{14}$$

This design procedure can also be used for a system with a chain of integrators. The only difference is that there will be more virtual states to "backstep" through. Starting with the state "farthest" from the actual control, each step of the backstepping technique can be broken up into three parts:

1) Introduce a virtual control $\alpha$ and an error state $z$, and rewrite the current state equation in terms of these;

2) choose a CLF for the system, treating it as a final stage;

3) choose an equation for the virtual control that makes the CLF stabilizable.

The CLF is augmented at subsequent steps to reflect the presence of new virtual states, but the same three stages are followed at each step. Hence, backstepping is a recursive design procedure.

### B. Adaptive Backstepping

For systems with parametric uncertainties there exists a method called adaptive backstepping [36], which achieves boundedness of the closed-loop states and convergence of the tracking error to zero.

Consider the parametric strict-feedback system

$$\begin{aligned} \dot{x}_1 &= x_2 + \varphi_1(x_1)^T\theta \\ \dot{x}_2 &= x_3 + \varphi_2(x_1, x_2)^T\theta \\ &\vdots \\ \dot{x}_{n-1} &= x_n + \varphi_{n-1}(x_1, \ldots, x_{n-1})^T\theta \\ \dot{x}_n &= \beta(x)u + \varphi_n(x)^T\theta \end{aligned} \tag{15}$$

where $\beta(x) \neq 0$ for all $x \in \mathbb{R}^n$, $\theta$ is a vector of unknown constant parameters, and $\varphi_i$ are known (smooth) function vectors. The control objective is to asymptotically track a given reference $y_r(t)$ with $x_1$. All derivatives of $y_r(t)$ are assumed to be known.

The adaptive backstepping design procedure is similar to the normal backstepping procedure, only this time a control law (static

part) and a *parameter update law* (dynamic part) are designed along with a control Lyapunov function to guarantee global stability. The control law makes use of a parameter estimate $\hat{\theta}$, which is constantly adapted by the dynamic parameter update law. Furthermore, the control Lyapunov function now contains an extra term that penalizes the parameter estimation error $\tilde{\theta} = \theta - \hat{\theta}$.

*Theorem III.1 (Adaptive Backstepping Method)*: To stabilize the system Eq. (15) an error variable is introduced for each state

$$z_i = x_i - y_r^{(i-1)} - \alpha_{i-1} \tag{16}$$

along with a virtual control law

$$\alpha_i\left(\bar{x}_i, \hat{\theta}, \bar{y}_r^{(i-1)}\right) = -c_i z_i - z_{i-1} - \omega_i^T \hat{\theta} + \sum_{k=1}^{i-1}\left(\frac{\partial \alpha_{i-1}}{\partial x_k} x_{k+1} + \frac{\partial \alpha_{i-1}}{\partial y_r^{(k-1)}} y_r^{(k)}\right) + \frac{\partial \alpha_{i-1}}{\partial \hat{\theta}} \Gamma \tau_i + \sum_{k=2}^{i-1} \frac{\partial \alpha_{k-1}}{\partial \hat{\theta}} \Gamma \omega_i z_k \tag{17}$$

for $i = 1, 2, \ldots, n$, where the tuning function $\tau_i$ and the regressor vectors $\omega_i$ are defined as

$$\tau_i\left(\bar{x}_i, \hat{\theta}, \bar{y}_r^{(i-1)}\right) = \tau_{i-1} + \omega_i z_i \tag{18}$$

and

$$\omega_i\left(\bar{x}_i, \hat{\theta}, \bar{y}_r^{(i-2)}\right) = \varphi_i - \sum_{k=1}^{i-1} \frac{\partial \alpha_{i-1}}{\partial x_k} \varphi_k \tag{19}$$

where $\bar{x}_i = (x_1, \ldots, x_i)$ and $\bar{y}_r^{(i)} = (y_r, \dot{y}_r, \ldots, y_r^{(i)})$. $c_i > 0$ are design constants. With these new variables the control and adaptation laws can be defined as

$$u = \frac{1}{\beta(x)}\left[\alpha_n\left(x, \hat{\theta}, \bar{y}_r^{(n-1)}\right) + y_r^{(n)}\right] \tag{20}$$

and

$$\dot{\hat{\theta}} = \Gamma \tau_n\left(x, \hat{\theta}, \bar{y}_r^{(n-1)}\right) = \Gamma W z \tag{21}$$

where $\Gamma = \Gamma^T > 0$ is the adaptation gain matrix and $W$ the regressor matrix

$$W(z, \hat{\theta}) = [\omega_1, \ldots, \omega_i] \tag{22}$$

The control law Eq. (20) together with the update law Eq. (22) renders the derivative of the Lyapunov function

$$V = \frac{1}{2}\sum_{i=1}^{n} z_i^2 + \frac{1}{2}\tilde{\theta}^T \Gamma^{-1} \tilde{\theta} \tag{23}$$

negative definite and thus this adaptive controller guarantees global boundedness of $x(t)$ and asymptotically tracking of a given reference $y_r(t)$ with $x_1$.

Proof of this theorem can be found in Sec. 4.3 of [15].

### C. Constraint Handling

The standard adaptive backstepping procedure as has been discussed so far has a number of drawbacks.

1) The analytic calculation of the virtual control derivatives is tedious, especially for large systems;

2) the procedure can only handle systems that can be transformed into a lower-triangular form;

3) constraints on the inputs and states are not taken into account.

The third drawback can be a major problem when designing for flight control, because the actuators of an aircraft have rate, bandwidth, and magnitude constraints. When the control signal demanded by the backstepping controller cannot be generated by the actuators, that is, the actuators saturate, stability can no longer be guaranteed. The problem becomes worse when online parameter update laws are used, because these tend to be aggressive while seeking the desired tracking performance. Because the desired control signal is not achieved during saturation, the tracking error will increase. Because this tracking error is not just the result from the parameter estimation error, the update law may "unlearn" during these saturation periods.

In [23–25] a method is proposed that fits within the recursive adaptive backstepping design procedure and deals with the constraints on both the control variables and the intermediate states used as virtual controls. An additional advantage of the method is that it also eliminates the two other drawbacks of the adaptive backstepping method, that is, the time consuming analytic computation of virtual control signal derivatives and the restriction to nonlinear systems of a lower-triangular form.

The proposed method extends the adaptive backstepping framework in two ways.

1) Command filters are used to eliminate the analytic computation of the time derivatives of the virtual controls. The command filters are designed as linear, stable, low-pass filters with unity gain from its input to its output. The inputs of these filters are the desired (virtual) control signals and the outputs are the actual (virtual) control signal and its time derivative. Using command filters to calculate the virtual control derivatives, it is still possible to prove stability in the sense of Lyapunov in the absence of constraints on the control input and state variables.

2) A stable parameter estimation process is ensured even when constraints on the control variables and states are in effect. During these periods the tracking error may increase because the desired control signal cannot be implemented due to these constraints imposed on the system. In this case the desired response is too aggressive for the system to be feasible and the primary goal is to maintain stability of the online function approximation. The command filters keep the control signal and the state variables within their mechanical constraints and operating limits, respectively. The effect these constraints have on the tracking errors can be estimated and this effect can be implemented in modified tracking error definitions. These modified tracking errors are only the result of parameter estimation errors as the effect of the constraints on the control input and state variables has been removed. These modified tracking errors can thus be used by the parameter update laws to ensure a stable estimation process.

The command filtered adaptive backstepping approach is summarized in the following theorem.

*Theorem III.2 (Constrained Adaptive Backstepping Method)*: For the parameter strict-feedback system Eq. (15) the tracking errors are again defined as

$$z_i = x_i - y_r^{(i-1)} - \alpha_{i-1} \tag{24}$$

for $i = 1, 2, \ldots, n$. The nominal or desired virtual control laws can be defined as

$$\alpha_i^0 = -c_i z_i - \bar{z}_{i-1} - \varphi_i^T \hat{\theta} + \dot{\alpha}_{i-1} - \chi_{i+1}, \qquad i = 1, 2, \ldots, n-1 \tag{25}$$

where

$$\bar{z}_i = z_i - \chi_i, \qquad i = 1, 2, \ldots, n \tag{26}$$

are the modified tracking errors and where

$$\dot{\chi}_i = -c_i \chi_i + \left(\alpha_i - \alpha_i^0\right), \qquad i = 1, 2, \ldots, n-1 \tag{27}$$

are the filtered versions of the effect of the state constraints on the tracking errors $z_i$. The nominal virtual control signals $\alpha_i^0$ are filtered to produce the magnitude, rate, and bandwidth limited virtual control signals $\alpha_i$ and its derivatives $\dot{\alpha}_i$ that satisfy the limits imposed on the state variables. This command filter can for instance be chosen as [25]

$$\begin{bmatrix}\dot{q}_1(t)\\ \dot{q}_2(t)\end{bmatrix} = \begin{bmatrix} q_2 \\ 2\zeta\omega_n\left[S_R\left(\frac{\omega_n^2}{2\zeta\omega_n}[S_M(\alpha_i^0)-q_1]\right)-q_2\right]\end{bmatrix}\begin{bmatrix}\alpha_i\\ \dot{\alpha}_i\end{bmatrix} = \begin{bmatrix}q_1\\ q_2\end{bmatrix} \tag{28}$$

where $S_M(\cdot)$ and $S_R(\cdot)$ represent the magnitude and rate limit functions, respectively. These saturation functions are defined similarly as

$$S_M(x) = \begin{cases} M & \text{if } x \geq M \\ x & \text{if } |x| < M \\ -M & \text{if } x \leq -M \end{cases}$$

The effect of implementing the achievable virtual control signals instead of the desired ones is estimated by the $\chi_i$ filters. With these filters the modified tracking errors $\bar{z}_i$ can be defined. It can be seen from Eq. (26) that when the limitations on the states are not in effect the modified tracking error converges to the tracking error. The nominal control law is defined in a similar way as

$$u^0 = \frac{1}{\beta}\left(-c_n z_n - \bar{z}_{n-1} - \varphi_n^T\hat{\theta} + \dot{\alpha}_{n-1} + y_r^{(n)}\right) \tag{29}$$

which is again filtered to generate the magnitude, rate, and bandwidth limited control signal $u$. The effect of implementing the limited control law instead of the desired one can again be estimated with

$$\dot{\chi}_n = -c_n\chi_n + \beta(u - u^0) \tag{30}$$

Finally, the update law that now uses the modified tracking errors is defined as

$$\dot{\hat{\theta}} = \Gamma\sum_{i=1}^{n}\varphi_i\bar{z}_i \tag{31}$$

The resulting control law will render the derivative of the control Lyapunov function

$$V = \frac{1}{2}\sum_{i=1}^{n}\bar{z}_i^2 + \frac{1}{2}\tilde{\theta}^T\Gamma^{-1}\tilde{\theta} \tag{32}$$

negative definite, which means that the closed-loop system is asymptotically stable. Using the theorem by LaSalle [37] and Yoshizawa [38] it follows that the modified tracking errors converge to zero:

$$\lim_{t\to\infty}\bar{z}_i(t) = 0$$

A scheme of the command filtered adaptive backstepping approach is shown in Fig. 1.

