Submitted to *IEEE Transactions on Control Systems Technology* (January 2000)

Fault Tolerant Stable Adaptive Fuzzy/Neural Control for a Turbine Engine*

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 jet engines is complicated by the presence of engine-to-engine manufacturing differences and engine deterioration during normal operation, the complexity of an accurate engine model, and our inability to directly measure certain engine variables. Here, we work with a sophisticated component level model (CLM) simulation of a turbine engine (the General Electric XTE46) that can simulate the effects of manufacturing and deterioration differences, in addition to a variety of faults. To develop a fault tolerant control system we begin by using the CLM to generate data that is used by a Levenberg-Marquardt method to train a Takagi-Sugeno fuzzy system to represent the engine. The resulting nonlinear model provides a reasonably accurate representation of manufacturing differences, engine deterioration, and fault effects. Stable indirect and direct adaptive controllers are applied to achieve fault tolerant engine control by using Takagi-Sugeno fuzzy systems to "learn" the unknown dynamics caused by faults and to accommodate faults by updating the controller, and this nonlinear model provides *a priori* knowledge about the nominal engine dynamics. We prove that both adaptive schemes achieve asymptotic tracking. The performance of the fault tolerant indirect and direct adaptive controllers is demonstrated through the component level model simulation of the XTE46 engine.

1 Introduction

Motivated by the growing need for high levels of system performance and reliability in the presence of unexpected changes of system functions, fault tolerant control is receiving increasing attention. Survey papers by Patton [1] and Stengel [2] present excellent overviews of advances in the research of fault tolerant control. Due to the complexity and importance of jet engines, the study of fault diagnosis and fault accommodation for jet engines has become a very active research topic for both theoretical and practical reasons. In [3, 4, 5] the authors studied sensor failure detection for jet engines using a Kalman filter with a Generalized Likelihood Ratio Testing based scheme. In [6, 7] the authors derived linearized models of jet engine systems via the

*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).

α-canonical form parameterization identification method and applied a parameter estimation approach in fault detection and isolation (FDI) for the space shuttle main engine. In [8, 9, 10] the authors studied fault detection of jet engine sensor systems using an eigenstructure assignment technique to design observer based residual generators, and they also studied its robustness. The related reconfigurable flight control problem has been studied for many years and the focus there is to safely recover from structural damage or system faults. In [11] the authors detected sensor and actuator faults of flight control systems using multiple model adaptive estimation and in [12, 13] multiple model adaptive control method was used in reconfigurable flight control. In [14, 15] model following methods were used for a reconfigurable flight control system. In [16] the authors used feedback linearization for restructurable flight control.

Recently, artificial intelligence has received increasing attention and been applied in the field of nonlinear fault diagnosis and fault tolerant control (e.g., see the discussion in [17]). The artificial intelligent methods such as fuzzy systems, neural networks and expert systems have the potential to "learn" the plant model from input-output data or "learn" fault knowledge from past experience, and they can be used as function approximators to construct the analytical model for residual generation, or as supervisory schemes to make the fault analysis decisions [18]. The nonlinear modeling ability of neural networks has been utilized for nonlinear FDI problems [19, 20, 21, 22, 23, 24]. In addition, the learning ability has also been studied and successfully used in nonlinear robust FDI [25, 26, 27, 28, 29, 30]. Meanwhile, expert systems and fuzzy logic have been used in model based FDI [31, 32, 33] and in [34] the "fuzzy model reference learning controller", supervised by an expert system strategy, was used to compensate for actuator failures in an F-16 aircraft. Additionally, genetic algorithms have applied to fault estimation [35].

This paper presents fault tolerant engine control methods achieved by stable indirect and direct adaptive controllers. The analytical models of the engine, which are in the form of Takagi-Sugeno fuzzy systems, have been developed using a nonlinear system identification methodology (Section 2). In Section 3 the stable adaptive indirect and direct controllers are developed and the stability is studied. In Section 4 several component level model simulations are conducted to illustrate the effectiveness of proposed fault tolerant engine control method.

