ACC01-IEEE1443

Intelligent Fault Tolerant Control Using Adaptive Schemes and Multiple Models *

Yixin Diao and Kevin M. Passino

Department of Electrical Engineering
The Ohio State University
2015 Neil Avenue, Columbus, OH 43210

Abstract

Fault tolerant control for nonlinear time-varying systems is of both theoretical and practical importance. In this paper we present an intelligent fault tolerant control methodology using an on-line approximation based adaptive controller to try to achieve system stability and robustness and a fault diagnosis unit to provide fault information and then reconfigure the control law. Simulation examples for a fault tolerant engine control problem are given to demonstrate the effectiveness of the proposed scheme.

1 Introduction

The last two decades have seen continuous improvement in control techniques resulting from the spectacular progresses in control theory and computer technology. Meanwhile, stimulated by the growing demand for improving the reliability and performance of systems, many fault diagnosis and fault tolerant control methods have been developed, which have the capability of detecting the occurrence of faults and retaining satisfactory system performance in the presence of faults [1, 2, 3].

Fault tolerance of dynamic systems can be achieved either from system robustness to faults as well as other uncertainties, or from controller reconfiguration or restructuring in response to specific faults. Actually, a well designed control system may have some fault tolerance capabilities in that it can be designed (e.g, by properly choosing feedback gains) to compensate for some system uncertainties such as disturbances and noise, and the fault can be considered as a certain kind of system uncertainty. Since no information about faults is typically utilized by control systems (e.g., in linear "robust control"), this type of control system may be referred

*This work was supported by the NASA Glenn Research Center under grant NAG3-2084. The authors would like to thank Dr. S. Adibhatla at General Electric Aircraft Engines for providing the CLM for the engine, technical advice on how to use it, and feedback on the development of the modeling and control strategies. Please address all correspondence to K. Passino (passino@ee.eng.ohio-state.edu, (614) 292-5716).

to as a "passive fault tolerant control system" [4, 5]. However, the magnitude of faults that can be accommodated by a fixed control structure and parameters without using the knowledge of faults is often limited and more restricted than that of a reconfigurable controller (at least that is what has been found via current simulation studies). By utilizing the fault information obtained from fault detection and identification, "reconfigurable control" modifies the control function (parameters or structures) in response to the faults so that it is referred to as "active fault tolerant control." An active fault tolerant control system can be obtained by control law re-scheduling [6, 7], linear-quadratic control [8, 9], pseudo-inverse methods [10], or adaptive control methods [11, 12, 13, 14, 15, 16, 17, 18, 19]. Since most plants are inherently nonlinear and the faults may often amplify the nonlinearities by driving the plants from a relatively "linear" operating point into a more nonlinear operating region, the study of fault tolerant control for nonlinear systems has become a very active research topic for both theoretical and practical reasons [19, 20, 21]. Furthermore, the existing work in the literature mainly considers fault tolerant control in the context of time invariant systems (as if a fault has already occurred).

In this paper we propose an intelligent fault tolerant control methodology for time-varying nonlinear systems. In Section 2 on-line approximation based stable indirect adaptive controllers are constructed for nonlinear time varying systems, which serves as a robust baseline controller. In the fault diagnosis system that we describe in Section 3, a bank of multiple models are used, and an expert supervisory scheme determines the proper model bank to generate residuals and analyzes the residuals to detect and identify faults. In Section 4 the proposed method is applied to a turbine engine (General Electric XTE46) to demonstrate its effectiveness.

2 On-line Approximation Based Adaptive Control

Consider the following single-input single-output nonlinear time-varying system

$$\dot{x} = f(x,t) + g(x,t)u \tag{1}$$

$$y = h(x,t) \tag{2}$$

where $x = [x_1, x_2, \ldots, x_n]^\top$ is the state vector, u is the (scalar) input and y is the (scalar) output of the system. The functions $f : D \times [0,\infty) \to \Re^n$, $g : D \times [0,\infty) \to \Re^n$ and $h : D \times [0,\infty) \to \Re$ are smooth time varying functions, and $D \subset \Re^n$ is a domain that contains the origin $x = 0$. For convenience we assume that if $u = 0, \forall t \geq 0$, the origin is an equilibrium point at $t = 0$ and for all subsequent times, that is, $f(0,t) = 0, \forall t \geq 0$.

Note that the standard Lie derivative and strong relative degree for time invariant systems [22] are not adequate for time-varying systems, and modifications need to be made to explicitly account for the time variability of the system. Let $\bar{L}_f^d h(x,t)$ be the d^{th} Lie derivative of $h(x,t)$ with respect to $f(x,t)$. In particular, define

$$\bar{L}_f h(x,t) = \frac{\partial h}{\partial t} + \left(\frac{\partial h}{\partial x}\right)^\top f(x,t) \tag{3}$$

and, for example,

$$\bar{L}_f^2 h(x,t) = \bar{L}_f[\bar{L}_f h(x,t)] = \frac{\partial \bar{L}_f h}{\partial t} + \left(\frac{\partial \bar{L}_f h}{\partial x}\right)^\top f(x,t) \tag{4}$$

Note that the *modified* Lie derivative $\bar{L}_f^d h(x,t)$ for time-varying systems is a straightforward extension of the standard Lie derivative $L_f^d h(x)$ (so that no specific definition is typically given in differential geometry). Particularly, define $x_0 = t$ and consider the "extended" vectors $X = [x_0, x_1, x_2, \ldots, x_n]^\top$ and $F(X) = [1, f(X)^\top]^\top$ (note that $\dot{x}_0 = \dot{t} = 1$), so that we have

$$L_F h(X) = \left(\frac{\partial h}{\partial X}\right)^\top F(X) = \frac{\partial h}{\partial t} + \left(\frac{\partial h}{\partial x}\right)^\top f(x,t) = \bar{L}_f h(x,t)$$

Next, we define the "strong relative degree" of the time-varying system. A system is said to have strong relative degree d, $1 \le d \le n$, in a region $D_0 \subset D$ if

$$L_g h(x,t) = L_g \bar{L}_f h(x,t) = \cdots = L_g \bar{L}_f^{d-2} h(x,t) = 0 \tag{5}$$

and

$$L_g \bar{L}_f^{d-1} h(x,t) \neq 0 \tag{6}$$

for all $x \in D_0$ and $t \in [0, \infty)$. Note that we use both the standard and the modified Lie derivatives above to provide a compact representation of the definition.

