JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 29, No. 3, May–June 2006

Fault-Tolerant Structured Adaptive Model Inversion Control

Monish D. Tandale* and John Valasek†
Texas A&M University, College Station, Texas 77843-3141

An adaptive dynamic inversion control formulation is presented that takes advantage of the inherent dynamic structure of the state-space description of a large class of systems. The formulations impose the exact kinematic differential equations, thereby restricting the adaptation process that compensates for model errors to the acceleration level. The utility of this formulation is demonstrated for the problem of fault tolerance to actuator failures on redundantly actuated systems. The approach incorporates an actuator failure model in the controller formulation, so that actuator failure can be identified as a change in the parameters of the failure model. Tracking of reference trajectories is imposed, and initial error conditions and structured parametric uncertainties are incorporated explicitly in both the plant parameters and the control influence matrix. A numerical example consisting of a nonlinear model of an F-16 type aircraft with thrust vectoring is presented. Simulation results show that the fault-tolerant adaptive controller is capable of simultaneously handling parametric uncertainties, large initial condition errors, and actuator failures while maintaining adequate tracking performance.

I. Introduction

In recent years, there has been much interest in the development of reconfigurable control systems that can accommodate actuator failures without compromising mission integrity. There has been substantial progress in the development of real-time failure detection and isolation algorithms, system identification after failure, and control reconfiguration techniques in aerospace applications. In Ref. 1, a survey of various reconfigurable flight control methodologies is presented and it is shown that most traditional reconfiguration flight control approaches rely on failure detection and isolation. The complexity of such a system with this feature grows with the increase in the number of failures, and there tends to be a significant possibility of false alarms.[1,2] A different approach to reconfigurable flight control is based on adaptive control theory, in which the adaptive control structure implicitly reconfigures the control law using adaptive estimates of the altered dynamics after failure.[3] In Ref. 3, an adaptive control scheme is presented that uses a linear approximation of the plant model to compute the control, and a neural network based adaptive control law for flight reconfiguration has been developed and successfully flight tested.[4–6] A robust fault-tolerant controller has also been developed to reject state-dependent disturbances.[7]

The approach presented in this paper uses a structured nonlinear adaptive dynamic inversion control methodology. Instead of using an explicit failure detection and isolation algorithm, this methodology is based on the adaptive control theory where the controller is constantly updating itself. This methodology is applicable to a general class of nonlinear systems that are affine in the control with uncertain parameters appearing linearly. Fault-tolerance capability is introduced by incorporating a failure model in the controller so that a failure can be identified and compensated for by a change in the parameters of the failure model.

First, model reference adaptive control, structured model reference adaptive control, and structured adaptive model inversion control are introduced. Then, a failure model that can handle a wide range of failures is developed, followed by the mathematical development and stability analysis of the fault-tolerant structured adaptive model inversion (SAMI) controller. A nonlinear six-degree-of-freedom simulation of an F-16 type aircraft with thrust vectoring is developed. Finally, a numerical example is presented that demonstrates fault-tolerant performance of the controller in the presence of actuator failure, using the simulation model developed.

All dynamic systems that exist in practice have uncertain parameters, which may be constant with respect to time, or may vary with respect to the changing environment in which the dynamic system is functioning. Because of this variation, performance of the system may degrade if a fixed parameter controller is used. However, performance can be ensured if the controller parameters can be adjusted to adapt to the changes in the plant parameters. The basic premise of adaptive control is that the controller parameters are variable, and there exists a mechanism for updating the parameters of the system real time, based on the signals of the system. In the broad class of adaptive control there are several variants, some of which are described here. In model reference adaptive control (MRAC),[8,9] a reference model is specified that in turn specifies the desired dynamics that the plant is supposed to follow. A feedback controller exists for the plant as usual, but the parameters of this controller are varying. The adaptation mechanism for the controller parameters is driven by the error between the output of the actual plant and the reference model (Fig. 1). The adaptation mechanism is usually synthesized by the use of Lyapunov stability theory (see Ref. 10). The basic theory of MRAC is well developed due to significant contributions by Narendra and Annaswamy,[11] Sastry and Bodson,[12] and Iannou and Sun.[13]

When advantage is taken of the inherent kinematic and dynamic structure of a system, complexity in the mathematical formulation of the controller can be reduced, resulting in a more efficient control algorithm. The dynamics of most mechanical systems can frequently be represented in the form of a second-order differential equation, which can be separated into a kinematic part and a dynamic part. In some other cases, the system is in first-order form, but can naturally be partitioned into a cascade form with kinematic and dynamic equations. The kinematic equations are accurately known and do not contain uncertain system parameters such as mass, moment of inertia, etc. The uncertainties exist only in the momentum level dynamic equations, and for this reason it makes sense to restrict adaptation to the momentum level equations only.

As a simple example, the second-order differential equation $F = ma$ can be written as $\dot{x} = v$ and $\dot{v} = F/m$. The uncertain system parameter m exists only in the dynamic part, and so adaptation can be restricted to the second equation only. When the adaptation is restricted to a subspace of the entire state space, the adaptation

Presented as Paper 2002-4437 at the AIAA Guidance, Navigation, and Control Conference, Monterey, CA, 5–8 August 2002; received 22 December 2004; accepted for publication 11 January 2005.

*Graduate Research Assistant, Flight Simulation Laboratory, Aerospace Engineering Department; monish@neo.tamu.edu. Student Member AIAA.
†Associate Professor & Director, Flight Simulation Laboratory, Aerospace Engineering Department; valasek@aero.tamu.edu. Associate Fellow AIAA.

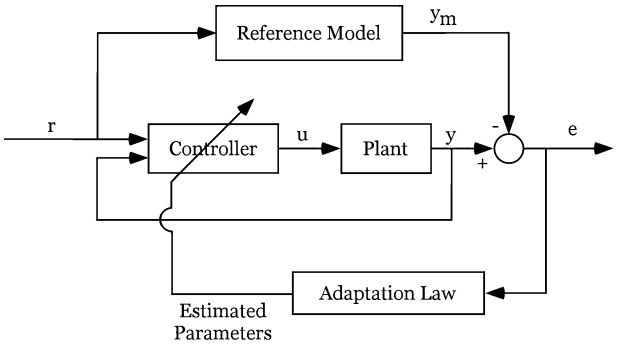

Fig. 1 MRAC system.

becomes more efficient. Structured MRAC (SMRAC)[14] has been shown to be effective in the presence of large system model errors,[15] bounded disturbances,[16] and actuator saturation limits.[17]

SAMI[18] is based on the concepts of feedback linearization,[9] dynamic inversion, and SMRAC. SAMI uses dynamic inversion to solve for the control. In dynamic inversion, the desired trajectories that the system is supposed to follow are specified beforehand. The actual plant dynamics are constructed using a plant mathematical model, so that the calculated controls simultaneously cancel out the plant dynamics and augment the desired dynamics.

For dynamic inversion to work efficiently, the mathematical model of the plant should be accurate so that the actual dynamics of the plant are canceled exactly. However, in practice, an exact cancellation is usually not achieved because the system parameters are not always known accurately. In SAMI, an adaptive controller is wrapped around the dynamic inversion to account for the uncertainties in the system parameters. This controller is designed to drive the error between the output of the actual plant and that of a model reference to zero, as in MRAC. The adaptation included in this framework can be limited to only the momentum level states, so that the structured flavor is retained in the controller. SAMI has been shown to be effective for tracking spacecraft[19] and for aggressive aircraft maneuvers.[20]

II. Actuator Failure

In high-performance dynamic systems, the total number of actuators used may be greater than the number of states to be controlled or tracked. This control redundancy generally exists to achieve optimality with respect to control effort. Consider a case where the number of actuators is greater than the number of states to be tracked. Mathematically, there are more variables than the equations to be satisfied, therefore, an infinite number of solutions exist. All of these solutions have varying control energy requirements, and so the solution with minimum control energy can be selected, and optimality can be achieved.