2 Model Development Using Takagi-Sugeno Fuzzy Systems

2.1 The XTE46 Turbine Engine

The General Electric XTE46 engine, as shown in Figure 1, is a simplified, unclassified version of the original IHPTET engine [36]. To develop a fault tolerant engine controller an accurate representation of the engine dynamics is desired. This model may be used as the "truth model" (or "design model") in the control simulation to represent the real engine, or may serve as the nominal model (known dynamics) in a "model-based" control strategy. However, modeling a turbine engine is undoubtedly a very difficult problem in the fact that the jet engine system has an iterative structure, which means that the model cannot be written down in a differential-algebraic equation form. Fortunately, a thermodynamic simulation package, the component level engine

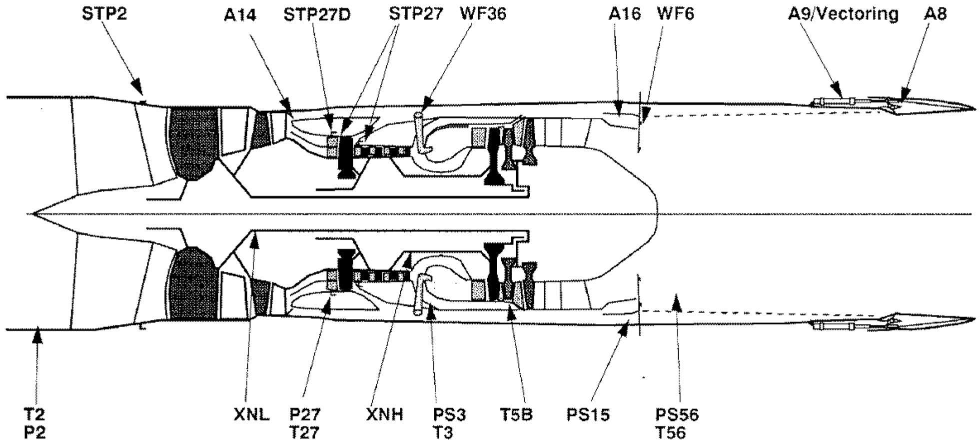

Figure 1: Schematic of the XTE46 turbine engine.

cycle model (CLM) of the XTE46 engine, was provided by General Electric Aircraft Engines (GEAE). This is a sophisticated highly nonlinear dynamic model 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 CLM is a low frequency transient turbofan engine simulation and volume dynamics and airflow storage affects, which are high frequency phenomenon, are not included. The operating condition of the engine is defined by the altitude (ALT), the mach number (XM), the difference of temperature (DTAMB) and the 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).

From the point of view of theoretical studies in fault tolerant control, the component level model is too complicated. Although the CLM does provide a driver to trim the model to specified operating conditions and generate linearized models, studies show that the accuracy of the linear models is not adequate for our purposes (where we consider faults with significant nonlinear effects). Here, we developed a nonlinear model with Takagi-Sugeno fuzzy systems using a system identification methodology that utilized nonlinear transient data generated by the CLM. This model will be used for theoretical studies (i.e., building the stable adaptive controller), for representing the dynamics of nominal engine (as "known" system dynamics in the indirect adaptive controller), and for preliminary simulation studies (i.e., it serves as the truth model of the engine).

2.2 The Takagi-Sugeno Fuzzy System

Developing mathematical models for nonlinear systems can be quite challenging. Takagi-Sugeno fuzzy systems are capable of serving as the analytical model for nonlinear systems due to the universal approximation property, that is, any desired approximation accuracy can be achieved by increasing the size of the approximation structure and properly defining the parameters of the approximator [37]. A Takagi-Sugeno fuzzy system can be defined by

$$y = F_{ts}(x,\theta) = \frac{\sum_{i=1}^{R} g_i(x)\mu_i(x)}{\sum_{i=1}^{R}\mu_i(x)} \tag{1}$$

$$g_i(x) = a_{i,0} + a_{i,1}x_1 + \cdots + a_{i,n}x_n \tag{2}$$

$$\mu_i(x) = \prod_{j=1}^{n} \exp\left(-\frac{1}{2}\left(\frac{x_j - c_j^i}{\sigma_j^i}\right)^2\right) \tag{3}$$

where y is the output of the fuzzy system, $x = [x_1, x_2, \ldots, x_n]^\top$ holds the n inputs, and $i = 1, 2, \ldots, R$ represent R different rules. The shapes of the membership functions are chosen to be Gaussian, and center-average defuzzification and product are used for the premise and implication in the structure of the fuzzy system. The $g_i(x)$, $i = 1, 2, \ldots, R$ are called consequent functions of the fuzzy system, where the $a_{i,j}$ are constants. The premise membership functions $\mu_i(x)$ are assumed to be well defined so that $\Sigma_{i=1}^{R}\mu_i(x) \neq 0$ for all x. The parameters that enter in a nonlinear fashion are c_j^i and σ_j^i, which are the centers and relative widths of the membership functions, respectively, for the j^{th} inputs and the i^{th} rules. Actually, the Takagi-Sugeno fuzzy system, where the consequent parts are chosen to be affine functions, is a special case of "functional fuzzy systems" [37]. If other functions such as polynomial or neural networks are used as consequent functions, different kinds of functional fuzzy systems will be generated.