Under the above definitions, if system (1) and (2) have strong relative degree d, then

$$\begin{aligned}
\dot{y} &= \frac{\partial h}{\partial t} + \left(\frac{\partial h}{\partial x}\right)^\top [f(x,t) + g(x,t)u] \\
&= \left[\frac{\partial h}{\partial t} + \left(\frac{\partial h}{\partial x}\right)^\top f(x,t)\right] + \left[\left(\frac{\partial h}{\partial x}\right)^\top g(x,t)\right] u \\
&= \bar{L}_f h(x,t) \\
\ddot{y} &= \frac{\partial \bar{L}_f h}{\partial t} + \left(\frac{\partial \bar{L}_f h}{\partial x}\right)^\top [f(x,t) + g(x,t)u] \\
&= \left[\frac{\partial \bar{L}_f h}{\partial t} + \left(\frac{\partial \bar{L}_f h}{\partial x}\right)^\top f(x,t)\right] + \left[\left(\frac{\partial \bar{L}_f h}{\partial x}\right)^\top g(x,t)\right] u \\
&= \bar{L}_f^2 h(x,t)
\end{aligned}$$

and so on, so that the system dynamics may be written in the normal form as

$$
\begin{aligned}
\dot{\xi}_1 &= \xi_2 = \bar{L}_f h(x,t) \\
\dot{\xi}_2 &= \xi_3 = \bar{L}_f^2 h(x,t) \\
&\vdots \\
\dot{\xi}_{d-1} &= \xi_d = \bar{L}_f^{d-1} h(x,t) \\
\dot{\xi}_d &= \bar{L}_f^d h(x,t) + L_g \bar{L}_f^{d-1} h(x,t) u = \alpha(\xi,\pi,t) + \beta(\xi,\pi,t) u \\
\dot{\pi} &= f_0(\xi,\pi,t)
\end{aligned}
$$

with $\xi \in \Re^d$, $\pi \in \Re^{n-d}$, and $\xi_1 = y$. This transformation of the model form can be taken by a change of variables

$$
z = T(x,t) = \begin{bmatrix} \psi_1(x,t) \\ \vdots \\ \psi_d(x,t) \\ \cdots\cdots \\ \phi_1(x,t) \\ \vdots \\ \phi_{n-d}(x,t) \end{bmatrix} = \begin{bmatrix} \psi(x,t) \\ \cdots\cdots \\ \phi(x,t) \end{bmatrix} = \begin{bmatrix} \xi \\ \cdots \\ \pi \end{bmatrix} \tag{7}
$$

where $\psi(x,t)$ and $\phi(x,t)$ are given by

$$
\begin{aligned}
\psi_1(x,t) &= h(x,t) && (8) \\
\psi_{i+1}(x,t) &= \frac{\partial \psi_i}{\partial t} + \left(\frac{\partial \psi_i}{\partial x}\right)^\top f(x,t) && (9) \\
\left(\frac{\partial \phi_j}{\partial x}\right)^\top g(x,t) &= 0 && (10)
\end{aligned}
$$

for $i = 1, 2, \ldots, d-1$, $j = 1, 2, \ldots, n-d$, $\forall x \in D_x$ and $\forall t \in [0, \infty)$, so that $T(x,t)$ is a diffeomorphism on a domain $D_x \subset D_0$. It has been shown in [23] that if the system (1, 2) has relative degree $d < n$ in a domain D_0 for all $t \in [0,\infty)$, then there exist functions ϕ_1 to ϕ_{n-d} that satisfy (10) and make $T(x,t)$ a diffeomorphism. A "global" normal form can be obtained if and only if the conditions (8, 9, 10) hold for all $x \in \Re^n$ and T is proper, that is, $\lim_{\|x\| \to \infty} \|T(x,t)\| = \infty$.

For convenience, we assume that $x = 0$ is an equilibrium point of the original system and y vanishes at $x = 0$, that is, $f(0,t) = 0, h(0,t) = 0, \forall t \geq 0$. By choosing $\phi(x,t)$ such that $\phi(0,t) = 0, \forall t \geq 0$, the equilibrium point of the transformed system is defined by $\bar{\xi} = [h(0,t), 0, \ldots, 0]^\top = 0, \bar{\pi} = \phi(0,t) = 0$.

The normal form decomposes the system states into an external part ξ and an internal part π. For the external part, if we let $y^{(d)}$ denote the d^{th} derivative of y, it may be rewritten as

$$
y^{(d)} = [\alpha_k(t) + \alpha(x,t)] + [\beta_k(t) + \beta(x,t)]u \tag{11}
$$

where $\alpha_k(t)$ and $\beta_k(t)$ are "known" dynamics of the system (which are assumed to be bounded if x is bounded), and $\alpha(x,t)$ and $\beta(x,t)$ represent nonlinear time-varying dynamics of the plant that are unknown. We assume that for some known $\beta_0 > 0$, we have $|\beta_k(t)+\beta(x,t)| \geq \beta_0$ so that it is always bounded away from zero (for convenience we further assume that $\beta_k(t) + \beta(x,t) > 0$, however, the following analysis may easily be modified for systems which are defined with $\beta_k(t) + \beta(x,t) < 0$). The external part may be stabilized by the control u (which we will show later), while the internal part is made unobservable by the same control. By having $\xi = 0$ in the inner part, the "zero dynamics" of the system are given by

$$\dot{\pi} = f_0(0, \pi, t) \tag{12}$$

If the plant is of relative degree $d = n$, then there are no zero dynamics. Alternatively, if the relative degree $d < n$, we assume that the zero dynamics are uniformly exponentially attractive so that we have

$$\begin{aligned} \gamma_1 \|\pi\|^2 \leq v_1(\pi, t) &\leq \gamma_2 \|\pi\|^2 && (13) \\ \frac{\partial v_1}{\partial t} + \left(\frac{\partial v_1}{\partial \pi}\right)^\top f_0(0, \pi, t) &\leq -\gamma_3 \|\pi\|^2 && (14) \\ \left\|\frac{\partial v_1}{\partial \pi}\right\| &\leq \gamma_4 \|\pi\| && (15) \end{aligned}$$

for some positive constants γ_1, γ_2, γ_3, and γ_4. If $f_0(0, \pi, t)$ is Lipschitz in ξ, then $\|f_0(\xi, \pi, t) - f_0(0, \pi, t)\| \leq k_1 \|\xi\|$ for some positive k_1. Now, if we have some control u so that $\|\xi\| \leq k_2$ where k_2 is some positive constant, we have

$$\begin{aligned} \dot{v}_1 &= \frac{\partial v_1}{\partial t} + \left(\frac{\partial v_1}{\partial \pi}\right)^\top f_0(\xi, \pi, t) \\ &= \frac{\partial v_1}{\partial t} + \left(\frac{\partial v_1}{\partial \pi}\right)^\top f_0(0, \pi, t) + \left(\frac{\partial v_1}{\partial \pi}\right)^\top [f_0(\xi, \pi, t) - f_0(0, \pi, t)] \\ &\leq -\gamma_3 \|\pi\|^2 + \gamma_4 k_1 \|\xi\| \|\pi\| \\ &\leq -\gamma_3 \|\pi\|^2 + \gamma_4 k_1 k_2 \|\pi\| \end{aligned}$$

Therefore, $\dot{v}_1 \leq 0$ if $\|\pi\| \geq \gamma_4 k_1 k_2 / \gamma_3$. This ensures boundedness of π uniformly.