In this case of redundant actuation, it is still possible to track closely the desired states, even if some of the actuators fail, as long as the number of active actuators is greater than or equal to the number of states to be tracked. Thus, if it is possible to reconfigure the control after failure, the stability and performance of the system can in theory be maintained. Many actuator failure schemes employ some form of failure detection algorithm to detect the failure. A new control effectiveness matrix is estimated, and the controller is redesigned by recalculating the control gains.[21]

This approach depends strongly on the efficacy of the failure detection algorithm. The algorithm may fail to detect a failure, or may give false warning when there is no actuation failure. In contrast, the SAMI controller is constantly updating its parameters so that it does not need to detect the failure specifically. The failure is implicitly identified as a change in the parameters of the failure model, and the adaptation mechanism adapts to this change. Therefore, SAMI is a good candidate to address the problem of actuator failure.

III. Mathematical Modeling of Actuator Failure

The actuator failures commonly encountered in aircraft include the following : 1) lock in place, 2) hard-over failure, 3) partial loss of effectiveness, and 4) float. In lock-in-place type failures, the control surface freezes or remains fixed at a position that may or may not be the neutral position or zero. In this case, the remaining operating actuators must not only compensate for the lack of the desired control effort of the failed actuator, but must also cancel the undesired control effect produced if the actuator freezes at any position other than zero. In the hard-over failure, the actuator freezes at either one of the position limits. Partial loss of effectiveness may occur due to physical damage of the control effector. Float-type failure occurs when the actuator floats in the airflow and does not produce any control moment.

The preceding control failures can be modeled for each control by the following mathematical model:

$$u_{\text{app}} = du_{\text{cal}} + e \tag{1}$$

where d and e are constants for a particular control configuration, but may change and settle to other constant values if a control failure occurs. Here, u_{cal} is the control demanded by the controller and u_{app} is the control that can actually be applied because of the saturation limit. Because $u_{\text{app}} = u_{\text{cal}}$ in the absence of failure, d is initialized to one, and e is initialized as zero. An actuator freeze can be modeled by d going to zero and e going to the constant value at which the control surface has frozen. Thus, this model accounts for lock in place and hard-over failures. This mathematical model can also model loss of effectiveness. Consider a case in which a control surface is damaged/lost such that only one-half of it remains. If it is assumed that the control effectiveness is reduced by one-half, d should go to 0.5, and e should remain at zero. Float-type failures can also be accounted for, with d going to zero and e remaining at zero.

For the case in which the control is a vector, the failure model can be assembled as follows:

$$\begin{bmatrix} u_{a1} \\ u_{a2} \\ u_{a3} \end{bmatrix} = \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{bmatrix} \begin{bmatrix} u_{c1} \\ u_{c2} \\ u_{c3} \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \tag{2}$$

Thus,

$$\boldsymbol{u}_{\text{app}} = D\boldsymbol{u}_{\text{cal}} + \boldsymbol{E} \tag{3}$$

By augmentation of the SAMI formulation with this control failure model, a framework is created that accommodates actuator failures and damage as changes in the parameters of the system.

IV. Mathematical Formulation of Fault-Tolerant SAMI Controller

A. Definition of Plant and Reference Model

Consider a nonlinear dynamic system that is affine in the control and can be split into a structured form consisting of an exactly known kinematic differential equation and a momentum level equation with uncertain parameters

$$\dot{\boldsymbol{\sigma}} = J(\boldsymbol{\sigma})\boldsymbol{\omega} \tag{4}$$

$$\dot{\boldsymbol{\omega}} = \boldsymbol{A}(\boldsymbol{\sigma}, \boldsymbol{\omega}) + B(\boldsymbol{\sigma}, \boldsymbol{\omega})\boldsymbol{u}_{\text{app}} \tag{5}$$

where $\boldsymbol{\sigma} \in \mathbb{R}^n$ is the vector of position level coordinates, $\boldsymbol{\omega} \in \mathbb{R}^n$ is the vector of velocity level coordinates, $J \in \mathbb{R}^{n \times n}$ is the nonlinear transformation relating $\dot{\boldsymbol{\sigma}}$ and $\boldsymbol{\omega}$, $\boldsymbol{A}(\boldsymbol{\sigma},\boldsymbol{\omega})$ quantifies the unforced behavior of the system, and $B(\boldsymbol{\sigma},\boldsymbol{\omega})$ is the control effectiveness matrix.

The dynamics of the system are assumed to be modeled accurately, and only structured parametric uncertainties exist in the model. Note that the most convenient velocity coordinates may not always be the derivatives of the position. When the case of an aircraft is considered, the position coordinates are the translational displacements along the Earth-fixed inertial axis. The velocity level equation is written in the body axis because the moment of inertia remains constant in the body axis at every instant, but the moment of inertia in the inertial axis changes, depending on the orientation

of the body. Thus, the velocity coordinates are the body-axis velocities, and the velocity coordinate and the position coordinate are related via a series of coordinate axis rotations. Thus, the term J in Eq. (4) is a result of multiplication of a series of coordinate rotation matrices.

Substituting for $\boldsymbol{u}_{\text{app}}$ from Eq. (3) into Eq. (5), we get

$$\dot{\omega} = \boldsymbol{A}(\boldsymbol{\sigma}, \boldsymbol{\omega}) + B(\boldsymbol{\sigma}, \boldsymbol{\omega}) D\boldsymbol{u}_{\text{cal}} + B(\boldsymbol{\sigma}, \boldsymbol{\omega})\boldsymbol{E} \tag{6}$$

The control objective is to track the reference trajectory in terms of $\boldsymbol{\sigma}_r$ and $\boldsymbol{\omega}_r$ where $\boldsymbol{\sigma}_r$ and $\boldsymbol{\omega}_r$ are related by $\dot{\boldsymbol{\sigma}}_r = J_r\ (\boldsymbol{\sigma}_r)\boldsymbol{\omega}_r$.

For the case of an unmanned air vehicle, predefined reference trajectories $(\boldsymbol{\sigma}_r, \boldsymbol{\omega}_r)$ can be provided to the controller. If the vehicle is piloted, these reference trajectories can be generated in real time as the output of a reference model. Consider a reference model having a structure similar to that of the nonlinear plant

$$\dot{\boldsymbol{\sigma}}_r = J_r(\boldsymbol{\sigma}_r)\boldsymbol{\omega}_r \tag{7}$$

$$\dot{\boldsymbol{\omega}}_r = \boldsymbol{A}(\boldsymbol{\sigma}_r, \boldsymbol{\omega}_r) + B(\boldsymbol{\sigma}_r, \boldsymbol{\omega}_r)\boldsymbol{u}_r \tag{8}$$

Thus, the reference model is a representation of the dynamics desired by the pilot flying the aircraft, and u_r are the pilot's inputs.

B. Calculating Control by Dynamic Inversion of Dynamic Level Equation

Let the error in the position and the velocity level states be $\boldsymbol{s}$ and $\boldsymbol{x}$, respectively,

$$\boldsymbol{s} = \boldsymbol{\sigma} - \boldsymbol{\sigma}_r \tag{9}$$

$$\boldsymbol{x} = \boldsymbol{\omega} - \boldsymbol{\omega}_r \tag{10}$$

The control appears only in the velocity level equation and, hence, the corresponding velocity level error. The control affects the position through the integration of Eq. 5 via the coupling seen in Eq. 4,

$$\dot{\boldsymbol{x}} = \dot{\boldsymbol{\omega}} - \dot{\boldsymbol{\omega}}_r \tag{11}$$

$$\dot{\boldsymbol{x}} = \boldsymbol{A} + BD\boldsymbol{u}_{\text{cal}} + B\boldsymbol{E} - \dot{\boldsymbol{\omega}}_r \tag{12}$$

We want the error between the reference and the plant to go to zero. Therefore, we prescribe the following dynamics for $\boldsymbol{x}$:

$$\dot{\boldsymbol{x}} = A_h\boldsymbol{x} + \boldsymbol{\phi} \tag{13}$$

where A_h is a Hurwitz matrix, that is, all eigenvalues lie in the open left-half plane so that the velocity error dynamics are stable and the velocity error goes to zero asymptotically. A_h can be selected arbitrarily, but a proper choice can specify how fast the velocity error goes to zero. Here, $\boldsymbol{\phi}$ is a forcing function on the velocity error dynamics, which helps in achieving the tracking objective. This is discussed in detail later.