The tunable parameter vector θ in (1) can be composed of both premise membership function parameters (c_j^i and σ_j^i) and consequent function parameters ($a_{i,j}$). This is referred to as *nonlinear in the parameter* case. A *nonlinear in the parameter* Takagi-Sugeno fuzzy system can be tuned by a variety of gradient methods such as the steepest descent method and Levenberg-Marquardt method. Alternatively, the parameter vector θ may be composed of only the consequent function parameters so that y is a linear function of θ. To tune the fuzzy approximator for this *linear in the parameter* case, a linear least square method will normally be suitable.

2.3 Fuzzy Modeling for the XTE46 Engine

The CLM for the XTE46 aircraft engine is a complicated multiple-input multiple-output nonlinear system (involving schedules, look-up tables, and partial differential equations). We assume that its fundamental dynamic characteristics, however, can be represented by a single-input single-output system in the form

$$\dot{x} = f(x, u, c, p) \tag{4}$$

$$y = h(x, u, c, p) \tag{5}$$

where $x = [\text{XNL, XNH, TMPC}]^\top$ 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 the measurement of the state variable XNL.

The analytical model of the engine is developed in two steps. Fuzzy identification is applied first to generate (a grid of) "node" models specified by operating conditions and quality parameters. Afterwards, the "global" model can be constructed by fuzzy interpolation on these node models. (We use this two-step method rather than identifying the model directly from the data collected from the whole space of operating conditions and quality parameters in that it is practically impossible to train an approximator using such a large amount of data, due to the limitations of computing resources.) Given a specific operating condition (c_i) and fixed values of quality parameters (p_i), the node model for the XTE46 engine can be obtained using nonlinear identification technique as

$$\begin{aligned}
\widehat{\text{XNLdot}}(k) &= F_{ts}^1(\widehat{\text{XNL}}(k), \widehat{\text{XNH}}(k), \text{WF36}(k), \theta^1(c_i, p_i)) && (6)\\
\widehat{\text{XNHdot}}(k) &= F_{ts}^2(\widehat{\text{XNL}}(k), \widehat{\text{XNH}}(k), \text{WF36}(k), \theta^2(c_i, p_i)) && (7)\\
\widehat{\text{XNL}}(k+1) &= \widehat{\text{XNL}}(k) + T_s\widehat{\text{XNLdot}}(k) && (8)\\
\widehat{\text{XNH}}(k+1) &= \widehat{\text{XNH}}(k) + T_s\widehat{\text{XNHdot}}(k) && (9)\\
\widehat{\text{XN2}}(k+1) &= \widehat{\text{XNL}}(k+1) && (10)
\end{aligned}$$

where two multiple-input single-output (MISO) Takagi-Sugeno fuzzy systems, F_{ts}^1 and F_{ts}^2, are specified with corresponding parameters θ^1 and θ^2, respectively. The variables $\widehat{\text{XNL}}(k)$ and $\widehat{\text{XNH}}(k)$ denote the estimated values of $\text{XNL}(k)$ and $\text{XNH}(k)$ ($\widehat{\text{XNL}}(0) = \text{XNL}(0)$ and $\widehat{\text{XNH}}(0) = \text{XNH}(0)$ to let the fuzzy model have the same initial values as the engine,) and $\widehat{\text{XN2}}(k)$ is the estimated value of $\text{XN2}(k)$. $\widehat{\text{XNLdot}}(k)$ and $\widehat{\text{XNHdot}}(k)$ are the outputs of the fuzzy systems. T_s is the sample time, which is 0.02 seconds. The fuzzy systems are trained using engine data generated by the transient driver of the component level model (CLM) simulator of the XTE46 engine. One thousand engine input-output data pairs are collected which reflect the transient performance of the engine for 20 seconds (sampled every 0.02 seconds) at a specific "node" (of operating conditions and quality parameters). For the k^{th} experimental data pair, the input data are the state variables $\text{XNL}(k)$ and $\text{XNH}(k)$, and the input variable $\text{WF36}(k)$. The output data are $\text{XNLdot}(k)$ and $\text{XNHdot}(k)$, which denote the derivatives of $\text{XNL}(k)$ and $\text{XNH}(k)$, respectively.