The on-line learning abilities of neural networks are considered here to approximate the time-varying dynamics of the nonlinear system. In particular, the linear in the parameter radial basis function networks may take the form of

$$\begin{aligned} \hat{\alpha}(x, t) &= \theta_\alpha^\top(t) \phi_\alpha(x) && (16) \\ \hat{\beta}(x, t) &= \theta_\beta^\top(t) \phi_\beta(x) && (17) \end{aligned}$$

where the vectors $\theta_\alpha(t)$ and $\theta_\beta(t)$ are updated on line and are assumed to be defined within the compact parameter sets Ω_α and Ω_β, respectively. In addition, we define the subspace $S_x \subseteq \Re^n$ as the space through which the state trajectory may travel under closed-loop control (we are making no *a priori* assumptions here about the size of S_x). We also define the actual system as

$$\alpha(x,t) = \theta_\alpha^{*\top}(t)\phi_\alpha(x) + \omega_\alpha(x,t) \tag{18}$$
$$\beta(x,t) = \theta_\beta^{*\top}(t)\phi_\beta(x) + \omega_\beta(x,t) \tag{19}$$

where

$$\theta_\alpha^*(t) = \arg\min_{\theta_\alpha\in\Omega_\alpha}\left(\sup_{x\in S_x}|\theta_\alpha^\top\phi_\alpha(x) - \alpha(x,t)|\right) \tag{20}$$
$$\theta_\beta^*(t) = \arg\min_{\theta_\beta\in\Omega_\beta}\left(\sup_{x\in S_x}|\theta_\beta^\top\phi_\beta(x) - \beta(x,t)|\right) \tag{21}$$

are the optimal time-varying parameters and $\omega_\alpha(x,t)$ and $\omega_\beta(x,t)$ are approximation errors which arise when $\alpha(x,t)$ and $\beta(x,t)$ are represented by finite size approximators. We assume that

$$|\omega_\alpha(x,t)| \leq W_\alpha(x) \tag{22}$$
$$|\omega_\beta(x,t)| \leq W_\beta(x) \tag{23}$$

where $W_\alpha(x)$ and $W_\beta(x)$ are known state dependent time invariant bounds on the error in representing the actual system with approximators. We also define parameter errors to be

$$\tilde\theta_\alpha(t) = \theta_\alpha(t) - \theta_\alpha^*(t) \tag{24}$$
$$\tilde\theta_\beta(t) = \theta_\beta(t) - \theta_\beta^*(t) \tag{25}$$

We view adaptive control to be a tracking problem, that is, to design a control system which will cause the output $y(t)$ and its derivatives $\dot y(t),\ldots,y^{(d)}(t)$ to track a desired reference trajectory $y_m(t)$ and its derivatives $\dot y_m(t),\ldots,y_m^{(d)}(t)$, respectively, which we assume to be bounded. The reference trajectory may be defined by a reference signal whose first d derivatives may be measured, or by any reference input $r(t)$ passing through a reference model, with relative degree equal to or greater than d. In particular, a linear reference model may be

$$\frac{Y_m(s)}{R(s)} = \frac{q(s)}{p(s)} = \frac{q_0}{s^d + p_{d-1}s^{d-1} + \cdots + p_0} \tag{26}$$

where $p(s)$ is the pole polynomial with stable roots.

The indirect adaptive control law

$$u = u_{ce} + u_{si} \tag{27}$$

is comprised of a "certainty equivalence" control term u_{ce} and a "sliding mode" control term u_{si}. Let the tracking error be $e(t) = y_m(t) - y(t)$ and a measure of the tracking error be $e_s(t) = e^{(d-1)}(t) + k_{d-2}e^{(d-2)}(t) + \cdots + k_1\dot e(t) + k_0 e(t)$, that is, in the frequency domain, $e_s(s) = L(s)e(s)$ with $L(s) = s^{(d-1)} + k_{d-2}s^{(d-2)} + \cdots + k_1 s + k_0$ whose roots are chosen to be in the (open) left half plane. Also, for convenience below we let $\bar e_s(t) = \dot e_s(t) - e^{(d)}(t)$. Notice that our control goal is to drive $e_s(t) \to 0$ as $t \to \infty$ and the shape of the error dynamics is dictated by the choice of the design parameters in $L(s)$. The certainty equivalence control term is defined as

$$u_{ce} = \frac{1}{\beta_k(t) + \hat{\beta}(x,t)}(-(\alpha_k(t) + \hat{\alpha}(x,t)) + \nu(t)) \tag{28}$$

where $\beta_k(t) + \hat{\beta}(x,t)$ is bounded away from zero (which will be ensured later using projection) so that u_{ce} is well-defined, and

$$\nu(t) = y_m^{(d)} + \eta e_s + \bar{e}_s \tag{29}$$

with $\eta > 0$ as a design parameter. Consider the update laws

$$\dot{\theta}_\alpha(t) = -Q_\alpha^{-1}\phi_\alpha(x)e_s \tag{30}$$

$$\dot{\theta}_\beta(t) = -Q_\beta^{-1}\phi_\beta(x)e_s u_{ce} \tag{31}$$

where Q_α and Q_β are positive definite and diagonal and serve as adaptation gains for the parameter updates. Note that the above adaptation laws do not guarantee that $\theta_\alpha \in \Omega_\alpha$ and $\theta_\beta \in \Omega_\beta$ so that we will use a projection method to ensure this, in particular, to make sure that $\beta_k(t) + \hat{\beta}(x,t) \geq \beta_0$. Additionally, the sliding mode control term is defined as

$$u_{si} = \frac{(W_\alpha(x) + W_\beta(x)|u_{ce}|)}{\beta_0} sgn(e_s) \tag{32}$$

To develop a stable adaptive controller for nonlinear time-varying systems, some assumptions about the form of the model and other technical conditions are necessary. Similar to linear time-varying systems [24], a common assumption with respect to nonlinear time-varying systems is to assume the boundedness of parameter variations, that is, we assume $\dot{\theta}_\alpha^*$ and $\dot{\theta}_\beta^*$ are bounded. Therefore, consider the nonlinear time-varying system (1) and (2) with strong relative degree d. Assume that (i) $\alpha_k(t)$ and $\beta_k(t)$ in (11) are bounded if x is bounded, (ii) $\beta_k(t) + \beta(x,t) \geq \beta_0$ for some known $\beta_0 > 0$, (iii) $|\omega_\alpha(x,t)| \leq W_\alpha(x)$ and $|\omega_\beta(x,t)| \leq W_\beta(x)$ with known $W_\alpha(x)$ and $W_\beta(x)$, (iv) $y_m(t), \dot{y}_m(t), \ldots, y_m^{(d)}(t)$ are measurable and bounded, (v) $x(t), y(t), \dot{y}(t), \ldots, y^{(d-1)}(t)$ are measurable, (vi) $1 \leq d < n$ with the zero dynamics uniformly exponentially attractive or $d = n$, and (vii) $\dot{\theta}_\alpha^*$ and $\dot{\theta}_\beta^*$ are bounded. Under these conditions there exist indirect adaptive control law (27, 28, 32) and update law (30, 31) such that all internal signals are uniformly bounded and the tracking error e is "small in the mean." Note that η in (29) is a design parameter and that by choosing it large we can obtain smaller mean. (Refer to [19] for the proof.)