When $A_h\boldsymbol{x} + \boldsymbol{\phi}$ is added and subtracted on the right-hand side of Eq. (12),

$$\dot{\boldsymbol{x}} = A_h\boldsymbol{x} + \boldsymbol{\phi} + \boldsymbol{A} + BD\boldsymbol{u}_{\text{cal}} + B\boldsymbol{E} - (\dot{\boldsymbol{\omega}}_r + A_h\boldsymbol{x} + \boldsymbol{\phi}) \tag{14}$$

We assume that the state variables are measurable, which is a reasonable assumption for the aircraft model. Because the quantity in the brackets is known, let

$$\boldsymbol{\psi} \triangleq \dot{\boldsymbol{\omega}}_r + A_h\boldsymbol{x} + \boldsymbol{\phi} \tag{15}$$

Therefore,

$$\dot{\boldsymbol{x}} = A_h\boldsymbol{x} + \boldsymbol{\phi} + \boldsymbol{A} + BD\boldsymbol{u}_{\text{cal}} + B\boldsymbol{E} - \boldsymbol{\psi} \tag{16}$$

We calculate the control so that the velocity error x has the desired dynamics as shown in Eq. (13). When dynamic inversion is used to solve for the control,

$$\boldsymbol{u}_{\text{cal}} = (BD)^{-1}(\boldsymbol{\psi} - \boldsymbol{A} - B\boldsymbol{E}) \tag{17}$$

If m is equal to n, that is, the number of controls is equal to the number of velocity level states, (BD) will be an $n \times n$ matrix of full rank, and its inverse can be computed. If the system is underactuated (m less than n), (BD) will be rank deficient, and its inverse cannot be computed. If the system is overactuated (such that actuation is redundant) then m is greater than n, (BD) will be a wide matrix, and the minimum norm solution can be used to calculate $\boldsymbol{u}_{\text{cal}}$. If there is actuation failure, (BD) will lose rank because the diagonal element corresponding to the failed control goes to zero. Rank (BD) is equal to the number of active controls; hence, the number of active controls after failure should be greater than or at least equal to n. For redundant actuation,

$$\boldsymbol{u}_{\text{cal}} = \text{pinv}(BD)(\boldsymbol{\psi} - \boldsymbol{A} - B\boldsymbol{E}) \tag{18}$$

where

$$\text{pinv}(BD) = (BD)^T[BD(BD)^T]^{-1} \tag{19}$$

C. Definition of Adaptive Learning Parameters

Equation (18) requires the exact values of $\boldsymbol{A}$ and B so that the plant dynamics are canceled exactly. However, the system parameters $\boldsymbol{A}$ and B are not assumed to be known accurately, and so best guesses for $\boldsymbol{A}$ and B ($\boldsymbol{A}_{\text{est}}$ and B_{est}) will be used. Let

$$\boldsymbol{u}_{\text{cal}} = \text{pinv}(B_{\text{est}}D)(\boldsymbol{\psi} - C_a\boldsymbol{A}_{\text{est}} - B_{\text{est}}\boldsymbol{E}) \tag{20}$$

where C_a is the adaptive learning matrix that will be updated online to ensure stability and performance of the system. There exists a matrix C_a^*, which is the ideal value for the learning parameter C_a, such that

$$C_a^*\boldsymbol{A}_{\text{est}} = \boldsymbol{A} \tag{21}$$

The D matrix that had been introduced to incorporate the control failure can also account for the parametric uncertainty in B, just as the uncertainty in $\boldsymbol{A}$ is accommodated using C_a. Similarly,

$$B_{\text{est}}D^* = BD \tag{22}$$

$$B_{\text{est}}E^* = BE \tag{23}$$

From Eq. (20)

$$\boldsymbol{\psi} = C_a\boldsymbol{A}_{\text{est}} + B_{\text{est}}D\boldsymbol{u}_{\text{cal}} + B_{\text{est}}\boldsymbol{E} \tag{24}$$

Substituting Eqs. (21–23) in Eq. (16) results in

$$\dot{\boldsymbol{x}} = A_h\boldsymbol{x} + \boldsymbol{\phi} + C_a^*\boldsymbol{A}_{\text{est}} + B_{\text{est}}D^*\boldsymbol{u}_{\text{cal}} + B_{\text{est}}\boldsymbol{E}^* - C_a\boldsymbol{A}_{\text{est}} - B_{\text{est}}D\boldsymbol{u}_{\text{cal}} - B_{\text{est}}\boldsymbol{E} \tag{25}$$

When

$$\tilde{C}_a \triangleq C_a^* - C_a \tag{26}$$

$$\tilde{D} \triangleq D^* - D \tag{27}$$

$$\tilde{\boldsymbol{E}} \triangleq \boldsymbol{E}^* - \boldsymbol{E} \tag{28}$$

are defined, Eq. (25) now becomes

$$\dot{\boldsymbol{x}} = A_h\boldsymbol{x} + \boldsymbol{\phi} + \tilde{C}_a\boldsymbol{A}_{\text{est}} + B_{\text{est}}\tilde{D}u_{\text{cal}} + B_{\text{est}}\tilde{\boldsymbol{E}} \tag{29}$$

D. Incorporating Position into Tracking Error

The control is derived from the dynamic part, and it is assumed that the adaptive mechanism provides perfect velocity tracking. However, this does not ensure that the position reference will be tracked correctly. The initial errors in position or errors in the velocity during the transient stage, before perfect velocity tracking is achieved, will cause the position to stray from the reference, and no attempts at correcting this error will be made unless this deviation is identified as an error. Therefore, the tracking error should have a

contribution from the position level state also. Let the total tracking error be defined as

$$\boldsymbol{y} \triangleq \dot{\boldsymbol{s}} + \lambda \boldsymbol{s} \tag{30}$$

where $\lambda \in \boldsymbol{R}^{n\times n}$ is a positive definite matrix. As $t \to \infty$, if $\boldsymbol{y}$ is driven to zero, it is ensured that $s \to 0$ and $\dot{s} \to 0$.

When Eq. (9) is differentiated with respect to time,

$$\dot{\boldsymbol{s}} = \dot{\boldsymbol{\sigma}} - \dot{\boldsymbol{\sigma}}_r \tag{31}$$

$$\dot{\boldsymbol{s}} = J\boldsymbol{\omega} - J_r\boldsymbol{\omega}_r \tag{32}$$

$$\boldsymbol{y} = J\boldsymbol{\omega} - J_r\boldsymbol{\omega}_r + \lambda \boldsymbol{s} \tag{33}$$

When $J\boldsymbol{\omega}_r$ is added and subtracted from the right-hand side,

$$\boldsymbol{y} = J(\boldsymbol{\omega} - \boldsymbol{\omega}_r) + J\boldsymbol{\omega}_r - J_r\boldsymbol{\omega}_r + \lambda \boldsymbol{s} \tag{34}$$

By definition, $\boldsymbol{x} = \boldsymbol{\omega} - \boldsymbol{\omega}_r$

$$\boldsymbol{y} = J\boldsymbol{x} + J\boldsymbol{\omega}_r - J_r\boldsymbol{\omega}_r + \lambda \boldsymbol{s} \tag{35}$$

When Eq. (35) is differentiated to obtain the derivative of the tracking error,

$$\dot{\boldsymbol{y}} = J\dot{x} + \dot{J}x + (\dot{J} - \dot{J}_r)\boldsymbol{\omega}_r + (J - J_r)\dot{\boldsymbol{\omega}}_r + \lambda \dot{\boldsymbol{s}} \tag{36}$$

When $\dot{\boldsymbol{x}}$ is substituted for from Eq. (29),

$$\dot{\boldsymbol{y}} = J(A_h\boldsymbol{x} + \boldsymbol{\phi} + \tilde{C}_a\boldsymbol{A}_{\text{est}} + B_{\text{est}}\tilde{D}\boldsymbol{u}_{\text{cal}} + B_{\text{est}}\tilde{\boldsymbol{E}}) + \dot{J}\boldsymbol{x} + (\dot{J} - \dot{J}_r)\boldsymbol{\omega}_r + (J - J_r)\dot{\boldsymbol{\omega}}_r + \lambda \dot{\boldsymbol{s}} \tag{37}$$