*Proof*: Using the definitions Eqs. (24–27) and denoting $x_0 \equiv \alpha_0 \equiv 0$, $x_{n+1} \equiv \beta(x)u$, the derivative of the modified tracking error variable $z_i$, $i = 1, \ldots, n$, becomes

$$\begin{aligned}\dot{\bar{z}}_i &= \dot{x}_i - y_r^{(i)} - \dot{\alpha}_{i-1} - \dot{\chi}_i = z_{i+1} + \alpha_i + \varphi_i^T\theta - \dot{\alpha}_{i-1} + c_i\chi_i \\ &\quad - \left(\alpha_i - \alpha_i^0\right) = \bar{z}_{i+1} - c_i\bar{z}_i - \bar{z}_{i-1} + \varphi_i^T\tilde{\theta}\end{aligned} \tag{33}$$

The choice of control law in the last step guarantees that $\bar{z}_{n+1} \equiv 0$. The closed-loop error system can therefore be expressed as

$$\begin{aligned}\dot{\bar{z}}_1 &= -c_1\bar{z}_1 + \bar{z}_2 + \varphi_1^T\tilde{\theta} \\ \dot{\bar{z}}_2 &= -c_2\bar{z}_2 - \bar{z}_1 + \bar{z}_3 + \varphi_2^T\tilde{\theta} \\ &\vdots \\ \dot{\bar{z}}_{n-1} &= -c_{n-1}\bar{z}_{n-1} - \bar{z}_{n-2} + \bar{z}_n + \varphi_{n-1}^T\tilde{\theta} \\ \dot{\bar{z}}_n &= -c_n\bar{z}_n - \bar{z}_{n-1} + \varphi_n^T\tilde{\theta} \\ \dot{\hat{\theta}} &= \Gamma\textstyle\sum_{i=1}^{n}\varphi_i\bar{z}_i\end{aligned} \tag{34}$$

The control Lyapunov function Eq. (32) can now be used to prove stability and convergence. The derivative along the solutions of Eq. (33) is

$$\begin{aligned}\dot{V} &= \sum_{i=1}^{n}\bar{z}_i\dot{\bar{z}}_i - \dot{\hat{\theta}}^T\Gamma^{-1}\tilde{\theta} = \sum_{i=1}^{n}\left[-c_i\bar{z}_i^2 + \bar{z}_i\varphi_i^T\tilde{\theta} - \bar{z}_i\varphi_i^T\tilde{\theta}\right] \\ &= -\sum_{i=1}^{n}c_i\bar{z}_i^2 < 0 \\ &\bar{z}_i \neq 0\end{aligned} \tag{35}$$

This means that during periods when there is no saturation, $\bar{z}_i$ will converge to $z_i$. If saturation does occur the actual tracking errors $z_i$ may increase, but the modified tracking errors $\bar{z}_i$ will still converge to zero and the estimation process remains stable, because it uses these modified tracking error definitions.

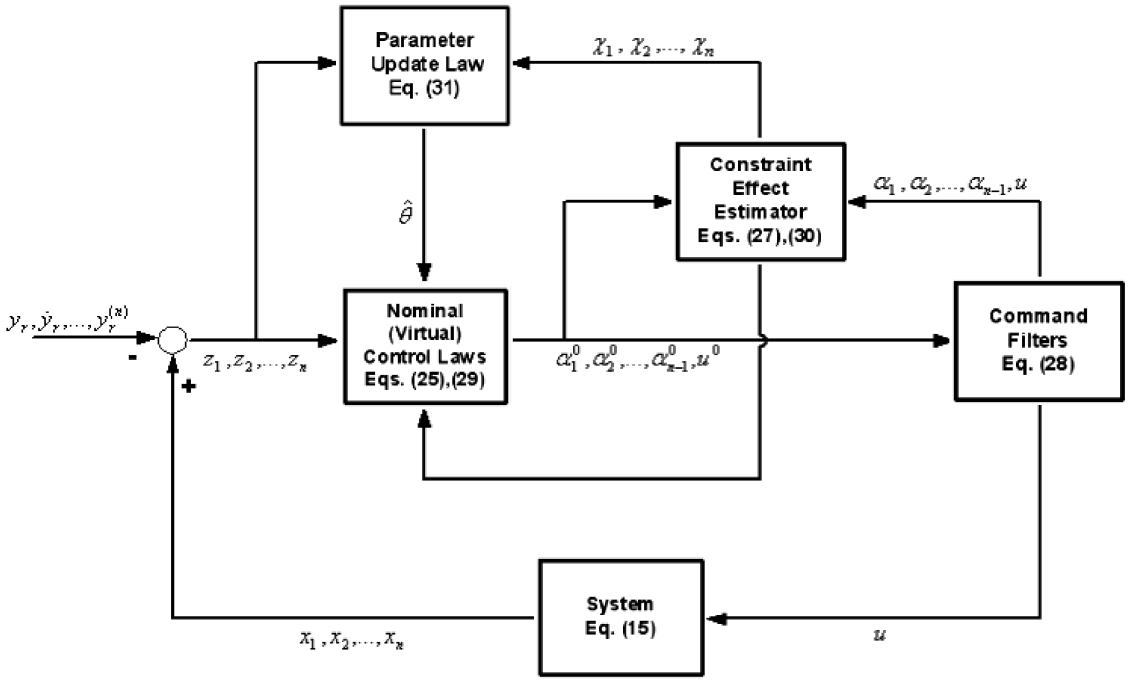


**Fig. 1 Command filtered adaptive backstepping control scheme.**

## IV. Flight Control Law

The command filtered adaptive backstepping approach discussed in the preceding section is now used to design a control law for the F-16/MATV model. Following [29,39], common choices for the variables to be controlled are $\alpha$, $\beta$ (which should be kept at zero degrees), and $p_s$.[§] An obvious extension adopted here is the control of the total airspeed $V_T$. Summarizing, the task of the control system is to make sure that

$$V_T = V^{\text{ref}} \qquad \alpha = \alpha^{\text{ref}} \qquad \beta = 0 \qquad p_s = p_s^{\text{ref}}$$

is an asymptotic equilibrium. The available control variables are the thrust $F_T$, the control surface deflections $\delta_e$, $\delta_a$, $\delta_r$, and the engine nozzle deflections $\delta_{\tau_y}$, $\delta_{\tau_z}$. All control variables are subject to rate, magnitude, and bandwidth constraints as given in Table 1.

### A. Assumptions for Control

Because command filters will be used to calculate the derivatives of the virtual control variables, the system to be controlled no longer has to be of a lower-triangular form. This means that the force equations (1) do not have to be changed. However, the complete aircraft dynamics (1–3) must still be affine in the control variables $F_T$, $\delta_e$, $\delta_a$, $\delta_r$, $\delta_{\tau_y}$, and $\delta_{\tau_z}$.

To achieve this, the deflections of the engine nozzle in the moment equations (3) will have to be linearized. This linearization is expected to be valid, because the maximum deflection of the F-16/MATV's engine nozzle is only 17 deg in all directions. The engine nozzle deflections will not be used for control in the force equations. (1), because $F_t$, $q_s$, and $r_s$ have already been selected as the (virtual) control variables in this step. This means linearization of the terms containing these deflections is not necessary in these equations. Because $\dot{\alpha}$, $L_T$, $M_T$, and $N_T$ are the only parts of the moment equations (3) that contain engine nozzle deflections, it must be made sure that $\delta_{\tau_y}$ and $\delta_{\tau_z}$ appear affine in these expressions:

$$\dot{\alpha} = q_s - p_s \tan\beta + \frac{1}{mV_T \cos\beta}[-L - F_T(\sin\alpha + \delta_{\tau_y}\cos\alpha) + mg_3] \tag{36}$$

$$L_T = -F_T z_T \delta_{\tau_z} \qquad M_T = F_T(x_T \delta_{\tau_y} + z_T) \qquad N_T = F_T x_T \delta_{\tau_z} \tag{37}$$

Furthermore, it is useful to split the aerodynamic forces and moments in parts dependent and independent of the control variables

$$\bar{X} = \bar{X}' + \bar{X}_{\delta_e}\delta_e \qquad \bar{Y} = \bar{Y}' + \bar{Y}_{\delta_a}\delta_a + \bar{Y}_{\delta_r}\delta_r$$
$$\bar{Z} = \bar{Z}' + \bar{Z}_{\delta_e}\delta_e \qquad \bar{L} = \bar{L}' + \bar{L}_{\delta_e}\delta_e + \bar{L}_{\delta_a}\delta_a + \bar{L}_{\delta_r}\delta_r$$
$$\bar{M} = \bar{M}' + \bar{M}_{\delta_e}\delta_e \qquad \bar{N} = \bar{N}' + \bar{N}_{\delta_e}\delta_e + \bar{N}_{\delta_a}\delta_a + \bar{N}_{\delta_r}\delta_r$$

Each of these aerodynamic forces and moments is built up of different nondimensional aerodynamic coefficients. The values of these nondimensional coefficients were obtained from wind tunnel experiments [34] and have been stored in lookup tables as a function of the current flight condition (e.g., $\alpha$ and $\beta$). As an example the expression for the aerodynamic force coefficient along the body $X$ axis, $C_{X_T}$, may be written as

$$C_{X_T} = C_X(\alpha,\beta,\delta_e) + \delta C_{X_{\text{LEF}}}\left(1 - \frac{\delta_{\text{LEF}}}{25}\right) + \frac{q\bar{c}}{2V_T}\left[C_{X_q}(\alpha) + \delta C_{X_{q_{\text{LEF}}}}(\alpha)\left(1 - \frac{\delta_{\text{LEF}}}{25}\right)\right]$$

where

$$\delta C_{X_{\text{LEF}}} = C_{X_{\text{LEF}}}(\alpha,\beta) - C_X(\alpha,\beta,\delta_e = 0 \text{ deg})$$

For the F-16/MATV model all these lookup tables are 1-D, 2-D, or 3-D. In principle, all coefficients need to be affine in the control variables when designing a control law, which is, for example, not the case for the coefficient $C_X(\alpha,\beta,\delta_e)$. To solve this problem, $C_X(\alpha,\beta,\delta_e)$ can be approximated as

$$C_X(\alpha,\beta,\delta_e) = C_{X_0}(\alpha,\beta) + C_{X_{\delta_e}}(\alpha)\delta_e \tag{38}$$

However, because command filters are used to calculate the actual control variables a more accurate approximation can be used

$$C_X(\alpha,\beta,\delta_e) = C_{X_0}(\alpha,\beta) + C_{X_{\delta_e}}(\alpha,\delta_e)\delta_e \tag{39}$$

since the current $\delta_e$ is already available from the corresponding command filter. All other coefficients depending nonaffine on control surface deflections may be approximated in a similar manner.