3 Fault Diagnosis by Interfacing Multiple Models with a Supervisory Scheme

Note that in (11) $\alpha_k(t)$ and $\beta_k(t)$ may represent the known nominal model of system dynamics, while the on-line approximators may be used to learn the unknown dynamics, including modeling errors (or system changes due to time-varying parameters). During the initial stage where no fault occurs, the learning capabilities of the on-line approximator may be used to learn the modeling errors, thus enhancing the robustness properties. In addition, the adaptive control approach does not rely on explicit fault information so that there is no need to wait for fault information and it may tolerate a larger class of faults (including some that may not have been anticipated). However, since no fault information is explicitly used by the adaptive controller, this framework may be referred to as "passive" fault tolerant control, whose fault tolerance ability solely depends on the adaptation of parameters and it may take a long time to adapt to the proper parameters if the system dynamics are changed considerably due to faults. (Note that even small faults may cause drastic changes in system dynamics.) Although the adaptive scheme may work fine for small or incipient faults, for the faults that significantly change the system dynamics, the speed of adaption may be inadequate. Alternatively, we may use the time-varying functions $\alpha_k(t)$ and $\beta_k(t)$ to represent the basic system dynamics in fault-free or fault situations, which may be specified by the fault diagnosis scheme, so as to achieve an "active" fault tolerant control system.

To develop a fault diagnosis scheme, we use a "model-switching" idea to construct multiple models [25]. Note that the plant may run in many situations: no fault or different types of faults, possibly with manufacturing differences and deterioration. Comparing to the method which only uses a single model of the plant and discriminates between different faults by setting different thresholds for different residuals, the multiple model method may be preferable. The multiple model approach requires a bank of models to identify different situations by comparing the residuals generated these models. Assume that there are N possible fault situations. Using nonlinear system identification with Takagi-Sugeno fuzzy systems, we may have $N+1$ models $\{M_j\}_{j=0}^{N}$, where the nominal model M_0 corresponds to the situation where there is no fault, and the fault models M_j, $j = 1, 2, \ldots, N$ represent the j^{th} fault situations. By running a bank of models on-line, as shown in Figure 1, the residuals r_j can by generated by $r_j = y - \hat{y}_j$, where y is the output of the plant and $\hat{y}_j$ is the output of model M_j, (which we think of as an estimate of the plant output.)

The evaluation of the residual signals generated by the models is performed using an expert supervisory scheme. The heuristic knowledge of faults and processing experience can be incorporated into the expert system in the form of rules easily, and thus its advantages are the transparency of operation and simple integration of *a priori* knowledge. Basically, the rule-based expert supervisory system performs two functions:

1. **Determine the bank of models to generate the residuals:**

   Due to our understanding of the plant, we already have some knowledge of faults, i.e., the possible types of faults and the possible combinations of different faults. According to this, $N+1$ models have been established to describe the characteristics of these possible situations in which the plant may operate. Besides, we may know that there are some relationships among these fault situations. For instance, suppose fault i and fault j are two different types of faults, if fault i and only fault i is happening at present, then the next possible situation may be the existence of fault j together with fault i, or no

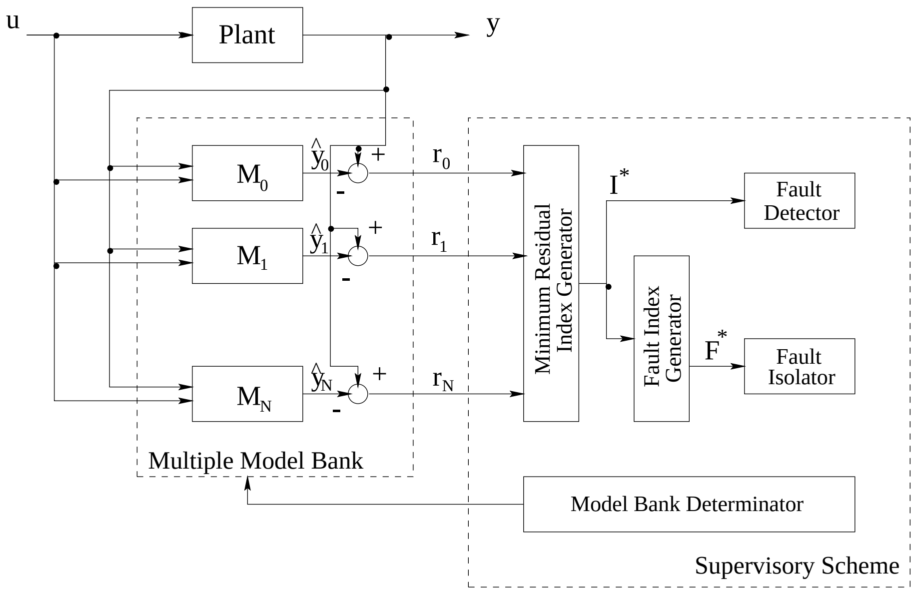

Figure 1: Structure of the fault diagnosis scheme.

fault, but not the situation where only fault j exists under the assumption that no two faults can happen or extinguish simultaneously. Therefore, at each instant only a subset of $N + 1$ situations may happen and thus only a subset of models may be required to generate residuals, which can be summarized as the rule: *IF (fault situation i is currently in existence) THEN (the subset S_i of models $\{M_j\}_{j=0}^{N}$ will be used as the model bank to generate residuals afterwards).* In particular, the rules may be

- *IF (there is no fault) THEN (the model bank consists of no fault model and all possible single fault models)*
- *IF (there is a single fault) THEN (all single fault models except the one that has been isolated will be removed from the model bank and the multiple fault models including the isolated fault and another possible fault will be added)*

The initial situation is usually assumed to be no fault. Using this type of knowledge the computation complexity can be reduced and the possibility of false alarms can be decreased as well.