The structure of fuzzy approximators is specifically selected to satisfy the requirement of stable adaptive control (it should be affine, i.e., in the form of $\dot{x} = f(x) + g(x)u$, for the sake of feedback linearization) and to simplify the stability analysis of the system "zero dynamics". In particular, the premise input is chosen to be $\widehat{\text{XNL}}(k)$ only so that the Takagi-Sugeno fuzzy system can be written in the affine form as

$$
\begin{aligned}
& F_{ts}(\widehat{\mathrm{XNL}}(k), \widehat{\mathrm{XNH}}(k), \mathrm{WF36}(k), \theta) \\
= & \frac{\sum_{j=1}^{R}(a_{j,0} + a_{j,1}\widehat{\mathrm{XNL}}(k) + a_{j,2}\widehat{\mathrm{XNH}}(k) + a_{j,3}\mathrm{WF36}(k))\mu_j(\widehat{\mathrm{XNL}}(k))}{\sum_{j=1}^{R}\mu_j(\widehat{\mathrm{XNL}}(k))} \\
= & \frac{\sum_{j=1}^{R}(a_{j,0} + a_{j,1}\widehat{\mathrm{XNL}}(k) + a_{j,2}\widehat{\mathrm{XNH}}(k))\mu_j(\widehat{\mathrm{XNL}}(k))}{\sum_{j=1}^{R}\mu_j(\widehat{\mathrm{XNL}}(k))} \\
& + \frac{\sum_{j=1}^{R} a_{j,3}\mu_j(\widehat{\mathrm{XNL}}(k))}{\sum_{j=1}^{R}\mu_j(\widehat{\mathrm{XNL}}(k))}\mathrm{WF36}(k)
\end{aligned}
$$

Using trial and error, fuzzy models with $R = 3$ rules each were selected and tuned by a Levenberg-Marquardt method for each node.

Notice that we did not use TMPC as one of the model inputs because it is not measurable and the importance of TMPC to the model accuracy is trivial, which is verified by an input selection method referred to as "regressor analysis" (where the regression models are constructed and the regression coefficients are analyzed to determine the importance of each input), so that it is not necessary to estimate TMPC either. Also note that the fuzzy models are running in an "open loop manner," that is, the outputs of the models will be fed back into the models as the inputs, and only the basic behavior of the engine may be obtained due to the drifting effects caused by the accumulation of approximation errors. We use this approach because sometimes we want this analytical model to be running as the "truth engine" in the simulation study, where the CLM and thus the engine states XNL and XNH are not available. Of course, when we utilize the model (e.g., as the known dynamics of the nominal engine in our fault tolerant controller) to control the CLM, we may use the (measurable) real engine states to be the inputs of fuzzy systems and may expect an improvement in model accuracy.

By nonlinearly interpolating between a grid of node models obtained above from nonlinear system identification, the global model can then be constructed which is in the form

$$
\widehat{\mathrm{XNLdot}}(k) = \frac{\sum_{i=1}^{N} F_{ts}^{1}(\widehat{\mathrm{XNL}}(k), \widehat{\mathrm{XNH}}(k), \mathrm{WF36}(k), \theta^1(c_i, p_i))\mu_i(z)}{\sum_{i=1}^{N}\mu_i(z)} \tag{11}
$$

$$
\widehat{\mathrm{XNHdot}}(k) = \frac{\sum_{i=1}^{N} F_{ts}^{2}(\widehat{\mathrm{XNL}}(k), \widehat{\mathrm{XNH}}(k), \mathrm{WF36}(k), \theta^2(c_i, p_i))\mu_i(z)}{\Sigma_{i=1}^{N}\mu_i(z)} \tag{12}
$$

$$
\mu_i(z) = \prod_{j=1}^{14} \exp\left(-\frac{1}{2}\left(\frac{z_j - c_j^i}{\sigma_j^i}\right)^2\right) \tag{13}
$$

where $i = 1, 2, \ldots, N$ represent N different models and $z = [c^\top, p^\top]^\top$ is the premise input vector including 14 variables.