### B. Control Design Procedure

It is quite trivial to extend the command filtered adaptive backstepping approach discussed in Sec. III to multivariable systems, as will be demonstrated during the design of the flight control system in this section. The design procedure is initiated by defining the tracking errors as

$$\begin{bmatrix} z_{11} \\ z_{12} \\ z_{13} \end{bmatrix} = \begin{bmatrix} V_T - V^{\text{ref}} \\ \alpha - \alpha^{\text{ref}} \\ \beta - \beta^{\text{ref}} \end{bmatrix} \tag{40}$$

and

$$\begin{bmatrix} z_{21} \\ z_{22} \\ z_{23} \end{bmatrix} = \begin{bmatrix} p_s - p_s^{\text{ref}} \\ q_s - q_s^{\text{ref}} \\ r_s - r_s^{\text{ref}} \end{bmatrix} \tag{41}$$

where $q_s^{\text{ref}}$ and $r_s^{\text{ref}}$ are virtual control laws, which are defined later. Applying the command filtered adaptive backstepping approach the following nominal (virtual) control laws are found:

$$\begin{bmatrix} F_{T_0} \\ q_{s_0}^{\text{ref}} \\ r_{s_0}^{\text{ref}} \end{bmatrix} = B_1^T\left(-\begin{bmatrix} c_{11}z_{11} \\ c_{12}z_{12} \\ c_{13}z_{13} \end{bmatrix} + \begin{bmatrix} \dot{V}^{\text{ref}} \\ \dot{\alpha}^{\text{ref}} \\ \dot{\beta}^{\text{ref}} \end{bmatrix} - F_1\right) - \begin{bmatrix} 0 \\ \chi_{22} \\ \chi_{23} \end{bmatrix} \tag{42}$$

and

$$\begin{bmatrix} \delta_{e_0} \\ \delta_{a_0} \\ \delta_{r_0} \\ \delta_{\tau_{y_0}} \\ \delta_{\tau_{z_0}} \end{bmatrix} = B_2^{\dagger}\left(-\begin{bmatrix} c_{21}z_{21} \\ c_{22}z_{22} \\ c_{23}z_{23} \end{bmatrix} + \begin{bmatrix} \dot{p}_s^{\text{ref}} \\ \dot{q}_s^{\text{ref}} \\ \dot{r}_s^{\text{ref}} \end{bmatrix} - F_2 - B_1^T\begin{bmatrix} 0 \\ \bar{z}_{12} \\ \bar{z}_{13} \end{bmatrix}\right) \tag{43}$$

where $B_2^{\dagger}$ is the pseudoinverse of matrix $B_2$ used to distribute the desired control signals over the actual inputs. The pseudoinverse is the most simple way of control allocation [40,41]. The expressions for the vectors $F_1$, $F_2$ and the matrices $B_1$, $B_2$ can be found in the Appendix. Note that the estimates of the aerodynamic forces and moments are used by the control law, for example, the estimate $\hat{X}$ of $\bar{X}$. Also note that for these aircraft dynamics $B_2$ cannot lose rank due to the thrust vector terms, even when all estimates were equal to zero. If the possibility of rank loss would exist for $B_2$, parameter projection methods [42] could be applied to make sure that the inverse of $B_2$ does not become singular. The nominal (virtual) laws Eqs. (42) and

[§]Rolling around the stability reference frame $x$ axis is known as *velocity vector rolling*. The advantage of rolling around the stability $x$ axis instead of the body $x$ axis is that it reduces the amount of sideslip during the roll ($\alpha$ and $\beta$ remain constant during the velocity vector roll), especially at high angles of attack.

(43) can be filtered to produce the actual control signals and their derivatives. Magnitude, rate, and bandwidth limits can also be applied using these command filters. The influence of these state and input constraints on the tracking errors is expressed with stable linear $\chi$ filters:

$$\begin{bmatrix} \dot{\chi}_{11} \\ \dot{\chi}_{12} \\ \dot{\chi}_{13} \end{bmatrix} = -\begin{bmatrix} c_{11}\chi_{11} \\ c_{12}\chi_{12} \\ c_{13}\chi_{13} \end{bmatrix} + B_1 \begin{bmatrix} F_T - F_{T_0} \\ q_s^{\text{ref}} - q_{s_0}^{\text{ref}} \\ r_s^{\text{ref}} - r_{s_0}^{\text{ref}} \end{bmatrix} \tag{44}$$

and

$$\begin{bmatrix} \dot{\chi}_{21} \\ \dot{\chi}_{22} \\ \dot{\chi}_{23} \end{bmatrix} = -\begin{bmatrix} c_{21}\chi_{21} \\ c_{22}\chi_{22} \\ c_{23}\chi_{23} \end{bmatrix} + B_2 \begin{bmatrix} \delta_e - \delta_{e_0} \\ \delta_a - \delta_{a_0} \\ \delta_r - \delta_{r_0} \\ \delta_{\tau_y} - \delta_{\tau_{y_0}} \\ \delta_{\tau_z} - \delta_{\tau_{z_0}} \end{bmatrix} \tag{45}$$

With these filters the modified tracking errors that will be used by the parameter update laws to ensure a stable estimation process are defined as

$$\begin{bmatrix} \bar{z}_{11} \\ \bar{z}_{12} \\ \bar{z}_{13} \end{bmatrix} = \begin{bmatrix} z_{11} - \chi_{11} \\ z_{12} - \chi_{12} \\ z_{13} - \chi_{13} \end{bmatrix} \tag{46}$$

and

$$\begin{bmatrix} \bar{z}_{21} \\ \bar{z}_{22} \\ \bar{z}_{23} \end{bmatrix} = \begin{bmatrix} z_{21} - \chi_{21} \\ z_{22} - \chi_{22} \\ z_{23} - \chi_{23} \end{bmatrix} \tag{47}$$

Together with the correct choice of parameter update laws for the uncertain aerodynamic forces and moments this adaptive backstepping control law will globally stabilize the system.

## V. Parameter Update Laws

If all aerodynamic forces and moments are known and available, the backstepping control law designed in the previous section can stabilize the F-16/MATV aircraft model. However, it is assumed that they are not known and thus the control design has to be extended with parameter update laws to estimate the components of the aerodynamic forces ($\bar{X}, \bar{Y}, \bar{Z}$) and moments ($\bar{L}, \bar{M}, \bar{N}$). The design of such update laws will again be demonstrated for the aerodynamic force $\bar{X}$, which is defined as

$$\bar{X} = \bar{q}SC_{X_T} \tag{48}$$

where combining Eqs. (38) and (39), $C_{X_T}$ is now equal to

$$C_{X_T} = C_{X_0}(\alpha, \beta) + C_{X_{\delta_e}}(\alpha, \delta_e)\delta_e + \delta C_{X_{\text{LEF}}}\left(1 - \frac{\delta_{\text{LEF}}}{25}\right) + \frac{q\bar{c}}{2V_T}\left[C_{X_q}(\alpha) + \delta C_{X_{q_{\text{LEF}}}}(\alpha)\left(1 - \frac{\delta_{\text{LEF}}}{25}\right)\right] \tag{49}$$

To be able to reduce the number of update laws needed, the expressions for the aerodynamic moments and coefficients are simplified by incorporating the data of the leading edge flap in the other data tables. This is done by eliminating the Mach dependency of the flap position and neglecting its actuator dynamics, after which the data can be incorporated into the correspondent information on $\alpha$ [43]. The update laws should be able to compensate for the approximation error between the full aerodynamic model and its simplified version. For $C_{X_T}$ this simplification leads to

$$C_{X_T} = C_{X_0}(\alpha, \beta) + \frac{q\bar{c}}{2V_T}C_{X_q}(\alpha) + C_{X_{\delta_e}}(\alpha, \delta_e)\delta_e \tag{50}$$

Each of these aerodynamic coefficients is an unknown nonlinear function depending on several aircraft variables. However, adaptive backstepping is only applicable to systems with *parametric* uncertainties, which means that these functions have to be transformed into some sort of parameterized form so that they can be estimated. RBF neural networks can be used to approximate each aerodynamic coefficient and they can be written in a parameterized form that can be used by the backstepping update laws. Neural networks are known as universal approximators and can be used to approximate continuous functions at any arbitrary accuracy as long as the network is large enough [44,45]. A special type of RBF network is the B-spline network [27,28,46], which uses B splines of any desired order as basis functions. B-spline basis functions can be defined in the following way [47].