2. **Fault diagnosis by decision logic:**

Once the residuals are generated, the fault decision logic is used to diagnose the faults. A sequence of minimum residual indices is first generated by

$$I^*(k) = arg\min(r_i(k)) \tag{33}$$

where $I^*(k)$ denotes the index of the model whose residual is minimum, that is, the most appropriate model to represent the plant at instant k. Note that when the plant situation changes, the input-output characteristics of the system will change as well. Therefore, the change of index, i.e., the change of most suitable model, may be used to indicate the change of plant situations, i.e., the occurrence of new faults. However, the new index after the change does not mean that the corresponding fault situation has occurred. This is because during this transient phase the residuals may change drastically and some of them may happen to be very small for a short time and become large later since they are not the proper fault situations. In order to isolate the fault situations correctly, some logic rules are used to guarantee that a fault will be isolated only if it lasts at least for T_0 seconds. The results of isolation can be represented by a fault index F^*, as shown in Figure 1. This strategy may affect fault sensitivity to some extent but it is effective to isolate faults accurately which is more important.

This active method is expected to be effective when large "jump" faults occur. The stable adaptive fuzzy/neural controller first serves as a robust baseline controller to maintain stability, while the fault diagnosis unit locates and identifies the fault. Once the fault has been identified, a supervision scheme is employed to reconfigure the control law by recalling the pre-computed models corresponding to different faults and using them to represent the known dynamics at that time. Note that although the stable adaptive controller is capable of producing a *small in the mean* tracking error, in practice, it may take a relatively long time and the performance of the closed loop system may be degraded. Instead, the major change of system dynamics may be represented by choosing the "correct" known dynamics ($\alpha_k(t)$ and $\beta_k(t)$) from a bank of multiple models using the fault diagnostic information, so that the adaptive controller may "jump" to a closer region and compensate for modeling errors afterwards.

Note that here we are considering a general fault tolerant control system, which is composed of three main parts: fault detection and identification, baseline controller, and supervision scheme. An fault detection and identification unit provides suitable diagnostic information about the faults; the baseline controller (e.g., adaptive controller) guarantees the system performance with or without faults; the supervision scheme utilizes the fault information to reconfigure the baseline controller.

4 Simulation Examples: Fault Tolerant Engine Control

4.1 Model Development for the XTE46 Engine

To study the effectiveness of the proposed adaptive control methods, we apply them to the component level model simulation of an aircraft jet engine (General Electric XTE46). The General Electric XTE46 engine is a simplified, unclassified version of the original IHPTET engine [26]. The component level engine cycle model of the XTE46 engine is a thermodynamic simulation package, where each engine component is simulated. The CLM executes one pass within the digital control's sampling time and thermodynamic states are assumed to be in equilibrium after each pass through the simulation. The operating condition of the engine is defined by the altitude (ALT), mach number (XM), difference of temperature (DTAMB), and throttle setting represented by power code (PC). The health of the engine is described by ten "quality parameters" which include the flows and efficiencies of the fan, the compressor, and turbines. The model has three state variables, including the fan rotor speed (XNL), the core rotor speed (XNH), and the temperature at combustor inlet (TMPC). There are six actuators, but the major control variables are the combustor fuel flow (WF36), the exhaust nozzle area (A8), and the variable area bypass injector area (A16). For simplicity, we assume that the fundamental dynamic characteristics of the CLM can be represented by a single-input single-output system in the form

$$\begin{aligned} \dot{x} &= f(x,u,c,p) && (34)\\ y &= h(x,u,c,p) && (35) \end{aligned}$$

where $x = [\text{XNL, XNH}]^\top$ is the state vector, $u = \text{WF36}$ is the input variable, $y = \text{XN2}$ is the output of the engine, $c = [\text{ALT, XM, DTAMB, PC}]^\top$ represents the operating condition of the engine, $p = [\text{ZSW2, SEDM2, ZSW7D, SEDM7D, ZSW27, SEDM27, ZSW41, ZSE41, ZSW49, ZSE49}]^\top$ represents the quality parameter vector, $f(\cdot)$ denotes the unknown function representing the nonlinear characteristics of the engine, and $h(\cdot) = \text{XNL}$ because the output variable XN2 is just the measurement of the state variable XNL.

To develop a computational model for the XTE46 engine, we performed nonlinear system identification to approximate local engine dynamics (specified by fixed values of operating conditions and quality parameters), followed by interpolating these local models to generate the "global" model (actually, it is a "regional" model valid in the "climb" region) [27, 25]. The general form of the model can be described as

$$\begin{aligned} \dot{x} &= f(x,c,p) + g(x,c,p)u && (36)\\ y &= x_1 && (37) \end{aligned}$$

where

$$f(x,c,p) = \frac{\sum_{i=1}^{N} \underline{f}(x,c_i,p_i)\mu_i(c,p)}{\sum_{i=1}^{N} \mu_i(c,p)} \tag{38}$$

$$g(x,c,p) = \frac{\sum_{i=1}^{N} \underline{g}(x,c_i,p_i)\mu_i(c,p)}{\sum_{i=1}^{N} \mu_i(c,p)} \tag{39}$$

$$\underline{f}(x,c_i,p_i) = \frac{\sum_{j=1}^{R} [a_{j,0}(c_i,p_i) + a_{j,1}(c_i,p_i)x_1 + a_{j,2}(c_i,p_i)x_2]\underline{\mu}_j(x_1)}{\sum_{j=1}^{R} \underline{\mu}_j(x_1)} \tag{40}$$

$$\underline{g}(x,c_i,p_i) = \frac{\sum_{j=1}^{R} a_{j,3}(c_i,p_i)\underline{\mu}_j(x_1)}{\sum_{j=1}^{R} \underline{\mu}_j(x_1)} \tag{41}$$

where $u = \text{WF36}$ and $x = [x_1, x_2]^\top = [\widehat{\text{XNL}}, \widehat{\text{XNH}}]^\top$ which is positive since the fan speed and the core speed cannot be negative and $x \in S_x$ (a valid speed region). The value of c is the known operating condition vector, p is unknown quality parameter vector, and c_i, p_i are specified nodes where we establish node models. Also, $f = [f_1, f_2]^\top$, $g = [g_1, g_2]^\top$, $\underline{f} = [\underline{f}_1, \underline{f}_2]^\top$, and $\underline{g} = [\underline{g}_1, \underline{g}_2]^\top$ are 2×1 function vectors, $a_{j,0}, a_{j,1}, a_{j,2}, a_{j,3}$ are 2×1 vectors of linear parameters of Takagi-Sugeno fuzzy systems, $\mu_i(c,p)$ are membership functions of fuzzy interpolation between different operating conditions and quality parameters, and $\underline{\mu}_j(x_1)$ are membership functions describing the nonlinearity with respect to x_1. By inspecting the parameters that result from the identification process we found that $a^1_{j,3}(c_i, p_i) > a^2_{j,3}(c_i, p_i) > 0$ and $a^2_{j,2}(c_i, p_i) < a^1_{j,2}(c_i, p_i) < 0$ for any $i = 1, 2, \ldots, N, j = 1, 2, \ldots, R$. Thus, we know that the "relative degree" of the engine is 1 and the engine zero dynamics are uniformly exponentially attractive [27].