The quantities $\tilde{C}_a$, $\tilde{D}$, and $\tilde{E}$ are not known, whereas all other quantities are known. For the tracking error to stabilize, the following dynamics are prescribed to the known quantities. The uncertain quantities will be addressed with a Lyapunov analysis shown later. We want the tracking error $\boldsymbol{y}$ to go to zero, and so we select the forcing function $\boldsymbol{\phi}$ so that the tracking error will have the following dynamics:

$$\dot{\boldsymbol{y}} = A_h\boldsymbol{y} \tag{38}$$

Because A_h has all eigenvalues in the left-half plane, the tracking error dynamics will be stable and the tracking error will converge to zero asymptotically. Thus, when the known terms on the right-hand side of Eq. (37) are equated to $A_h\boldsymbol{y}$,

$$JA_h\boldsymbol{x} + J\boldsymbol{\phi} + \dot{J}\boldsymbol{x} + (\dot{J} - \dot{J}_r)\boldsymbol{\omega}_r + (J - J_r)\dot{\boldsymbol{\omega}}_r + \lambda \dot{\boldsymbol{s}} = A_h\boldsymbol{y} \tag{39}$$

When the forcing function $\boldsymbol{\phi}$ is solved for,

$$\boldsymbol{\phi} = J^{-1}(A_h\boldsymbol{y} - \lambda \dot{\boldsymbol{s}} - \dot{J}\boldsymbol{\omega} + J_r\dot{\boldsymbol{\omega}}_r + \dot{J}_r\boldsymbol{\omega}_r) - \dot{\boldsymbol{\omega}}_r - A_h\boldsymbol{x} \tag{40}$$

Finally, by substitution of the value of $\boldsymbol{\phi}$ in Eq. (37),

$$\dot{\boldsymbol{y}} = A_h\boldsymbol{y} + J(\tilde{C}_a\boldsymbol{A}_{\text{est}} + B_{\text{est}}\tilde{D}\boldsymbol{u}_{\text{cal}} + B_{\text{est}}\tilde{\boldsymbol{E}}) \tag{41}$$

E. Lyapunov Analysis and Update Laws for Adaptive Learning Parameters

Consider the candidate Lyapunov function V. The function V is a measure of the total error in the system, where $\boldsymbol{y}$ is the total tracking error, $\tilde{C}_a$ represents the error in the unforced dynamic behavior, $\tilde{D}$ represents the error in the control effectiveness and the failure model, and $\tilde{\boldsymbol{E}}$ is the error in the failure model, If P, W_1, W_2 are positive definite matrices,

$$V = \boldsymbol{y}^T P\boldsymbol{y} + Tr\left(\tilde{C}_a^T W_1\tilde{C}_a + \tilde{D}^T W_2\tilde{D}\right) + \tilde{\boldsymbol{E}}^T W_3\tilde{\boldsymbol{E}} \tag{42}$$

When the derivative of the Lyapunov function is taken,

$$\dot{V} = \boldsymbol{y}^T P\dot{\boldsymbol{y}} + \dot{\boldsymbol{y}}^T P\boldsymbol{y} + 2Tr\left(\tilde{C}_a^T W_1\dot{\tilde{C}}_a + \tilde{D}^T W_2\dot{\tilde{D}}\right) + 2\tilde{\boldsymbol{E}}^T W_3\dot{\tilde{\boldsymbol{E}}} \tag{43}$$

$$\dot{V} = \boldsymbol{y}^T PA_h\boldsymbol{y} + \boldsymbol{y}^T A_h^T P\boldsymbol{y} + 2Tr\left(A_{\text{est}}^T\tilde{C}_a^T J^T P\boldsymbol{y} + \boldsymbol{u}_{\text{cal}}^T D^T B_{\text{est}}^T J^T P\boldsymbol{y}\right) + 2\boldsymbol{E}^T B_{\text{est}}^T J^T P\boldsymbol{y} + 2Tr\left(\tilde{C}_a^T W_1\dot{\tilde{C}}_a + \tilde{D}^T W_2\dot{\tilde{D}}\right) + 2\tilde{\boldsymbol{E}}^T W_3\dot{\tilde{\boldsymbol{E}}} \tag{44}$$

P is selected such that $PA_h + A_h^T P = -Q$, where Q is a positive-definite matrix. The existence of P and Q to satisfy the preceding relation is guaranteed because A_h is Hurwitz. By the use of the identity if A and B are row and column vectors, respectively, then $AB = Tr(BA)$:

$$\dot{V} = -\boldsymbol{y}^T Q\boldsymbol{y} + 2Tr\left[\tilde{C}_a^T\left(J^T P\boldsymbol{y}\boldsymbol{A}_{\text{est}}^T + W_1\dot{\tilde{C}}_a\right)\right] + 2Tr\left[\tilde{D}^T\left(B_{\text{est}}^T J^T P\boldsymbol{y}\boldsymbol{u}_{\text{cal}}^T + W_2\dot{\tilde{D}}\right)\right] + 2\tilde{\boldsymbol{E}}^T\left(B_{\text{est}}^T J^T P\boldsymbol{y} + W_3\dot{\tilde{\boldsymbol{E}}}\right) \tag{45}$$

When only the negative definite part $-\boldsymbol{y}^T Q\boldsymbol{y}$ is retained and all other sign indefinite terms in Eq. (45) are set to zero,

$$\dot{\tilde{C}}_a = -W_1^{-1}\left(J^T P\boldsymbol{y}\boldsymbol{A}_{\text{est}}^T\right) \tag{46}$$

$$\dot{C}_a^* - \dot{C}_a = -W_1^{-1}\left(J^T P\boldsymbol{y}\boldsymbol{A}_{\text{est}}^T\right) \tag{47}$$

However, C_a^* is assumed to be constant, so that

$$\dot{C}_a = W_1^{-1}\left(J^T P\boldsymbol{y}\boldsymbol{A}_{\text{est}}^T\right) \tag{48}$$

Similarly,

$$\dot{D} = W_2^{-1}\left(B_{\text{est}}^T J^T P\boldsymbol{y}\boldsymbol{u}_{\text{cal}}^T\right) \tag{49}$$

$$\dot{\boldsymbol{E}} = W_3^{-1}\left(B_{\text{est}}^T J^T P\boldsymbol{y}\right) \tag{50}$$

These are the update equations for the various adaptive learning parameters. A schematic of a fault-tolerant SAMI controlled is shown in Fig. 2.

F. Stability Analysis

The Lyapunov function V is a function of $(\boldsymbol{y}, \tilde{C}_a, \tilde{D}, \tilde{\boldsymbol{E}})$, and $V = 0$ when $\boldsymbol{y} = 0$, $\tilde{C}_a = 0$, $\tilde{D} = 0$, and $\tilde{\boldsymbol{E}} = 0$. Here, 0 is a null vector or null matrix of appropriate dimension. The derivative of the Lyapunov function $\dot{V}$ is a function of $\boldsymbol{y}$ only. Thus, the derivative of the Lyapunov function is zero when $\boldsymbol{y} = 0$, irrespective of the values of $\tilde{C}_a$, $\tilde{D}$, and $\tilde{\boldsymbol{E}}$. Hence, $\dot{V}$ is negative semidefinite.

Thus, the adaptive control law (20), along with the update laws (48–50), ensure global stability. From the properties of V and $\dot{V}$ just stated, we conclude that $\boldsymbol{y} \in \mathcal{L}_2 \cap \mathcal{L}_\infty$ and $\tilde{C}a$, $\tilde{D}$, and $\tilde{\boldsymbol{E}} \in \mathcal{L}_\infty$.