*Definition V.1 (B-spline basis function)*: Let $U$ be a set of $m + 1$ nondecreasing numbers, $u0 \le u2 \le u3 \le \cdots \le u_m$. The $u_i$s are called knots, the set $U$ the knot vector, and the half-open interval $[u_i, u_{i+1})$ the $i$th knot span. Note that because some $u_i$s may be equal, some knot spans may not exist. If a knot $u_i$ appears $k$ times (i.e., $u_i = u_{i+1} = \cdots = u_{i+k-1}$), where $k > 1$, $u_i$ is a multiple knot of multiplicity $k$, written as $u_i(k)$. Otherwise, if $u_i$ appears only once, it is a simple knot. If the knots are equally spaced (i.e., $u_{i+1} - u_i$ is a constant for $0 \le i \le m - 1$), the knot vector or the knot sequence is said uniform; otherwise, it is nonuniform.

The knots can be considered as division points that subdivide the interval $[u_0, u_m]$ into knot spans. All B-spline basis functions are supposed to have their domain on $[u_0, u_m]$. To define B-spline basis functions, we need one more parameter, the degree of these basis functions, $p$. The $i$th B-spline basis function of degree $p$, written as $N_{i,p}(u)$, is defined recursively as follows:

$$F_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \le u < u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$F_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i}F_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}}F_{i+1,p-1}(u)$$

This formula is usually referred to as the Cox–de Boor recursion formula.

Because of the interesting properties of B-spline basis functions, B-spline neural networks have several characteristics that make them very suitable for online adaptive control [27,28]:

1) Because only a small number of B-spline basis functions is nonzero at any given time step, the weight updating scheme is local. This has the advantage that only a few update laws are used at the same time, resulting in a lower computational expense. Another advantage is that the network retains information of all flight conditions, because the local adaptation does not interfere with points outside the closed neighborhood. This means the estimator has memory capabilities and hence learns instead of simply adapting the weights.

2) The spline outputs are always positive and normalized, which provides numerical stability.

Each unknown nondimensional coefficient will be approximated using a B-spline neural network. For instance, the derivative coefficient $C_{X_q}$ which only depends on angle of attack $\alpha$ is approximated as

$$C_{X_q}(\alpha) = \sum_{i=1}^{n} w_i F_i(\alpha) \tag{51}$$

where $w_i$ are the weights and $F_i(\alpha)$ are the B-spline basis functions. The expression can also be written in a parametric form that can be used by the adaptive backstepping update laws:

$$C_{X_q} = \varphi_{C_{X_q}}^T(\alpha)\theta_{C_{X_q}} \tag{52}$$

where the known function $\varphi_{C_{X_q}}(\alpha)$ and the unknown parameter vector $\theta_{C_{X_q}}$ are defined as

$$\varphi_{C_{X_q}}(\alpha) = [F_1, F_2, \ldots, F_n]^T \qquad \theta_{C_{X_q}} = [w_1, w_2, \ldots, w_n]^T$$

The following update law for the unknown parameter vector $\theta_{C_{X_q}}$ (containing the network weights) can now be defined:

$$\dot{\theta}_{C_{X_q}} = \Gamma_{C_{X_q}} \varphi_{C_{X_q}}(\alpha) \bar{q} S \frac{q\bar{c}}{2V_T} \left( \frac{1}{m} \begin{bmatrix} \cos\alpha\cos\beta \\ -\frac{\sin\alpha}{V_T\cos\beta} \\ -\frac{\cos\alpha\sin\beta}{V_T} \end{bmatrix}^T \begin{bmatrix} \bar{z}_{11} \\ \bar{z}_{12} \\ \bar{z}_{13} \end{bmatrix} - \frac{\sin\alpha}{mV_T\cos\beta} \begin{bmatrix} r_s \\ 0 \\ -p_s \end{bmatrix}^T \begin{bmatrix} \bar{z}_{21} \\ \bar{z}_{22} \\ \bar{z}_{23} \end{bmatrix} \right) \tag{53}$$

where the update law gains $\Gamma_{C_{X_q}} = \Gamma_{C_{X_q}}^T > 0$. Update laws can be defined in a similar way for the two other nondimensional components of $C_{X_T}$, and the other aerodynamic force and moment coefficients. The advantage of using the (B-spline) neural networks inside of the update laws is that the learning process benefits from the strong convergence capabilities of the adaptive backstepping framework. With the definition of the update laws, the control design process that was started in Sec. III is finished. A control Lyapunov function similar to Eq. (32) can be defined and its derivative can be shown to be negative definite, which means the modified tracking errors converge to zero.

The resulting adaptive flight control law can stabilize the aircraft model even in the presence of large changes in the aerodynamic coefficients. However, if an aircraft suffers structural damage, the aerodynamic force and moment coefficients may become dependent on more aircraft states. This is because damage is usually asymmetric, that is, exactly the same and simultaneous damage on both sides of the aircraft is assumed to be improbable. Because the output depends on additional parameters after the failure, the number of inputs of the B-spline network will also have to be larger to be able to estimate the new input/output relations. If this is not the case, the approximation becomes inaccurate and likely time variant.

In [48] some specific asymmetric structural failures are discussed, that is, wing, fuselage, and vertical stabilizer damage. It is concluded that for each failure the aircraft aerodynamic characteristics become more coupled in case of an asymmetric failure. This means that all aerodynamic coefficients become dependent on all longitudinal and lateral aircraft states. Some failures (wing damage) will cause stronger coupling than others (fuselage damage), because it directly depends on the measure of asymmetry introduced into the aerodynamic characteristics of the aircraft. This means the B-spline networks will have to have all these aircraft states as inputs to accurately estimate the coefficients after failure. Because each coefficient is estimated by at least one B-spline network, this means that during online estimation each coefficient requires at least $k^n$ update laws to be active at the same time, where $k$ is the order of the B splines used in the network and $n$ the number of states. To provide enough smoothness, at least third order B splines will have to be used. In other words, the computational power required will make it impossible to adapt the weights online. For this reason only symmetric structural failures will be considered in the simulations of the next section, where the number of aircraft states the aerodynamic coefficients depend on does not change.

**Table 2 Command filter parameters**

| Command variable | $\omega_n$, rad/s | Mag. limit | Rate limit |
|---|---|---|---|
| $F_T$ | 2 | [1000, 100,000] N | ±40, 000 N/s |
| $q_s$ | 10 | ±35 deg/s | —— |
| $r_s$ | 10 | ±20 deg/s | —— |
| $\delta_e$ | 40.4 | ±25 deg | ±60 deg/s |
| $\delta_a$ | 40.4 | ±21.5 deg | ±80 deg/s |
| $\delta_r$ | 40.4 | ±30 deg | ±120 deg/s |
| $\delta_{\tau_y}$ | 40.4 | ±20 deg | ±50 deg/s |
| $\delta_{\tau_z}$ | 40.4 | ±20 deg | ±50 deg/s |

## VI. Tuning the Controller

### A. Control Law Gains

The easiest way of tuning the constrained adaptive backstepping flight control law is by starting with the control law gains of an equivalent backstepping flight control law without the update laws. There are no real guidelines for tuning a backstepping controller. Usually the controller gains of the second step are selected larger than the gains in the first step to get a nicer response, but in fact the Lyapunov stability theory only requires the gains to be larger than zero. In [39] the dynamics of the nonlinear system model are linearized around a suitable operating point and the gains are then selected to achieve some desired linear closed-loop behavior around

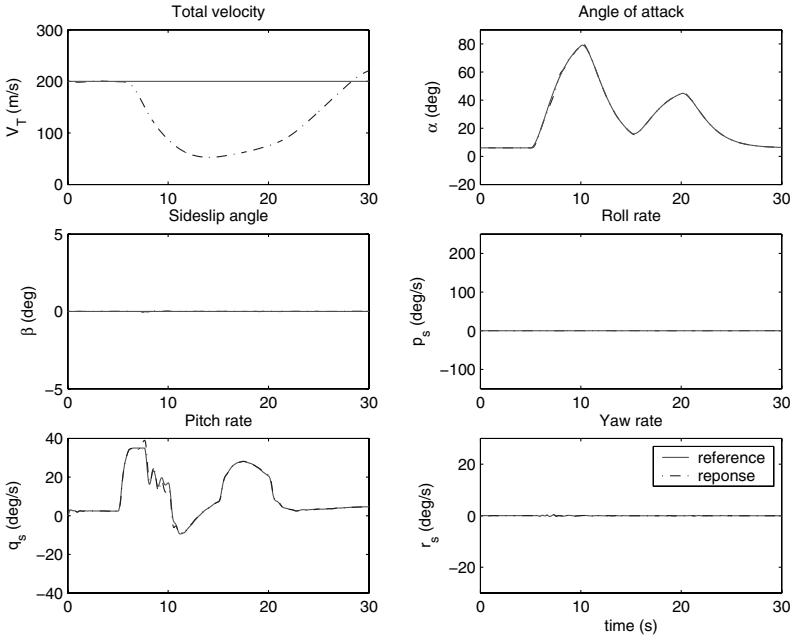


**Fig. 2 Longitudinal simulation—controlled variables and angular rates.**

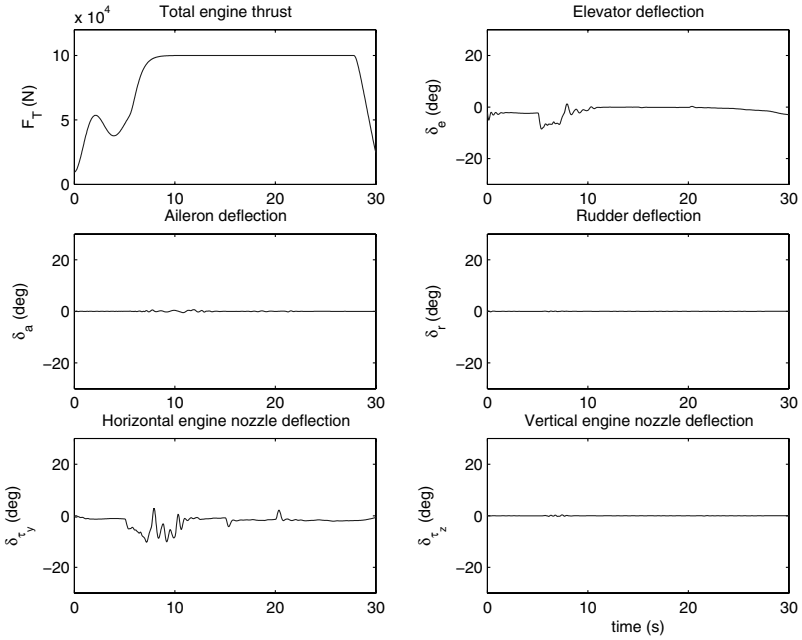


**Fig. 3 Longitudinal simulation—surface deflections and thrust force.**

this operating point. However, getting the desired closed-loop response remains a *trial and error* procedure. The controller gains for both controllers were selected after some experimental tuning and are as follows: $c_{11} = 2$, $c_{12} = 8$, $c_{13} = 8$, $c_{21} = 20$, $c_{22} = 10$, and $c_{23} = 10$. The tuning does not have to be very precise to get a good closed-loop response.