We choose to focus on fault tolerant controller development for the "climb" region (which is defined as ALT $\in$ [12500, 17500], XM $\in$ [0.6, 0.8], DTAMB $\in$ [-35, 35], and PC $\in$ [45, 50]). We partition each operating condition variable into three regions to define our grid. In this way, we have $3^4 = 81$ models to describe the nonlinearity presented in the climb region. The values of quality parameters are composed of four parts: the nominal value, the initial engine variation due to manufacturing differences, the quality parameter adjustment resulted from engine deterioration, and the quality parameter change due to the faults. Note that the effects of engine deterioration and faults are larger than the initial engine variation, and we will try to capture the characteristics of these two factors and leave the effect of initial engine variation to be model uncertainty. By assuming that engine deterioration affects ten quality parameters in the same way (so that we may have three grids to represent no deterioration, half deterioration, and full deterioration, respectively) and considering four sizes of faults (no fault, small, medium, or large fault), we have $3 \times 4 \times 4 = 48$ models to describe the nonlinearity presented in the quality parameters (for simplicity, we only consider two possible faults, fan fault and compressor hub fault, here.) In total, we have $81 \times 48 = 3888$ (nonlinear) node models to describe the nonlinearity in the climb region. We need this level of complexity to obtain a reasonably accurate "design model" for the development of our controller.

The general form of the model can be described as

$$\dot{x} = f(x, c, p) + g(x, c, p)u \tag{14}$$

where

$$f(x, c, p) = \frac{\sum_{i=1}^{N} \underline{f}(x, c_i, p_i)\mu_i(c, p)}{\sum_{i=1}^{N} \mu_i(c, p)} \tag{15}$$

$$g(x, c, p) = \frac{\sum_{i=1}^{N} \underline{g}(x, c_i, p_i)\mu_i(c, p)}{\sum_{i=1}^{N} \mu_i(c, p)} \tag{16}$$

$$\underline{f}(x, c_i, p_i) = \frac{\sum_{j=1}^{R}[a_{j,0}(c_i, p_i) + a_{j,1}(c_i, p_i)x_1 + a_{j,2}(c_i, p_i)x_2]\underline{\mu}_j(x_1)}{\sum_{j=1}^{R} \underline{\mu}_j(x_1)} \tag{17}$$

$$\underline{g}(x, c_i, p_i) = \frac{\sum_{j=1}^{R} a_{j,3}(c_i, p_i)\underline{\mu}_j(x_1)}{\sum_{j=1}^{R} \underline{\mu}_j(x_1)} \tag{18}$$

where $u = \mathrm{WF36}$ and $x = [x1, x2]^\top = [\widehat{\mathrm{XNL}}, \widehat{\mathrm{XNH}}]^\top$ 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, $f, g, \underline{f}, \underline{g}$ 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 $a_{j,3}^1(c_i, p_i) > a_{j,3}^2(c_i, p_i) > 0$ and $a_{j,2}^2(c_i, p_i) < a_{j,2}^1(c_i, p_i) < 0$ for any $i = 1, 2, \ldots, N, j = 1, 2, \ldots, R$. Thus, for instance, we can draw the conclusion that $g_1(x, c, p) > 0$

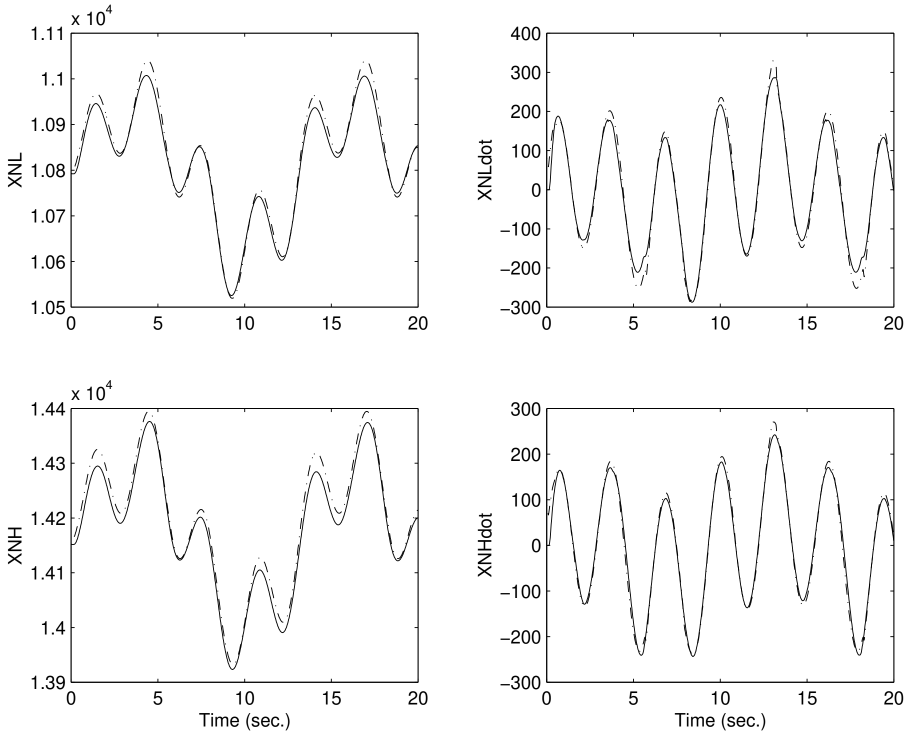

Figure 2: Illustrative example of the model performance.