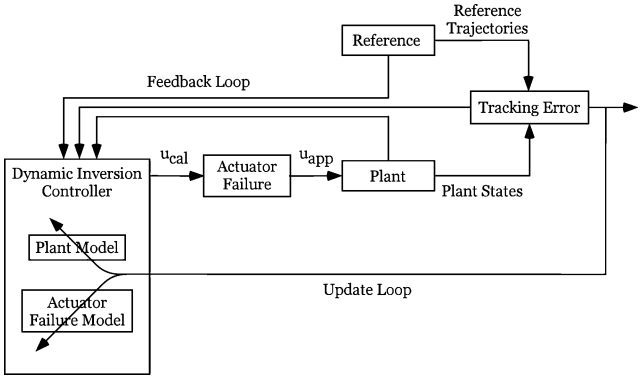

Fig. 2 Schematic of fault-tolerant SAMI controller.

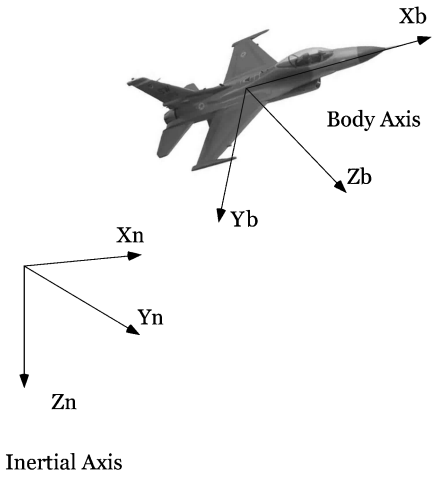

Fig. 3 Inertial and body axis.

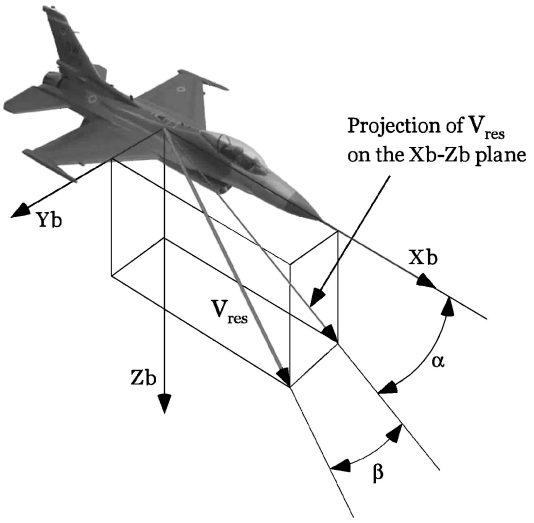

Fig. 4 Angle of attack α and sideslip angle β.

Here, $\boldsymbol{y}$ is defined as $\boldsymbol{y} = \dot{\boldsymbol{s}} + \lambda \boldsymbol{s}$. If $(\dot{\boldsymbol{s}} + \lambda \boldsymbol{s}) \in \mathcal{L}_2 \cap \mathcal{L}_\infty$, then $\boldsymbol{s} \in \mathcal{L}_2 \cap \mathcal{L}_\infty$, and $\dot{\boldsymbol{s}} \in \mathcal{L}_2 \cap \mathcal{L}_\infty$. Because the reference trajectories are bounded, $(\boldsymbol{\sigma}_r, \dot{\boldsymbol{\sigma}}_r) \in \mathcal{L}_\infty$. Thus, from the definition of $\boldsymbol{s}$ and $\dot{\boldsymbol{s}}$, it can be concluded that $(\boldsymbol{\sigma}, \dot{\boldsymbol{\sigma}}) \in \mathcal{L}_\infty$. Because $\dot{\boldsymbol{\sigma}} = J\boldsymbol{\omega}$, $\boldsymbol{\omega} \in \mathcal{L}_\infty$, which implies $A(\boldsymbol{\sigma}, \boldsymbol{\omega})$ and $\boldsymbol{\phi} \in \mathcal{L}_\infty$. Equation (41) shows that all of the signals in the $\dot{y}$ are bounded, hence $\dot{\boldsymbol{y}} \in \mathcal{L}_\infty$.

From Barbalat's lemma (see Ref. 10), we conclude that $\boldsymbol{y} \to 0$ as $t \to \infty$. Thus, $\boldsymbol{s} \to 0$ and $\dot{\boldsymbol{s}} \to 0$, which implies that $\boldsymbol{\sigma} \to \boldsymbol{\sigma}_r$ and $\boldsymbol{\omega} \to \boldsymbol{\omega}_r$ as $t \to \infty$. Thus, the states of the plant asymptotically converge to the reference, and perfect tracking can be achieved.

With perfect tracking, the system will follow a trajectory, and the the tracking error will go to zero as time goes to infinity. However, $\dot{V}$ is a function of the tracking error only, and so when the tracking error becomes zero, the system parameters do not update. The stability analysis guarantees asymptotic tracking of the reference trajectories. However, the adaptively estimated parameters may not converge to the actual parameters of the system during the duration of the maneuver. Because it is assumed that parameters like C_a^*, D^*, and $\boldsymbol{E}^*$ are constants, this formulation works only when the plant parameters are constant with respect to time, or slowly time varying, as compared to the rate of update of the adaptive parameters.

V. Nonlinear Six-Degree-of-Freedom Aircraft Simulation Model

The general six-degree-of-freedom equations of motion for an aircraft are derived for the body axis fixed to body with the origin located at the center of gravity.[22] The orientation and position is defined in terms of an inertial reference frame fixed to the Earth (Fig. 3). Let dx, dy, and dz denote the position of the aircraft along the X_n, Y_n, and Z_n axes, respectively. The angular orientation of the aircraft can be described by a 3–2–1 rotation sequence through the Euler angles ψ, θ, and ϕ, respectively. The vectors $\Omega = [p \;\; q \;\; r]^T$ and $\boldsymbol{V} = [u \;\; v \;\; w]^T$ denote the angular velocity and the linear velocity along the body axes, respectively.

The angle of attack and the sideslip angle are shown in Fig. 4 and can be calculated in terms of the velocity components as

$$\alpha = \tan^{-1}(w/u) \tag{51}$$

$$\beta = \sin^{-1}(v/V_{\text{res}}) \tag{52}$$

where V_{res} is the resultant velocity. The variables that completely define the state space for this aircraft model are the position level vector

$$\boldsymbol{\sigma} = [\phi \quad \theta \quad \psi \quad dx \quad dy \quad dz]^T \tag{53}$$

and the velocity level vector

$$\boldsymbol{\omega} = [p \quad q \quad r \quad u \quad v \quad w]^T \tag{54}$$

The aircraft model used in the examples is an F-16 type with thrust-vectoring capability. Following the earlier discussion, it is clear that the number of controls must be greater than the number of velocity level states. There are eight aerodynamic controls: right horizontal tail, left horizontal tail, right aileron, left aileron, rudder, and three controls for thrust vectoring: thrust along the X, Y, and Z axes. With a total of eight controls, the control algorithm can tolerate a maximum of two control failures, both of which may fail at arbitrary times.

We define the notation $S_\theta = \sin(\theta)$, $C_\theta = \cos(\theta)$, etc., for writing convenience. The structured model for this example with separate kinematic and dynamic parts can be written as 1) the kinematic part,

$$[\dot{\phi} \quad \dot{\theta} \quad \dot{\psi} \quad \dot{dx} \quad \dot{dy} \quad \dot{dz}]^T = \begin{bmatrix} 1 & S_\phi \tan(\theta) & C_\phi \tan\theta & 0 & 0 & 0 \\ 0 & C_\phi & -S_\phi & 0 & 0 & 0 \\ 0 & S_\phi \sec\theta & C_\phi \sec\theta & 0 & 0 & 0 \\ 0 & 0 & 0 & C_\theta C_\psi & S_\phi S_\theta C_\psi - C_\phi S_\psi & C_\phi S_\theta C_\psi - S_\phi S_\psi \\ 0 & 0 & 0 & C_\theta S_\psi & S_\phi S_\theta S_\psi - C_\phi C_\psi & C_\phi S_\theta S_\psi - S_\phi C_\psi \\ 0 & 0 & 0 & -S_\theta & S_\phi C_\theta & C_\phi C_\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \\ u \\ v \\ w \end{bmatrix} \tag{55}$$

which is in the form of the model, Eq. (4)

$$\dot{\boldsymbol{\sigma}} = J(\boldsymbol{\sigma})\boldsymbol{\omega}$$

and 2) the dynamic part, where I is allowed be the moment of inertia of the aircraft about the body axes and m the mass.
The inertia matrix is