4.2 Fault Tolerant Engine Control

Consider the engine in the form of

$$\begin{aligned} \dot{y} &= f_1(x, c, p) + g_1(x, c, p)u & (42) \\ &= [\alpha_k(c, p_0, t) + \alpha(x, c, p)] + [\beta_k(c, p_0, t) + \beta(x, c, p)]u & (43) \end{aligned}$$

where $x(t)$ and $y(t)$ are measurable according to the properties of the component level engine model. By studying dynamics of the developed nonlinear model we know that $g_1(x, c, p) > 0.32$ so that we can set $\beta_0 = 0.32$. We use our developed engine model to represent the nominal model dynamics $\alpha_k(c, p_0, t)$ and $\beta_k(c, p_0, t)$ by setting the quality parameters to be the nominal value p_0, and they are bounded if x is bounded since the model is in the form of a Takagi-Sugeno fuzzy system. The unknown dynamics $\alpha(x, c, p)$ and $\beta(x, c, p)$ describe both the model uncertainty caused by nominal model inaccuracy and system changes (time-varying characteristics) due to fault effects. They will be approximated by two radial basis function networks $\hat{\alpha}$ and $\hat{\beta}$ with 11 receptive field units for each. The inputs to the neural networks include two state variables and the parameters are updated on-line to capture the unknown time-varying dynamics affected by model inaccuracy and faults so that fault tolerance can be achieved. Note that the stable adaptive controller will ensure the stability of x_1, and the uniform exponential attractivity of the engine zero dynamics will ensure the stability of the unobservable state x_2. Since the relative degree of the system is 1, the error dynamics are simple ($e_s(t) = e(t)$ and $\bar{e}_s(t) = 0$). Taking into account of the engine dynamics, the model uncertainty is described by $W_\alpha = 0.01$ and $W_\beta = 0.01$. Note that we cannot explicitly know the model uncertainty so that the parameters W_α and W_β are treated as design parameters and tuned by trial and error to achieve good control performance. In addition, the adaptation gains are tuned to be $Q_\alpha^{-1} = 5e - 8$ and $Q_\beta^{-1} = 1e - 17$, and the design parameter $\eta = 1$. The reference trajectory is defined by passing a reference signal through a linear reference model $\frac{Y_m(s)}{R(s)} = \frac{3}{s+3}$ so that $y_m(t)$ and $\dot{y}_m(t)$ are measurable and bounded. Note that the time-varying dynamics caused by both incipient faults and jump-like faults satisfy the assumptions on the bounded parameter variations and bounded parameter rate of change, so that we could apply the stable adaptive control method developed above to solve the fault tolerant engine control problem.

We let the component level engine model run at the operating condition of ALT = 15000, XM = 0.7, DTAMB = 0, and PC = 46. For quality parameters of the engine, we set the initial engine variation to be $p_0 = [0.1\%, 0.1\%, 0.2\%, 0.1\%, -0.1\%, 0, -0.3\%, 0.2\%, -0.1\%, 0.1\%]$, and the engine deterioration index to be 0.1. Figure 2 shows the performance of fault tolerant control for a fault scenario, where an abrupt large compressor hub fault occurs at $T_f = 6$ sec. The occurrence of abrupt large compressor hub fault affects the system performance significantly.

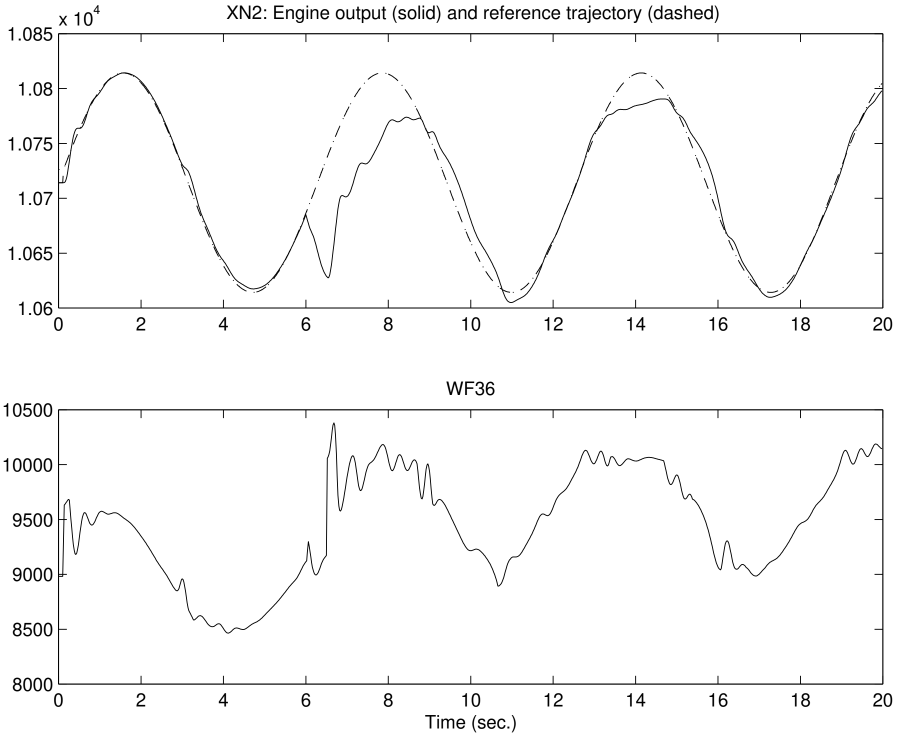

Figure 2: Performance of the integrated fault tolerant controller.

However, the fault tolerant controller accommodates for the faults quite well by identifying the fault type with the fault diagnosis unit and then modifying the control action by switching to the corresponding model.

The operation of the fault diagnosis unit is shown in Figure 3, where we use three models: the nominal model (whose residual is indicated by the solid line), the large fan fault model (whose residual is indicated by the dotted line), and the large compressor hub fault model (whose residual is indicated by the dashed line). As shown in Figure 3, after the original oscillations, which are absorbed by the isolation function in the supervisory scheme (where $T_0 = 0.5$ seconds), the fault is isolated at 6.5 seconds (where Index 0 represents the nominal model, Index 3 represents the large fan fault model, and Index 6 represents the large compressor hub fault model). Note that the on-line model bank is only composed of models corresponding to large faults but not small or medium faults because for those cases the adaptive controller itself can learn the fault effects quickly so that there is no need for model switching guided by the fault diagnosis unit. The effectiveness of integrating the fault diagnosis unit with the adaptive controller can be further demonstrated by comparing its performance with that of a fault tolerant controller with only an adaptive scheme but not fault diagnosis unit. As shown in Figure 4, although it does achieve fault accommodation abilities, it takes much longer time to do so. It is also worthy to mention that this integrated fault tolerance scheme is effective for large jump faults, while for incipient faults or small jump faults the adaptive scheme is capable of learning the fault effects relatively quickly and maintain good performance. Thus, although the fault diagnosis unit still provides the fault information, this will result in model switching.