### B. Command Filter Parameters

Now that the controller gains have been specified, the command filter dynamics can be selected, still using the nonadaptive backstepping control law. It is convenient to start with the command filters for the control surface deflections $\delta_e$, $\delta_a$, $\delta_r$, $\delta_{\tau_y}$, and $\delta_{\tau_z}$ at the second control design step, because these filters must have the same dynamics and magnitude/rate limits as the actuators of the F-16/MATV model. It is again natural to select a lower bandwidth for the filters in the first step, that is, the command filters which generate $q_s$ and $r_s$. For the selection of the bandwidths for these variables, the choice of bandwidths in [25] was used as a reference. Furthermore, the magnitude limits on the filters in the second step result in rate limits on the filters in the first step. Additional magnitude limits on $q_s$ and $r_s$ will be used to keep these variables within reasonable limits. The function of these additional magnitude limits is to keep maneuvers comfortable for the pilots and to help prevent saturation of the real control inputs. Table 2 shows the command filter parameters that were selected.

### C. Update Law Gains

Now that the controller gains and command filter parameters have been defined, the update law gains can be selected. Again the theory only requires that the gains should be larger than zero. Selecting small gains results in a slower estimation process, but requires less control power. Large gains will mean a faster estimation process, but will also require more control power. Because the adaptive controller makes use of B-spline networks in the update laws there are other problems to consider. In fact, the gains now specify the learning rate of the B-spline networks, or, in other words, the speed of learning.

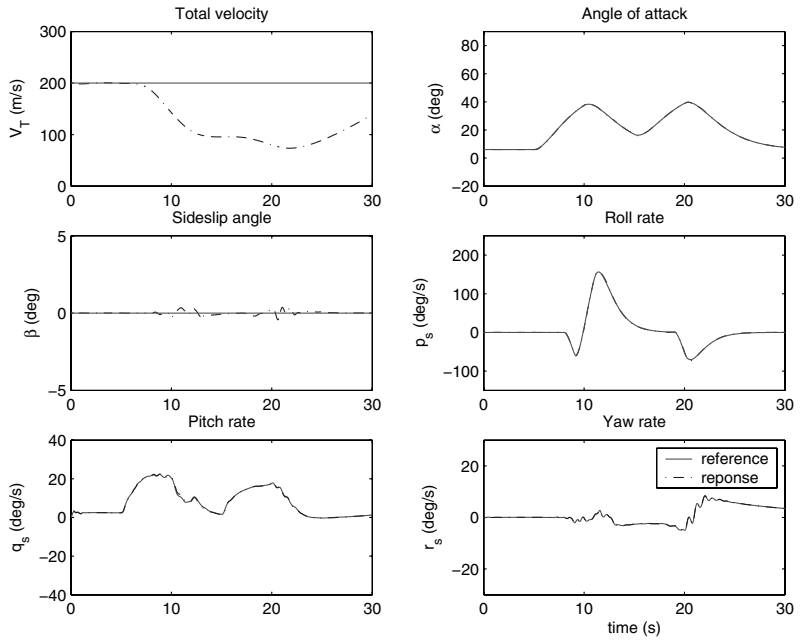


**Fig. 4 Mixed simulation—controlled variables and angular rates.**

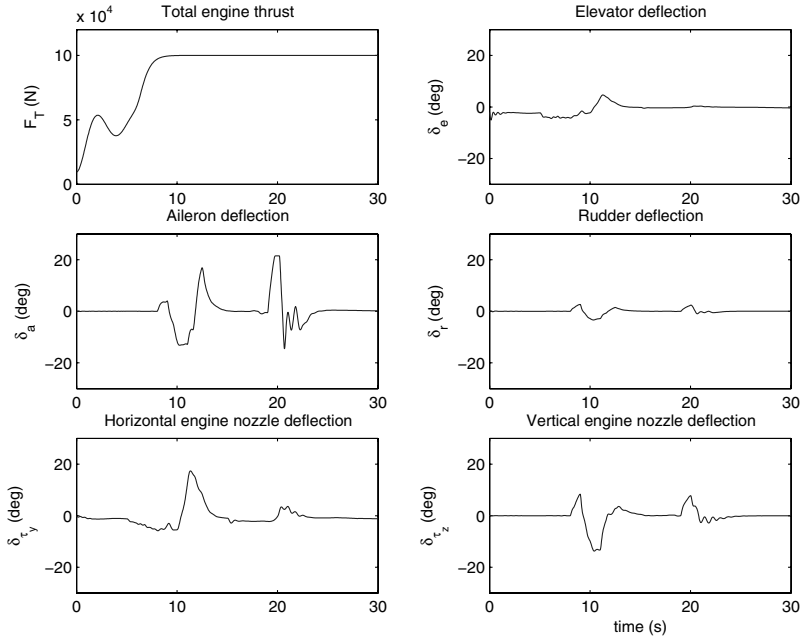


**Fig. 5 Mixed simulation—surface deflections and thrust force.**

Larger learning rates will cause larger changes in the network weights.

With these notions in mind, the tuning of the update law gains for an adaptive backstepping controller remains a trail and error process, just like the tuning of the controller gains. Simulations of an adaptive controller take up a lot of computational time in MATLAB/Simulink, which makes the experimental tuning of the update gains a tedious task. It takes a lot of time to see the influence of a certain change in a particular update gain, especially because there are so many of them for a large, complex system such as the F-16/MATV. This is the main drawback of the constrained adaptive backstepping procedure for such a complex system. The gains that were eventually selected did provide sufficient performance in the investigated failure cases, but a more detailed study of damage scenarios will be necessary to find the true optimal update law gains. Because the failure cases investigated only influence the longitudinal motions, the gains for the longitudinal update laws were selected quite large, while the gains of the lateral update laws were selected much smaller.

## VII. Simulation Results

The constrained adaptive backstepping flight control law designed in the previous sections will now be evaluated in a series of simulations. First, the control law is tested on the nominal (undamaged) F-16/MATV model, to check if performance is satisfactory. After that, structural and actuator damages are introduced in the aircraft model to evaluate the effectiveness of the update laws with the B-spline neural networks.

The controller gains selected for the simulations can be found in Sec. VI.A and Table 2 defines the command filter parameters. Third order B splines are selected to provide enough smoothness during the update process and all knots were spaced every 2.5 deg for $\alpha$, $\beta$, and $\delta_e$ (the lookup tables of the F-16/MATV model are spaced at least every 5 deg).

### A. Initial Simulations

The first series of simulations will test the control law on the nominal F-16/MATV model. The B-spline networks are supplied

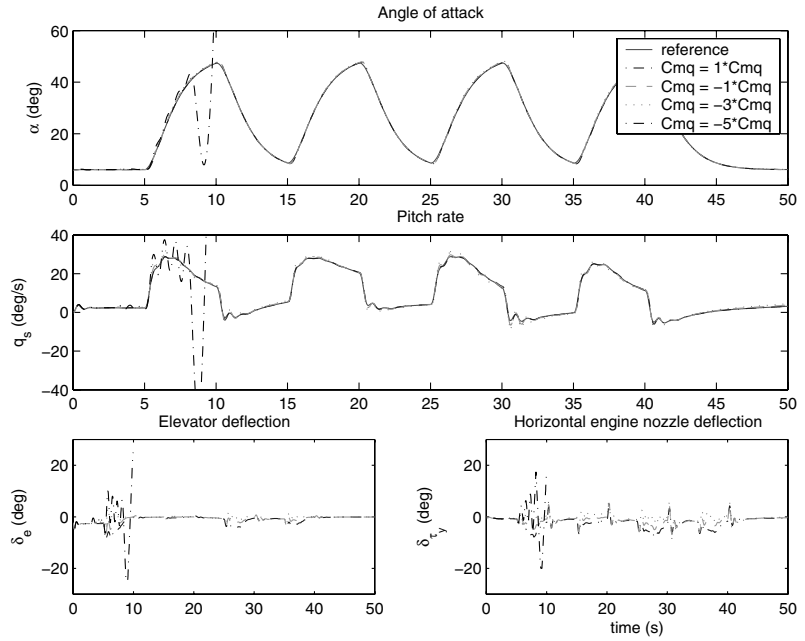


**Fig. 6 CBS controller—simulations with sudden change in damping term $C_{m_q}$ after 3 s.**

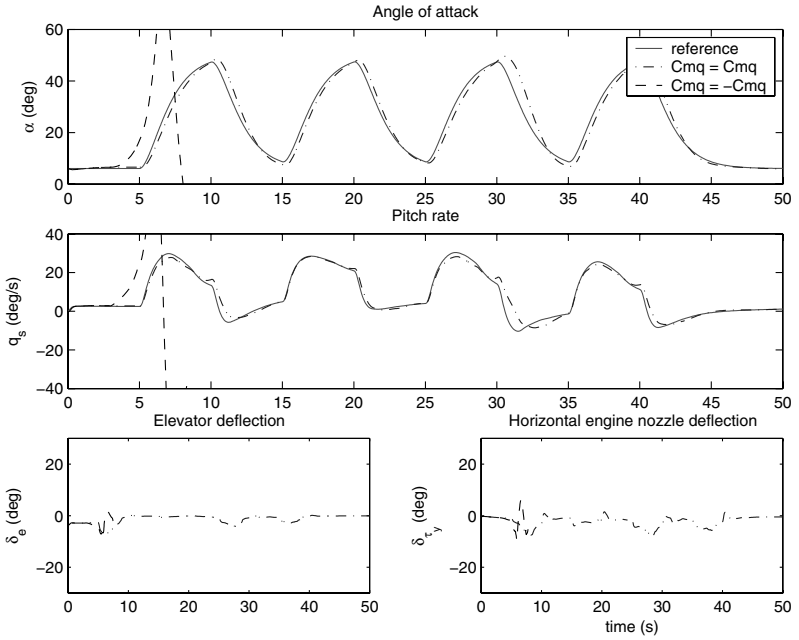


**Fig. 7 Detuned CBS controller—sudden change in damping term $C_{m_q}$.**

with the correct values for the initial flight condition, but will have to estimate the correct values for all other flight conditions online.