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \tag{56}$$

From Euler's rigid-body equations,

$$\dot{\Omega} = I^{-1}\left[\begin{bmatrix} \boldsymbol{L} \\ \boldsymbol{M} \\ \boldsymbol{N} \end{bmatrix} - \tilde{\Omega} I \boldsymbol{\Omega}\right] \tag{57}$$

$$\dot{\boldsymbol{V}} = \frac{1}{m}\begin{bmatrix} \boldsymbol{X} - mgS_\theta \\ \boldsymbol{Y} + mgC_\theta S_\phi \\ \boldsymbol{Z} + mgC_\theta C_\phi \end{bmatrix} \tag{58}$$

where $\tilde{\Omega}(\,)$ is the matrix representation of the cross product between vector $\boldsymbol{\Omega}$ and any compatible vector $(\,)$, $[\boldsymbol{X}\ \ \boldsymbol{Y}\ \ \boldsymbol{Z}]^T$ is the force vector and $[\boldsymbol{L}\ \ \boldsymbol{M}\ \ \boldsymbol{N}]^T$ is the moment vector in the body axes,

$$\tilde{\Omega} = \begin{bmatrix} 0 & -r & -q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \tag{59}$$

The external moments and forces have wing and body contributions that are dependent on angle of attack, sideslip angle, and the controls.

Consider the total external pitching moment acting on the aircraft,

$$\boldsymbol{M} = \left[\tfrac{1}{2}\rho \boldsymbol{V}^2 S\left(C_{mm_\beta}\beta + C_{mm_p}p + C_{mm_r}r\right)\right] + \left[\boldsymbol{C}_{mm_{\text{cont}}}\boldsymbol{u}\right] \tag{60}$$

where $\frac{1}{2}\rho \boldsymbol{V}^2$ is the dynamic pressure; S is the wing planform area; $\boldsymbol{C}_{mm_\beta}$, $\boldsymbol{C}_{mm_p}$, and $\boldsymbol{C}_{mm_r}$ are the corresponding stability derivatives; and $\boldsymbol{C}_{mm_{\text{const}}}$ is a row vector. The ith element of this vector represents the contribution to the pitching moment due to unit deflection of the ith control.

Because external force and moment contributions in the first term of Eq. (60) play a role in the unforced dynamic behavior, they constitute a part of the $\boldsymbol{A}$ vector. The external force and moment contributions in the second term of Eq. (60) show the control influence and, hence, constitute a part of the B matrix.

The dynamic part of the equation of motion can now be written as

$$\begin{bmatrix} \dot{\Omega} \\ \dot{\boldsymbol{V}} \end{bmatrix} = \begin{bmatrix} I^{-1}\left[\begin{bmatrix} L \\ M \\ N \end{bmatrix}_{\text{unforced}} - \tilde{\Omega} I \Omega\right] \\ \frac{1}{m}\left\{\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{\text{unforced}} + \begin{bmatrix} -mgS_\theta \\ mgC_\theta S_\phi \\ mgC_\theta C_\phi \end{bmatrix}\right\} - \tilde{\omega}\boldsymbol{V} \end{bmatrix} + \begin{bmatrix} C_{u_{\text{cont}}} \\ C_{mm_{\text{cont}}} \\ C_{nn_{\text{cont}}} \\ C_{x_{\text{cont}}} \\ C_{y_{\text{cont}}} \\ C_{z_{\text{cont}}} \end{bmatrix}\boldsymbol{u} \tag{61}$$

which has the same form as Eq. (5),

$$\dot{\omega} = \boldsymbol{A}(\boldsymbol{\sigma}, \boldsymbol{\omega}) + B(\boldsymbol{\sigma}, \boldsymbol{\omega})\boldsymbol{u}_{\text{app}}$$

as desired.

VI. Numerical Example

The example consists of an aileron actuator failure that occurs from an initially trimmed flight condition. The flight condition is Mach 0.8 at 50,000 ft. The simulations are performed with parametric errors in the vector $\boldsymbol{A}$ and the B matrix and with initial condition errors. Parametric errors of the order of $\pm 10\%$ are introduced in the various parameters such as weight and various inertia terms. The aircraft is given an initial angular velocity of 1 deg/s in all of the three body axes, to simulate initial condition errors. For each case, the right aileron fails at time equal to 5 s and then settles down to a specified constant value. Because of this failure, the aircraft rolls rapidly and without adaptation departs controlled flight. The adaptive controller is expected to adjust its parameters so that it compensates for this failure and provides a restoring rolling moment by using a differential horizontal tail. Two test cases are presented: aileron failure during steady, level, 1-g flight (case 1), and aileron failure during a steady level turn (case 2).

A. Case 1: Steady Level 1g Flight

The trim angle of attack is 3 deg, and, on failure, the aileron settles down to 2 deg. For case 1, the initial reference states are contained in Table 1.

Figure 5 shows the angular states, and Figure 6 shows the translational states. Because of failure of the right aileron, the aircraft begins to roll rapidly, and, with the nonadaptive controller unable to handle the failure, the states diverge and the aircraft departs from controlled flight. The same failure scenario, but with the adaptive controller operating is shown in Fig. 7. Deviations in the states from the reference when time is less than 5 s is due to initial condition errors and the parametric error in the system matrices. However, the states settle down to their reference values by the time the failure occurs at time equals 5 s. All of the states show perturbations at the onset of failure, but the controller adapts, and the states settle down to their reference values. The update parameters (not shown here) stop updating as soon as the tracking error goes to zero. Figure 8 shows the controls for both the nonadaptive and adaptive controllers.

Table 1 Reference states for case 1

Variable	State
$\phi, dy, dz, p, q, r, \beta, \nu$	0
$\theta = \alpha$	constant
u	$V_{\text{res}} \cos\alpha$
w	$V_{\text{res}} \sin\alpha$
ψ	constant
dx	$(V_{\text{res}})t$ where t is time
V_{res}	constant

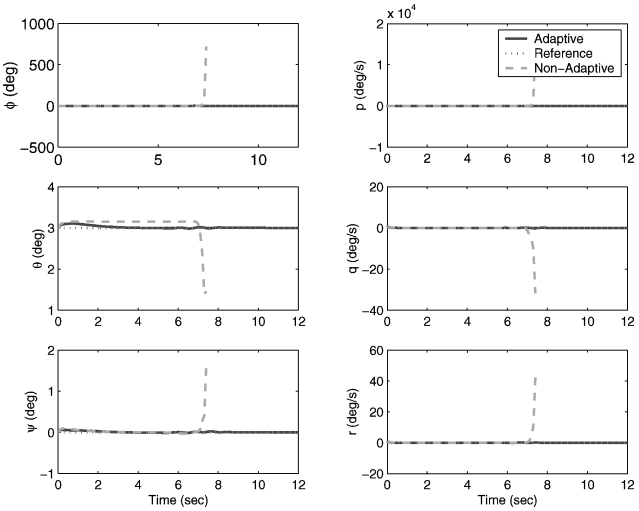

Fig. 5 Time histories of angular states, case 1, steady level flight.

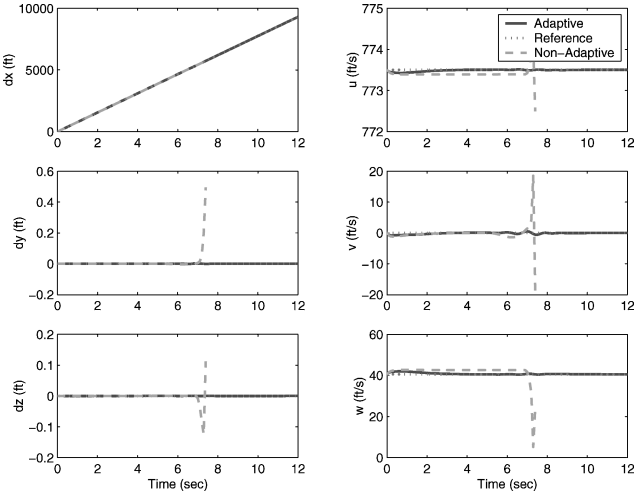

Fig. 6 Time histories of linear states, case 1, steady level flight.