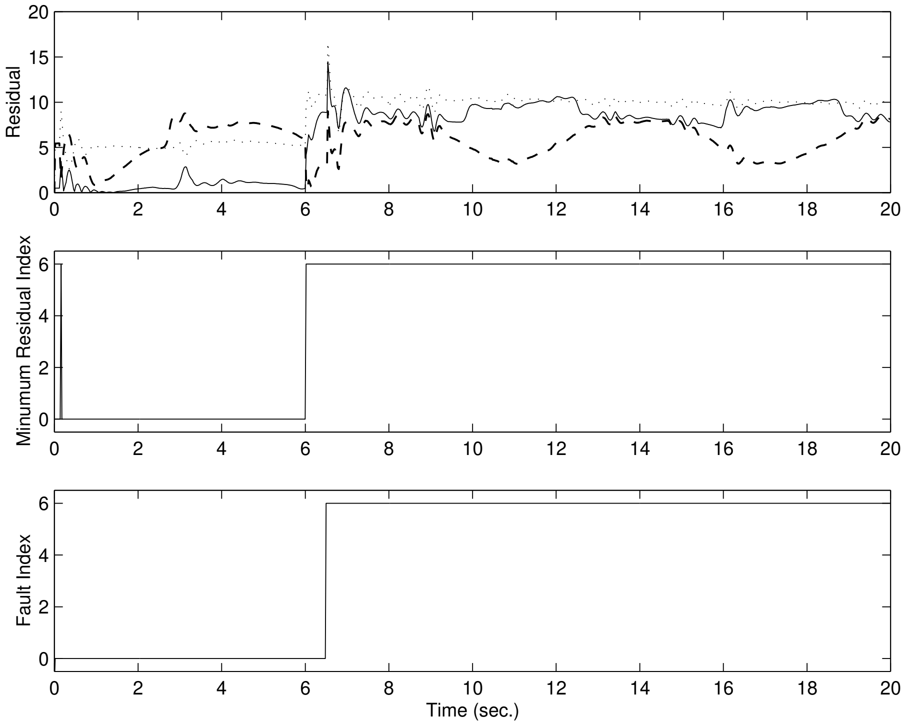

Figure 3: Operation of the fault diagnosis unit.

Once the fault is identified, the known dynamics $\alpha_k(t)$ and $\beta_k(t)$ are "switched" correspondingly from the nominal model to the large compressor hub fault model, and then the performance is further improved by using the adaptation scheme to compensate for model uncertainties. Indeed, the adaptive controller plays the major role in achieving system stability and robustness in the presence of uncertainties such as engine-to-engine manufacturing differences, engine deterioration during normal operation, and the occurrence of faults. Here, we examine an engine with large initial engine variation ($p_0 = [0.3\%, 0.5\%, 0.2\%, 0.5\%, 0.4\%, 0.3\%, -0.3\%, 0.5\%, -0.5\%, 0.5\%]$) and severe engine deterioration (the engine deterioration index to is 0.7), and still let the component level engine model run at the same operating condition (ALT $=$ 15000, XM $=$ 0.7, DTAMB $=$ 0, and PC $=$ 46) with same fault scenario (where an abrupt large compressor hub fault occurs at $T_f = 6$ sec). As shown in Figure 5, the fault tolerant controller presents strong robustness.

5 Conclusion

Fault tolerant control for nonlinear time-varying systems is of both theoretical and practical significance. In this paper we present a intelligent fault tolerant control methodology using an on-line approximation based adaptive controller to guarantee stability and robustness and an fault diagnosis unit to provide fault information and then reconfigure the control law. Simulation examples for a fault tolerant engine control problem are also given to demonstrate the effectiveness of the proposed scheme.

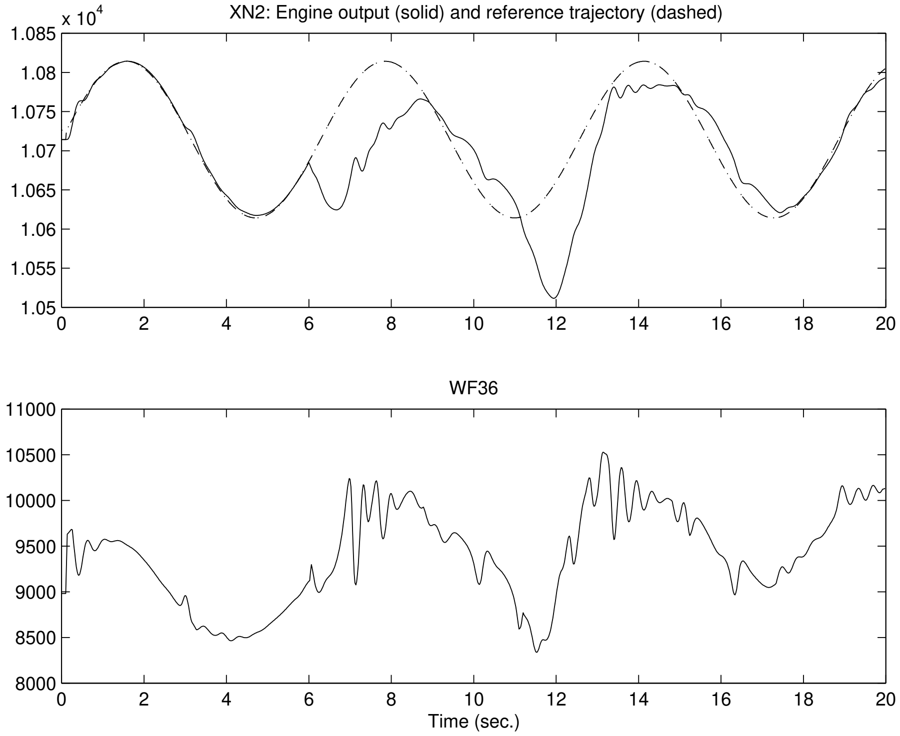

Figure 4: Performance of the adaptive controller without fault diagnosis unit.