The simulations are performed at an altitude of 5000 m at initial velocity of 200 m/s. In the first simulation $\alpha$ must follow a reference signal, while $\beta$ and $p_s$ are kept at zero. $V_T$ will drop during these maneuvers, but will return to its reference value if possible. The results of the simulation can be found in Figs. 2 and 3, where the solid lines represent the reference trajectories and the dashed lines the response. It can be seen that the control law has no problem following the reference trajectories, except for the desired airspeed which cannot be physically maintained during the high angles of attack. Note that when the thrust force becomes larger during the maneuvers, the effectiveness of the engine nozzle deflection becomes much higher. This results in less usage of the elevator, because the pseudoinverse of the control effectiveness matrix is used for control allocation. It can be seen that the pitch rate $q_s$ reaches the magnitude limit of 35 deg/s after about 6 s. During this short period of saturation the tracking error becomes somewhat larger, but the estimation process remains stable because the update laws use the modified tracking errors, where the effects of the saturation of both the pitch rate and the thrust force have been removed.

In the next simulation a maneuver is flown with both lateral and longitudinal command trajectories, to see if the control law does a good job of decoupling the responses. $V_T$ and $\beta$ must again remain constant, while $\alpha$ and $p_s$ follow reference trajectories. A simulation is again performed at 5000 m altitude at an airspeed of 200 m/s; the results have been plotted in Figs. 4 and 5. The results are again satisfactory, despite some actuator saturation during the maneuvers.

It can be concluded that the control law does a good job of decoupling the longitudinal and lateral responses. Furthermore, the modeling simplifications that were made during the control design prove to be valid, because the update laws are able to compensate for any differences between the simplified model and the F-16/MATV model.

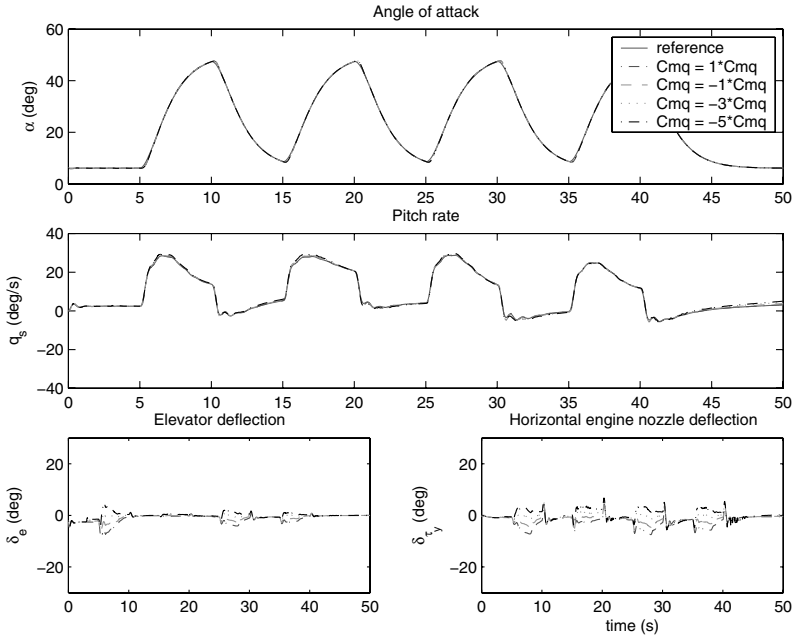


**Fig. 8 CABS controller—simulations with sudden change in damping term $C_{m_q}$ after 3 s.**

**Table 3 Robustness to instant $C_{m_q}$ changes**

| Damping term | CBS controller $(\alpha-\alpha^{\text{ref}})_{\text{rms}}$ | CBS controller $(q_s-q_s^{\text{ref}})_{\text{rms}}$ | CABS controller $(\alpha-\alpha^{\text{ref}})_{\text{rms}}$ | CABS controller $(q_s-q_s^{\text{ref}})_{\text{rms}}$ |
|---|---|---|---|---|
| $C_{m_q}=C_{m_q}$ | 0.2498 | 0.3598 | 0.2447 | 0.2339 |
| $C_{m_q}=-C_{m_q}$ | 0.3054 | 1.7651 | 0.2446 | 0.2495 |
| $C_{m_q}=-3C_{m_q}$ | 0.4683 | 3.4967 | 0.2437 | 0.2874 |
| $C_{m_q}=-5C_{m_q}$ | 12.8038 | 50.5346 | 0.2422 | 0.3547 |

### B. Simulations with Structural Damage

In the previous section it was shown the constrained adaptive backstepping control law does a good job of controlling the undamaged aircraft model. Now the control law will be evaluated in some simulation scenarios where (symmetric) structural damage occurs. These simulations will also be flown with a nonadaptive constrained backstepping control law with a correct onboard model of the F-16/MATV dynamics. By comparing the tracking performance of both control laws on the aircraft model with large, sudden changes in the aerodynamics, the effectiveness of the update laws with the B-spline neural networks can be illustrated more clearly. Furthermore, it will be interesting to see the robustness properties of the nonadaptive backstepping controller. The constrained adaptive backstepping control law will be designated as the CABS controller, while the constrained backstepping control law with correct onboard model is referred to as the CBS controller. Both control laws are identical except for the adaptation part.

One of the most important parameters of an aircraft when controlling the longitudinal motions is the total pitch moment coefficient $C_{m_T}$:

$$C_{m_T}=C_m(\alpha,\beta,\delta_e)+C_{Z_T}[x_{cg_r}-x_{cg}]+\delta C_{m_{\text{LEF}}}\left(1-\frac{\delta_{\text{LEF}}}{25}\right)+\frac{q\bar{c}}{2V_T}[C_{m_q}(\alpha)+\delta C_{m_{q_{\text{LEF}}}}(\alpha)]\left(1-\frac{\delta_{\text{LEF}}}{25}\right)+\delta C_m(\alpha)+\delta C_{m_{ds}}(\alpha,\delta_e)$$

An instant change in $C_{m_T}$ can, for instance, be caused by a change in the damping term $C_{m_q}$ or a longitudinal shift of the center of gravity $x_{cg}$.

In this paper several sudden changes in the $C_{m_q}$ damping term have been investigated. The consequences of a change in one of the pitch moment terms are best reflected in the angle of attack reference tracking task. The lateral commands for the sideslip angle and the roll rate are kept at zero during the simulations. The results of the simulation with instant changes in $C_{m_q}$ using the CBS controller can be found in Fig. 6. The aircraft starts again in trimmed condition at 5000 m altitude at an airspeed of 200 m/s, and the damage occurs after 3 s. Note that the CBS controller is surprisingly robust to the sudden change in damping term; only after taking $C_{m_q}$ equal to $-5C_{m_q}$ the controlled system can no longer be stabilized. The reason that the performance of the controller without adaptation is very robust to these large, sudden aerodynamic changes can be found in the size of the controller gains. The large gains give the derivative of the Lyapunov function for the closed-loop system more negativity. This is illustrated in Fig. 7, where the same maneuver is flown, only this time the gains of the CBS controller have been detuned to 0.5. The controller now stabilizes the aircraft in the nominal case, but fails to stabilize the aircraft if the value of the damping term is flipped.

Moving on to the plots of the maneuver with the aircraft using the CABS controller in Fig. 8, it can be seen that there is hardly any change at all in tracking performance with respect to the nominal case. Examining the rms values of the tracking errors for both controllers in Table 3 gives a better view of the results. From this table it can be clearly seen that the tracking performance of the adaptive controller is much better than that of the nonadaptive one after a failure occurs. The reason is that the weights of the B-spline networks have been adapted to the new aircraft dynamics. The weight update process for the pitch moment coefficients, during the simulation where the damping term becomes equal to $-5C_{m_q}$, is illustrated in Fig. 9. The local behavior of the B splines is clearly visible in these plots; only a few weights are updated at the same time. Note that, as discussed in Sec. III, the estimated values (weights) do not become equal to the real parameter values, because the adaptive backstepping theory only requires the parameter estimation error to become constant.

### C. Simulations with Actuator Failure

In this section sudden damage to the elevator actuator is considered. Actuator failures can be critical, because they change the structure of the aircraft model. The F-16/MATV model has redundant control surfaces, which means that the control law should still be able to stabilize the aircraft if the actuator of one of these surfaces fails.