Table 2 Reference states for case 2

Variable	State
α	constant
ψ	$\dot{\psi}t$
θ	constant
ϕ	constant
dx	$V_{\text{res}}t$
dy	$dx = \beta = v = 0$
p	$-\sin\theta\dot{\psi}$
q	$\cos\theta\sin\phi\dot{\psi}$
r	$\cos\theta\cos\phi\dot{\phi}$
u	$V_{\text{res}}\cos\alpha$
w	$V_{\text{res}}\sin\alpha$
V_{res}	constant

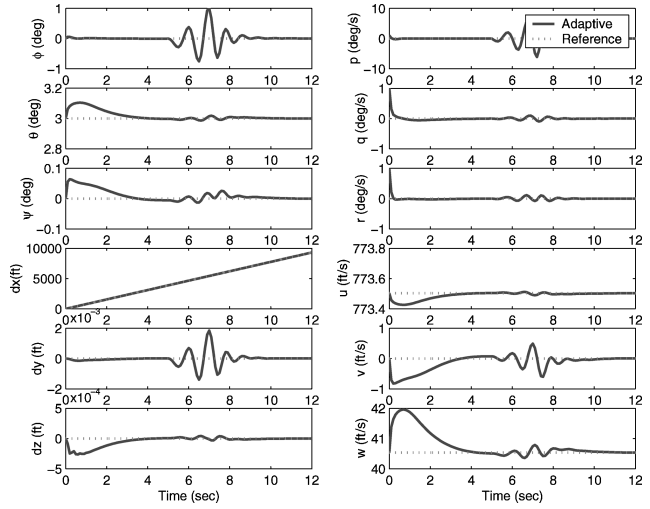

Fig. 7 Time histories of states with adaptive control, case 1, steady level flight.

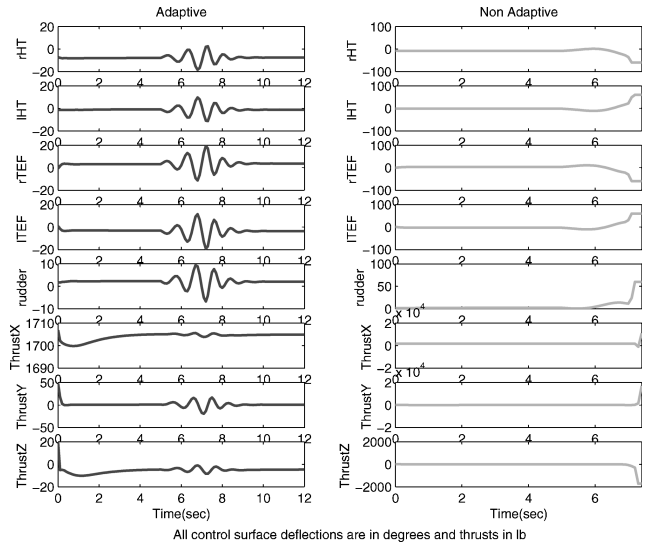

Fig. 8 Time histories of controls, adaptive and nonadaptive, case 1, steady level flight.

For the latter, recovery from failure and the performance shown is achieved with reasonable control deflection limits. The control rates (not shown here) are also within reasonable limits.

B. Case 2: Steady Level Turn

Trim angle of attack is 14 deg, bank angle is 52.4 deg, and turn rate is 3 deg/s. On failure, the aileron settles down to 1 deg. There is no initial condition error and no uncertainty in the system parameters. The reference states are shown in Table 2.

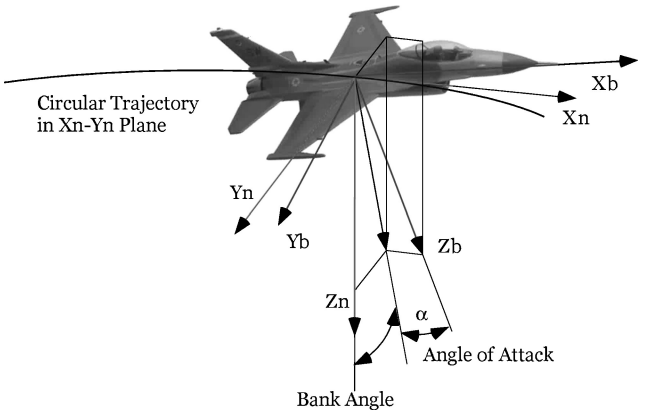

Fig. 9 Case 2 steady level turn.

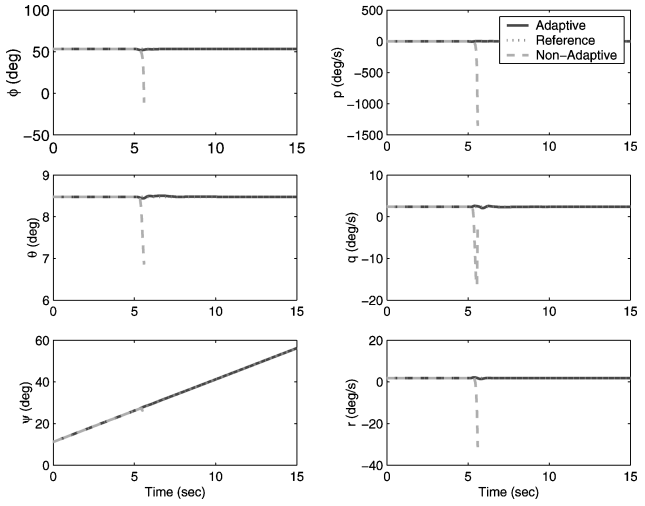

Fig. 10 Time histories of angular states, case 2, steady level turn.

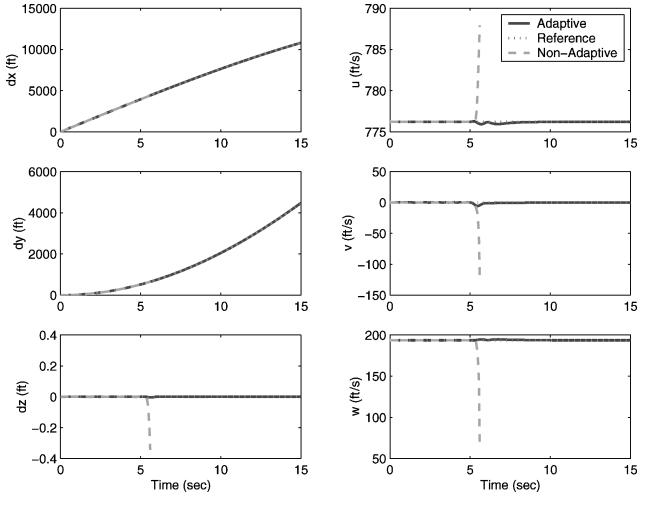

Fig. 11 Time histories of linear states, case 2, steady level turn.

The reference states for a steady level turn are a constant turn rate $\dot{\psi}$ and bank angle bank $= \tan^{-1}(\dot{\psi}V/g)$ (Fig. 9). The transformation from inertial to body axes is a 3–1–2 Euler angle rotation through ψ, bank, and α, respectively. This 3–1–2 rotation is converted to a 3–2–1 rotation to obtain ψ, θ, and ϕ.

Figures 10 and 11 show that the nonadaptive controller provides perfect tracking before failure, but diverges rapidly afterward. In Fig. 12, all of the states show perturbations at failure, but the controller successfully adapts, and the states settle down to their reference values. The adaptive learning parameters remain constant

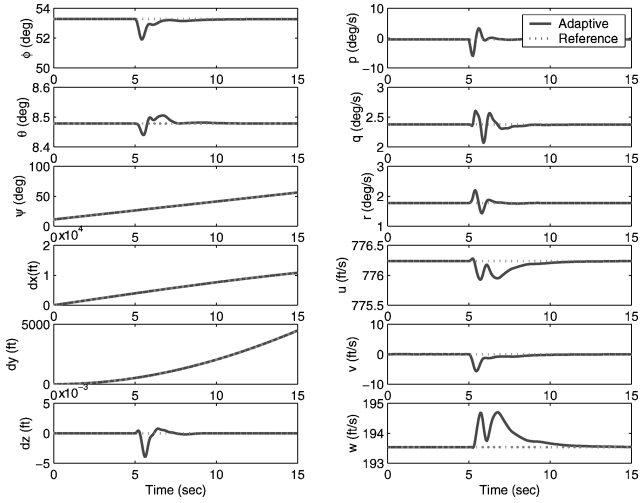

Fig. 12 Time histories of states with adaptive control, case 2, steady level turn.