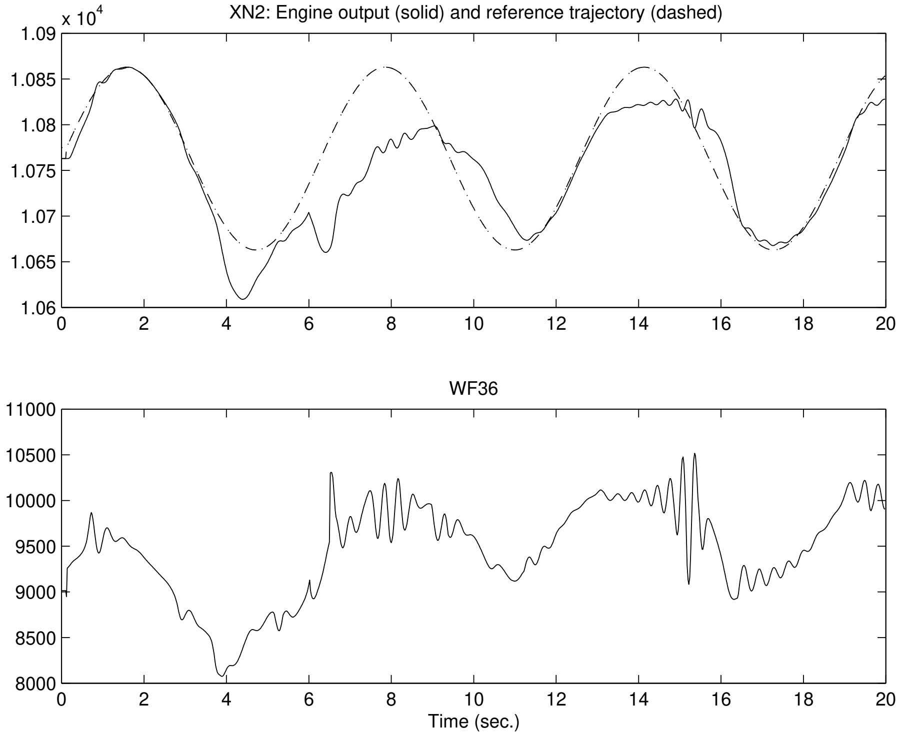

Figure 5: Performance of the fault tolerant controller for a poor engine.

References

[1] P. M. Frank, "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy - a survey and some new results," *Automatica*, vol. 26, pp. 459–474, 1990.

[2] R. F. Stengel, "Intelligent failure-tolerant control," *IEEE Control Systems Magazine*, vol. 11, pp. 14–23, June 1991.

[3] R. J. Patton, "Fault-tolerant control: The 1997 situation," in *IFAC Fault Detection, Supervision and Safety for Technical Processes, Kingston Upon Hull, UK*, pp. 1029–1051, 1997.

[4] R. Veillette, J. Medani, and W. Perkins, "Design of reliable control systems," *IEEE Transactions on Automatic Control*, vol. 37, no. 3, pp. 290–304, 1992.

[5] Z. Chen, R. Patton, and J. Chen, "Robust fault-tolerant system synthesis via LMI," in *IFAC Fault Detection, Supervision and Safety for Technical Processes, Kingston Upon Hull, UK*, pp. 337–342, 1997.

[6] D. Moerder, N. Halyo, J. Broussard, and A. Caglayan, "Application of precomputed control laws in a reconfigurable aircraft flight control system," *Journal of Guidance, Control and Dynamics*, vol. 12, no. 3, pp. 325–333, 1989.

[7] C. Aubrun, H. Noura, and D. Sauter, "On line reconfiguration strategy of control law for degraded systems," in *IFAC Fault Detection, Supervision and Safety for Technical Processes, Espoo, Finland*, pp. 104–107, 1994.

[8] D. Looze, J. Weiss, J. Eterno, and N. Barrett, "An automatic redesign approach for restructurable control systems," *IEEE Control Systems Magazine*, pp. 16–22, 1985.

[9] C. Huang and R. Stengel, "Restructurable control using proportional-integral implicit model-following," *Journal of Guidance*, vol. 13, no. 2, pp. 303–309, 1990.

[10] Z. Gao and P. Antsaklis, "Stability of the pseudo-inverse method for reconfigurable control systems," *International Journal of Control*, vol. 53, no. 3, pp. 717–729, 1991.

[11] W. Morse and K. Ossman, "Model-following reconfigurable flight control system for the AFTI/F-16," *Journal of Guidance*, vol. 13, no. 6, pp. 969–976, 1990.

[12] M. Bodson and J. Groszkiewicz, "Multivariable adaptive algorithms for reconfigurable flight control," *IEEE Transactions on Control Systems Technology*, vol. 5, no. 2, pp. 217–229, 1997.

[13] H. Noura, D. Sauter, and F. Hamelin, "Evaluation of a fault-tolerant control design for actuator faults," in *Proceedings on the 36th IEEE Conference on Decision and Control, San Diego, CA*, pp. 3982–3986, 1997.

[14] W. A. Kwong, K. M. Passino, E. G. Laukonen, and S. Yurkovich, "Expert supervision of fuzzy learning systems for fault tolerant aircraft control," *Proc. of the IEEE, Special Issue on Fuzzy Logic in Engineering Applications*, vol. 83, pp. 466–483, Mar. 1995.

[15] K. Narendra and S. Mukhopadhyay, "Intelligent control using neural networks," *IEEE Control Systems Magazine*, vol. 12, no. 2, pp. 11–18, 1992.

[16] J. Boskovic and R. Mehra, "A multiple model-based reconfigurable flight control system design," in *Proceedings on the 37th IEEE Conference on Decision and Control, Tampa, FL*, pp. 4503–4508, 1998.

[17] C. Chiang and H. Youssef, "Neural network and fuzzy logic approach to aircraft reconfigurable control design," in *Proceedings of the American Control Conference, Seattle, Washington*, pp. 3505–3509, 1995.

[18] J. Farrell, T. Berger, and B. Appleby, "Using learning techniques to accommodate unanticipated faults," *IEEE Control Systems Magazine*, vol. 13, no. 3, pp. 40–49, 1993.

[19] Y. Diao and K. M. Passino, "Stable adaptive control of feedback linearizable time-varying nonlinear systems with application to fault tolerant engine control," *Submitted to International Journal of Control*, 2000.

[20] A. Isidori, *Nonlinear Control Systems.* Springer-Verlag, 1995.

[21] M. Polycarpou, "Fault accommodation of a class of nonlinear dynamical systems using a learning approach," in *Proceedings of the 1999 IEEE International Symposium on Intelligent Control/Intelligent Systems and Semiotics*, (Cambridge, MA), pp. 185–190, Sept. 1999.

[22] H. K. Khalil, *Nonlinear Systems.* New York, NY: MacMillan, Inc., 1996.

[23] S. Palanki and C. Kravaris, "Controller synthesis for time-varying systems by input/output linearization," *Computer & Chemical Engineering*, vol. 21, no. 8, pp. 891–903, 1997.

[24] K. Tsakalis and P. Ioannou, *Linear Time-Varying Systems: Control and Adaptation.* Englewood Cliffs, New Jersey: Prentice Hall, 1993.

[25] Y. Diao and K. M. Passino, "Fault diagnosis for a turbine engine," in *Proceedings of the American Control Conference, Chicago, Illinois*, pp. 2393–2397, 2000.

[26] S. Adibhatla and T. Lewis, "Model-based intelligent digital engine control," in *AIAA-97-3192, 33rd Joint Propulsion Conference*, July 1997.

[27] Y. Diao and K. M. Passino, "Fault tolerant stable adaptive fuzzy/neural control for a turbine engine," *to appear in IEEE Transactions on Control Systems Technology*, 2001.