In the last section a simple longitudinal maneuver was considered. In this section simulations of the same maneuver are performed for four different cases of elevator actuator failure: after 15 s of simulation the elevator moves to and becomes stuck at 20, 10, −10, or −20 deg deflection. After the failure occurs, the engine nozzle

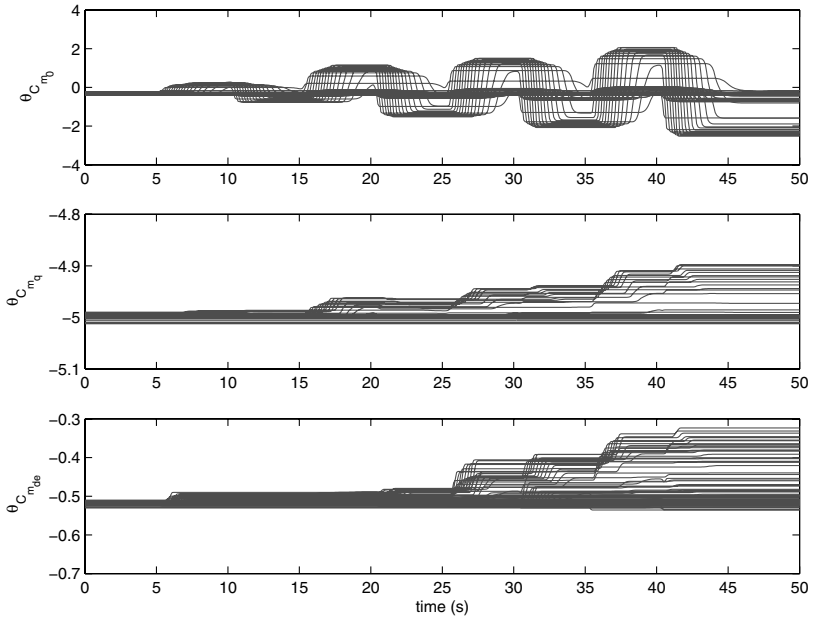


**Fig. 9 $C_{m_T}$ B-spline network weights—sudden change in damping term $C_{m_q}$.**

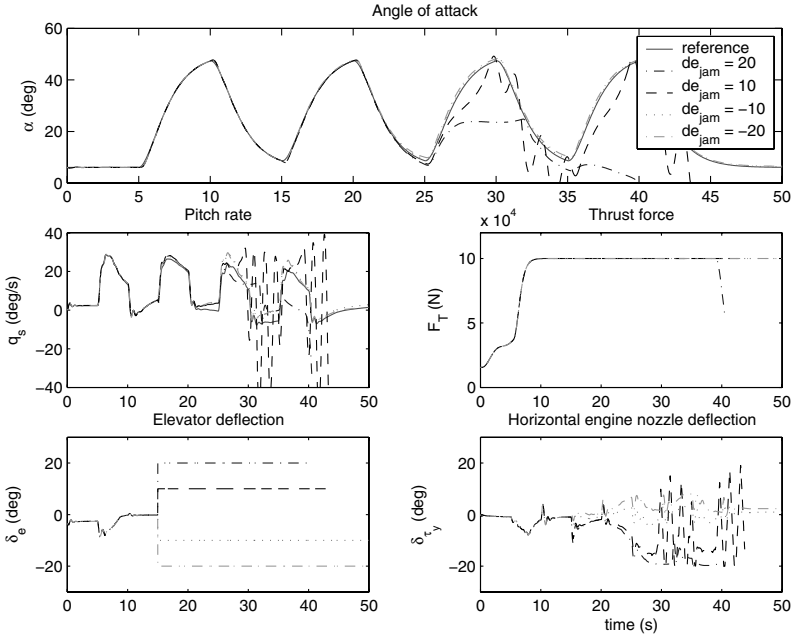


**Fig. 10 CBS controller—simulations with sudden failure of $\delta_e$ after 15 s.**

deflection $\delta_{\tau_y}$ remains the only available longitudinal control variable. The controller using the correct onboard model of the aerodynamics (CBS controller) will not notice that the control effectiveness of the elevator is now zero, while the update laws of the CABS controller should be able to estimate the effect of the failure.

The results of the simulations with both controllers are shown in Figs. 10 and 11. Note that the thrust $F_T$ has also been plotted to indicate the control effectiveness of the nozzle deflection. It can be seen that the CBS controller is able to control the aircraft for the cases where the elevator becomes stuck at $-10$ or $-20$ deg. For the other two cases the aircraft becomes unstable. These cases are indeed the most severe, because a positive angle of attack requires a negative elevator deflection. However, the engine nozzle deflection effectiveness is very high, because the thrust level is at its maximum, and thus should be able to compensate for the jammed elevator. But since this nonadaptive control law does not know that the elevator control surface is jammed, it still tries to generate part of the required control effort with it instead of canceling the negative effect. Eventually, this results in unstable behavior of the controlled system.

As expected, the CABS controller is still able to control the aircraft with the engine nozzle deflection, even for the most unfavorable elevator damage cases. The update laws have sensed the reduced control effectiveness and adapted the B-spline network weights accordingly.

The rms values of the tracking errors during the simulations can be found in Table 4 together with those for the same maneuver without damage (nominal case). The tracking performance of the CABS controller does decrease somewhat, but is still more than acceptable. The rms values of the tracking errors of the CBS controller are always larger then those of the adaptive controller, even for the cases when stability is not lost.

## VIII. Conclusions

In this paper a constrained adaptive backstepping flight control law is applied to a nonlinear fighter aircraft model with redundant control effectors. A major advantage of the adaptive backstepping method is that the stability and convergence properties of the control

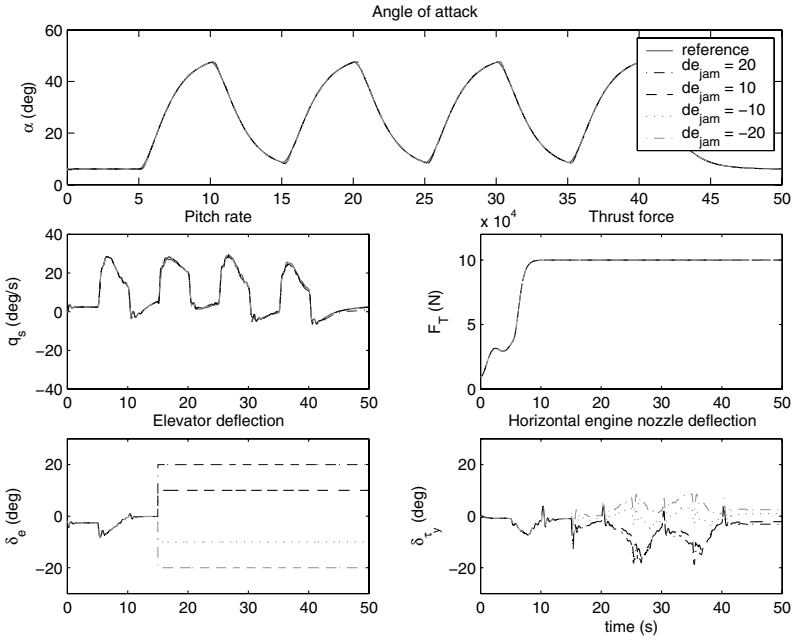


**Fig. 11 CABS controller—simulations with sudden failure of $\delta_e$ after 15 s.**

**Table 4 Robustness to sudden $\delta_e$ failure**

| $\delta_e$ position | CBS controller $(\alpha-\alpha^{\text{ref}})_{\text{rms}}$ | CBS controller $(q_s-q_s^{\text{ref}})_{\text{rms}}$ | CABS controller $(\alpha-\alpha^{\text{ref}})_{\text{rms}}$ | CABS controller $(q_s-q_s^{\text{ref}})_{\text{rms}}$ |
|---|---|---|---|---|
| Nominal case | 0.2498 | 0.3598 | 0.2447 | 0.2339 |
| $\delta_e=20$ | 12.6882 | 15.4910 | 0.2698 | 0.5764 |
| $\delta_e=10$ | 6.5048 | 21.4284 | 0.2657 | 0.4638 |
| $\delta_e=-10$ | 0.3911 | 2.5452 | 0.2596 | 0.4127 |
| $\delta_e=-20$ | 0.8409 | 5.9459 | 0.2579 | 0.5876 |

law can be proved using Lyapunov theory. Large changes in the aerodynamic forces and moments can be handled by the online update laws of the control algorithm, which make use of B-spline neural networks to approximate the nonlinear aerodynamic coefficients. The stability of the entire (weight) adaptation process in the presence of physical limits on the states and inputs is guaranteed. Computer simulations show that the control law has good tracking and decoupling capabilities even when large sudden actuator or symmetric structural failures occur.

In the paper, well-known techniques such as dead zones, leakage gains, and normalization of the regressor vectors to reduce computational load and to keep the values of the parameter estimates within reasonable values have not been considered. When a flight control law is actually implemented based on this control method, these techniques will of course have to be used.

## Appendix: Control Law Matrices

$$F_1=\begin{bmatrix} g_1 \\ -p_s\tan\beta+\frac{1}{mV_T\cos\beta}(-F_T\sin(\alpha+\delta_{\tau_y})+mg_3) \\ \frac{1}{mV_T}(-F_T\cos(\alpha+\delta_{\tau_y})\sin(\beta+\delta_{\tau_z})+mg_2)\end{bmatrix} +\frac{1}{m}\begin{bmatrix}\cos\alpha\cos\beta & \sin\beta & \sin\alpha\cos\beta \\ -\frac{\sin\alpha}{V_T\cos\beta} & 0 & \frac{\cos\alpha}{V_T\cos\beta} \\ -\frac{\cos\alpha\sin\beta}{V_T} & \frac{\cos\beta}{V_T} & -\frac{\sin\alpha\sin\beta}{V_T}\end{bmatrix}\begin{bmatrix}\hat{X}\\ \hat{Y}\\ \hat{Z}\end{bmatrix} \tag{A1}$$

$$B_1=\begin{bmatrix}\frac{1}{m}(\cos(\alpha+\delta_{\tau_y})\cos(\beta+\delta_{\tau_z})) & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & -1\end{bmatrix} \tag{A2}$$

$$F_2=\begin{bmatrix}r_s\\0\\-p_s\end{bmatrix}\left(q_s-p_s\tan\beta+\frac{(-\hat{X}'\sin\alpha+\hat{Z}'\cos\alpha-F_T\sin\alpha+mg_3)}{mV_T\cos\beta}\right) +T_{s/b}\begin{bmatrix}(c_1r+c_2p)q+c_3\hat{L}'+c_4\hat{N}'\\ c_5pr-c_6(p^2-r^2)+c_7(\hat{M}'+F_Tz_T)\\ (c_8p-c_2r)q+c_4\hat{L}'+c_9\hat{N}'\end{bmatrix} \tag{A3}$$

$$B_2=\begin{bmatrix} r_s\left(\frac{-\hat{X}_{\delta_e}\sin\alpha+\hat{Z}_{\delta_e}\cos\alpha}{mV_T\cos\beta}\right) & 0 & 0 & r_s\left(\frac{-F_T\cos\alpha}{mV_T\cos\beta}\right) & 0\\ 0 & 0 & 0 & 0 & 0\\ -p_s\left(\frac{-\hat{X}_{\delta_e}\sin\alpha+\hat{Z}_{\delta_e}\cos\alpha}{mV_T\cos\beta}\right) & 0 & 0 & -p_s\left(\frac{-F_T\cos\alpha}{mV_T\cos\beta}\right) & 0\end{bmatrix} +T_{s/b}\begin{bmatrix}c_3&0&c_4\\0&c_7&0\\c_4&0&c_9\end{bmatrix}\begin{bmatrix}\hat{L}_{\delta_e}&\hat{L}_{\delta_a}&\hat{L}_{\delta_r}&0&0\\ \hat{M}_{\delta_e}&0&0&-F_Tx_T&0\\ \hat{N}_{\delta_e}&\hat{N}_{\delta_a}&\hat{N}_{\delta_r}&0&-F_Tx_T\end{bmatrix} \tag{A4}$$

## References


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

[2] Shue, S. P., Sawan, M. E., and Rokhsaz, K., "Mixed H/H8 Method Suitable for Gain Scheduled Aircraft Control," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 4, 1997, pp. 699–706.