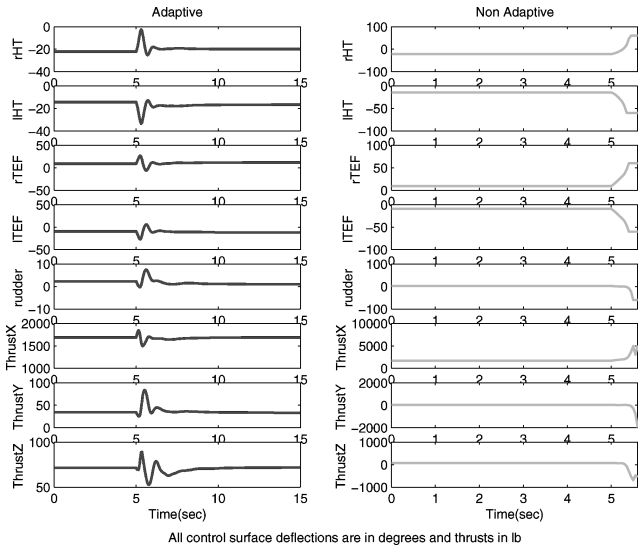

Fig. 13 Time histories of controls, adaptive and nonadaptive, case 2, steady level turn.

before failure because there are no initial condition errors and no errors in system parameters, but they update after failure and settle down to constant values that are different from the initial values. Figure 13 shows that the adaptive controller recovers from the failure with reasonable control effort and within position limits. The control rates (not shown) are also within limits.

VII. Conclusions

This paper derives and validates a fault-tolerant SAMI control law. A mathematical model of control effector failure was added to the plant model, so that the failure can be identified as a change in the system parameters. The control law ensures global asymptotic stability of the tracking errors, even though adaptive learning parameters may not converge to the actual parameters.

Based on the results presented in this paper, the following conclusions are made:

1) For the test cases evaluated, the fault tolerant adaptive controller is able to recover from actuator failure and achieve close tracking in cases where the nonadaptive controller loses tracking or is unable to maintain stability.

2) The fault-tolerant adaptive controller formulation presented can only handle parametric uncertainties that are constant with respect to time, or slowly time varying as compared to the rate of update of the adaptive learning parameters of the system. Thus, it can handle actuator failures in which the parameters in the failure model remain constant/piecewise constant after the failure. This constant value may or may not be zero. This conclusion is valid with the assumption that the system is controllable with the remaining active controls, such that the control effectiveness of the remaining controls is large enough to counter the undesirable effect produced when the failed actuator settles to an arbitrary value.

References

[1]Bodson, M., and Groszkiewicz, J., "Multivariable Adaptive Algorithms for Reconfigurable Flight Control," *IEEE Transactions on Control Systems Technology*, Vol. 5, No. 2, 1997, pp. 217–229.

[2]Boskovic, J. D., and Mehra, R. K., "Multiple Model Adaptive Flight Control Scheme for Accommodation of Actuator Failures," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 4, 2002, pp. 712–724.

[3]Ahmed-Zaid, F., Ioannou, P., Gousman, K., and Rooney, R., "Accommodation of Failures in the F-16 Aircraft Using Adaptive Control," *IEEE Control Systems Magazine*, Vol. 11, No. 1, 1991, pp. 73–78.

[4]"Reconfigurable Systems for Tailless Fighter Aircraft RESTORE," Scientific and Technical Repts., System Design Rept., Contract Data Requirement List (CDRL) Sequence A007 F33615-96-C-3612, Boeing Phantom Works, St. Louis, MO, May 1998.

[5]Brinker, J., and Wise, K., "Reconfigurable Flight Control of a Tailless Advanced Fighter Aircraft," AIAA Paper A98-37001, Aug. 1998.

[6]Calise, A., Lee, S., and Sharma, M., "Direct Adaptive Reconfigurable Control of a Tailless Fighter Aircraft," *Proceedings of the 1998 AIAA Guidance, Navigation, and Control Conference*, Vol. 1, AIAA, Reston, VA, 1998, pp. 88–97.

[7]Boskovic, J. D., and Mehra, R. K., "Robust Fault-Tolerant Control Design for Aircraft Under State-Dependent Disturbances," AIAA Paper 2003-5490, Aug. 2003.

[8]Astrom, K. J., and Wittenmark, B., *Adaptive Control*, 2nd ed., Prentice–Hall, Upper Saddle River, NJ, 1994.

[9]Slotine, J., and Li, W., *Applied Nonlinear Control*, Prentice–Hall, Upper Saddle River, NJ, 1990.

[10]Khalil, H. K., *Nonlinear Systems*, 3rd ed., Prentice–Hall, Upper Saddle River, NJ, 2001.

[11]Narendra, K. S., and Annaswamy, A., *Stable Adaptive Systems*, Prentice–Hall, Upper Saddle River, NJ, 1989.

[12]Sastry, S., and Bodson, M., *Adaptive Control: Stability, Convergence, and Robustness*, Prentice–Hall, Upper Saddle River, NJ, 1989.

[13]Iannou, P. A., and Sun, J., *Stable and Robust Adaptive Control*, Prentice–Hall, Upper Saddle River, NJ, 1995.

[14]Akella, M. R., *Structured Adaptive Control: Theory and Applications to Trajectory Tracking in Aerospace Systems*, Ph.D. Dissertation, Dept. of Aerospace Engineering, Texas A&M Univ., College Station, TX, 1998.

[15]Schaub, H., Akella, M. R., and Junkins, J. L., "Adaptive Realization of Linear Closed-Loop Tracking Dynamics in the Presence of Large System Model Errors," *Journal of Astronautical Sciences*, Vol. 48, No. 4, 2000, pp. 537–551.

[16]Akella, M. R., and Junkins, J. L., "Structured Model Reference Adaptive Control in the Presence of Bounded Disturbances," *Advances in the Astronautical Sciences*, Vol. 99, No. 1, 1998, pp. 98–121.

[17]Akella, M. R., and Junkins, J. L., "Structured Model Reference Adaptive Control with Actuator Saturation Limits," AIAA Paper 98-4472, Aug. 1998.

[18]Subbarao, K., *Structured Adaptive Model Inversion: Theory and Applications to Trajectory Tracking for Non-Linear Dynamical Systems*, Ph.D. Dissertation, Dept. of Aerospace Engineering, Texas A&M Univ., College Station, TX, Aug. 2001.

[19]Subbarao, K., Verma, A., and Junkins, J. L., "Structured Adaptive Model Inversion Applied to Tracking Spacecraft Maneuvers," AAS/AIAA Spaceflight Mechanics Meeting, American Astronautical Society, Paper AAS-00-202, Jan. 2000.

[20]Subarrao, K., Steinburg, M., and Junkins, J. L., "Structured Adaptive Model Inversion Applied to Tracking Aggressive Aircraft Maneuvers," AIAA Paper 2001-4019, Aug. 2001.

[21]Ward, D. G., Monaco, J. F., Barron, R. L., Bird, R. A., Virnig, J. C., and Landers, T. F., "Self-Designing Controller Design, Simulation, and Flight Test Evaluation," Air Vehicles Directorate, Tech. Rept. WL-TR-97-3095, U.S. Air Force Research Lab., U.S. Air Force Material Command, WL/FIGC, Wright–Patterson AFB, OH, 1996.

[22]Roskam, J., *Airplane Flight Dynamics and Automatic Flight Control*, Vol. 1, Design, Analysis, and Research Corp., Lawrence, KS, 1994, pp. 3–26, 65–122.