[3] Leith, D. J., and Leithead, W. E., "Gain-Scheduled Control: Relaxing Slow Variation Requirements by Velocity-Based Design," *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 6, 2000, pp. 988–1000.

[4] Richardson, T., Lowenberg, M., DiBernardo, M., and Charles, G., "Design of a Gain-Scheduled Flight Control System Using Bifurcation Analysis," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 2, 2006, pp. 444–453.

[5] da Costa, R. R., Chu, Q. P., and Mulder, J. A., "Reentry Flight Controller Design Using Nonlinear Dynamic Inversion," *Journal of Spacecraft and Rockets*, Vol. 40, No. 1, 2003, pp. 64–71.

[6] Littleboy, D. M., and Smith, P. R., "Using Bifurcation Methods to Aid Nonlinear Dynamic Inversion Control Law Design," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 4, 1998, pp. 632–638.

[7] Snells, S. A., Enns, D. F., and William, W. L., "Nonlinear Inversion Flight Control for a Supermaneuverable Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 4, 1992, pp. 976–984.

[8] Mulgund, S. S., and Stengel, R. F., "Aircraft Flight Control in Wind Shear Using Sequential Dynamic Inversion," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 5, 1995, pp. 1084–1091.

[9] Slotine, J. J. E., and Li, W., *Applied Nonlinear Control*, Prentice–Hall, Upper Saddle River, NJ, 1991.

[10] Isidori, A., *Nonlinear Control Systems*, 3rd ed., Springer, New York, 1995.

[11] Zhou, K., and Doyle, J., *Essentials of Robust Control*, Prentice–Hall, Upper Saddle River, NJ, 1997.

[12] van Oort, E. R., Chu, Q. P., and Mulder, J. A., "Robust Model Predictive Control of a Feedback Linearized F-16/MATV Aircraft Model," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA, Reston, VA, Aug. 2006.

[13] Bijnens, B., Chu, Q. P., Voorsluijs, G., and Mulder, J. A., "Adaptive Feedback Linearization Flight Control for a Helicopter UAV," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA, Reston, VA, Aug. 2005.

[14] Kannelakopoulos, I., Kokotović, P. V., and Morse, A. S., "Systematic Design of Adaptive Controllers for Feedback Linearizable Systems," *IEEE Transactions on Automatic Control*, Vol. 36, No. 11, Nov. 1991, pp. 1241–1253.

[15] Krstić, M., Kanellakopoulos, I., and Kokotović, P. V., "Adaptive Nonlinear Control Without Over-Parametrization," *Systems and Control Letters*, Vol. 19, Sept. 1992, pp. 177–185.

[16] Singh, S. N., and Steinberg, M., "Adaptive Control of Feedback Linearizable Nonlinear Systems with Application to Flight Control," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 4, July–Aug. 1996, pp. 871–877.

[17] Steinberg, M. L., and Page, A. B., "Nonlinear Adaptive Flight Control with Genetic Algorithm Design Optimization," *International Journal of Robust and Nonlinear Control*, Vol. 9, No. 14, 1999, pp. 1097–1115.

[18] Singh, S. N., Chandler, P., Schumacher, C., Banda, S., and Pachter, M., "Nonlinear Adaptive Close Formation Control of Unmanned Aerial Vehicles," *Dynamics and Control*, Vol. 10, No. 2, 2000, pp. 179–194.

[19] Sharma, M., and Ward, D. G., "Flight-Path Angle Control via Neuro-Adaptive Backstepping," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA, Reston, VA, Aug. 2002.

[20] Ju, H. S., Tsai, C. C., and Liu, S. W., "Design of Longitudinal Axis Full Envelope Control Law by Adaptive Backstepping," *Proceedings of the 2004 IEEE*, IEEE, Piscataway, NJ, March 2004.

[21] Kim, S. H., Kim, Y. S., and Song, C., "A Robust Adaptive Nonlinear Control Approach to Missile Autopilot Design," *Control Engineering Practice*, Vol. 12, No. 2, 2004, pp. 149–154.

[22] Farrell, J., Sharma, M., and Polycarpou, M., "On-Line Approximation Based Aircraft Longitudinal Control," *Proceedings of the 2003 American Control Conference*, American Automatic Control Council, Evanston, IL, 2003.

[23] Farrell, J., Polycarpou, M., and Sharma, M., "Adaptive Backstepping with Magnitude, Rate, and Bandwidth Constraints: Aircraft Longitude Control," *Proceedings of the 2003 American Control Conference*, American Automatic Control Council, Evanston, IL, 2003, pp. 3898–3903.

[24] Polycarpou, M., Farrell, J., and Sharma, M., "On-Line Approximation Control of Uncertain Nonlinear Systems: Issues with Control Input Saturation," *Proceedings of the 2003 American Control Conference*, American Automatic Control Council, Evanston, IL, 2003.

[25] Farrell, J., Sharma, M., and Polycarpou, M., "Backstepping Based Flight Control with Adaptive Function Approximation," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 6, 2005, pp. 1089–1102.
[26] Lee, T., and Kim, Y., "Nonlinear Adaptive Flight Control Using Backstepping and Neural Networks Controller," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 4, July–Aug. 2001, pp. 675–682.
[27] Cheng, K. W. E., Wang, H., and Sutanto, D., "Adaptive B-Spline Network Control for Three-Phase PWM AC-DC Voltage Source Converter," *Proceedings of the IEEE 1999 International Conference on Power Electronics and Drive Systems*, IEEE, Piscataway, NJ, July 1999.
[28] Ward, D. G., Sharma, M., Richards, N. D., Luca, J. D., and Mears, M., "Intelligent Control of Un-Manned Air Vehicles: Program Summary and Representative Results," *Proceedings of the 2nd AIAA Unmanned Unlimited Systems, Technologies and Operations Aerospace, Land and Sea*, AIAA, Reston, VA, 2003.
[29] Härkegård, O., "Flight Control Design Using Backstepping," M.S. Thesis, Linköping University, Linköping, Sweden, 2001.
[30] Shin, D. H., and Kim, Y., "Reconfigurable Flight Control System Design Using Adaptive Neural Networks," *IEEE Transactions on Control Systems Technology*, Vol. 12, No. 1, Jan. 2004, pp. 543–550.
[31] O'Rourke, M. J., Ralston, J. N., Bell, J. W., and Lash, S. F., "PC-Based Simulation of the F16/MATV," AIAA Paper 97-3728, 1997.
[32] Lewis, B. L., and Stevens, F. L., *Aircraft Control and Simulation*, Wiley, New York, 1992.
[33] Forssell, L. S., and Nilsson, U., *ADMIRE, The Aero-Data Model in a Research Environment*, FOI Swedish Defence Research Agency, 2004.
[34] Nguyen, L. T., Ogburn, M. E., Gilbert, W. P., Kibler, K. S., Brown, P. W., and Deal, P., "NASA Technical Paper 1538—Simulator Study of Stall/Post-Stall Characteristics of a Fighter Airplane with relaxed Longitudinal Static Stability," NASA, Tech. Rept., Dec. 1979.
[35] Reigelsperger, W. C., Hammett, K. D., and Banda, S. S., "Robust Control Law Design for Lateral/Directional Modes of an F-16/MATV Using $\mu$-Synthesis and Dynamic Inversion," *International Journal of Robust and Nonlinear Control*, Vol. 7, No. 8, 1997, pp. 777–795.
[36] Kanellakopoulos, I., "Adaptive Control of Nonlinear Systems," Ph.D. Thesis, University of Illinois, Urbana, IL, 1991.
[37] LaSalle, J. P., "Stability Theory for Ordinary Differential Equations," *Journal of Differential Equations*, Vol. 4, No. 11, 1968, pp. 57–65.
[38] Yoshizawa, T., "Stability Theory by Lyapunov' Second Method," The Mathematical Society of Japan, Tokyo, 1966.
[39] Härkegård, O., "Backstepping and Control Allocation with Applications to Flight Control," Ph.D. Thesis, Linköping University, Linköping, Sweden, 2003.
[40] Bordignon, K. A., "Constrained Control Allocation for Systems with Redundant Control Effectors, Ph.D. Thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, Dec. 1996.
[41] Durham, W. C., "Constrained Control Allocation," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 4, July–Aug. 1993, pp. 717–725.
[42] Ioannou, P. A., and Sun, J., *Stable and Robust Adaptive Control*, Prentice–Hall, Upper Saddle River, NJ, 1995.
[43] Russell, R. S., "Nonlinear F-16 Simulation Using Simulink and Matlab," University of Minnesota, Tech. Rept., Ver. 1.0, 2003.
[44] Haykin, S., *Neural Networks A Comprehensive Foundation*, Prentice–Hall, Upper Saddle River, NJ, 1994.
[45] Brown, M., and Harris, C., *Neurofuzzy Adaptive Modelling and Control*, Prentice–Hall, Upper Saddle River, NJ, 1994.
[46] 20-sim Control Toolbox, 20-sim help files, Controllab Products, B. V., www.20sim.com, 2005.
[47] Shene, C. K., *Introduction to Computing with Geometry Notes*, Department of Computer Science, Michigan Technological University, www.cs.tu.edu/ shene/COURSES/cs3621/NOTES, 2003.
[48] de Weerdt, E., Chu, Q. P., and Mulder, J. A., "Neural Network Aerodynamic Model Identification for Aerospace Reconfiguration," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA, Reston, VA, Aug. 2005.