for all x, c, p from this so that the "relative degree" of the engine can be determined (which we will discuss later). These relationships are important to study the stability of system.

The resulting nonlinear model provides a reasonably accurate representation of engine dynamics. Figure 2 gives an example (at a point different from the nodes where we generated the model, ALT = 16000, XM = 0.75, DTAMB = 0, PC = 46 and there are some initial engine variation, nearly half engine deterioration, and a fan fault a little bit larger than medium size), and compares the system response between the nonlinear model and the CLM, where the solid lines represent the system response of the CLM and the dashed lines represent that of the analytical model. We conducted many other such simulations to verify the quality of our design model; however, in the interest of brevity we do not include those plots here.

3 Fault Tolerant Stable Adaptive Fuzzy/Neural Control

In this section we will develop fault tolerant engine control using stable adaptive fuzzy/neural controllers [38]. Both indirect and direct adaptive control approaches are applied. We also show that under certain conditions asymptotic tracking of a reference signal and boundedness of all signals are achieved.

3.1 Indirect Adaptive Control

The general framework for modeling the nonlinear single-input single-output (SISO) system with various faults is described by the differential equation

$$\dot{x} = f(x) + g(x)u \tag{19}$$

$$y = h(x) \tag{20}$$

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, and functions $f(x)$, $g(x)$ and $h(x)$, which are smooth, represent both the nominal system dynamics and any change of system due to a fault.

Let $L_g^d h(x)$ be the d^{th} Lie derivative of $h(x)$ with respect to $g(x)$ (i.e., $L_g h(x) = \left(\frac{\partial h}{\partial x}\right)^\top g(x)$, $L_g^2 h(x) = L_g(L_g h(x)) = \left(\partial\left(\left(\frac{\partial h}{\partial x}\right)^\top g(x)\right)/\partial x\right)^\top g(x)$, etc.) A system is said to have "strong relative degree" d if $L_g h(x) = L_g L_f h(x) = \cdots = L_g L_f^{d-2} h(x) = 0$ and $L_g L_f^{d-1} h(x) \neq 0$ for all x. If system (19,20) has strong relative degree d, then

$$\begin{aligned}
\dot{\xi}_1 &= \xi_2 = L_f h(x) \\
\dot{\xi}_2 &= \xi_3 = L_f^2 h(x) \\
&\vdots \\
\dot{\xi}_{d-1} &= \xi_d = L_f^{d-1} h(x) \\
\dot{\xi}_d &= L_f^d h(x) + L_g L_f^{d-1} h(x) u
\end{aligned}$$

with $\xi_1 = y$, which, if we let $y^{(d)}$ denote the d^{th} derivative of y, may be rewritten as

$$y^{(d)} = [\alpha_k(t) + \alpha(x)] + [\beta_k(t) + \beta(x)]u \tag{21}$$

where $\alpha_k(t)$ and $\beta_k(t)$ are "known" dynamics of the system (e.g., the nominal model dynamics) that may depend on the states or known exogenous time dependent signals (which may represent the time profile of faults identified from the FDI scheme), and $\alpha(x)$ and $\beta(x)$ represent nonlinear dynamics of the plant that are unknown. Note that here we represent the relationship between known and unknown dynamics to be additive for the convenience of analysis but the actual relationship is not required to be additive. This is because no matter what kind of systems we consider and what part of dynamics is assumed to be known, the unknown dynamics (f_u) can always be represented as the difference between the whole system dynamics (f) and the known system dynamics (f_k), that is, $f_u = f - f_k$.

For the purpose of stability analysis we assume that for some $\beta_0 > 0$, we have $|\beta_k(t) + \beta(x)| \geq \beta_0$ so that it is always bounded away from zero (for convenience we assume that $\beta_k(t) + \beta(x) \geq 0$, however, the following analysis may easily be modified for systems which are defined with $\beta_k(t) + \beta(x) \leq 0$). We also assume that if x is bounded , then $\alpha_k(t)$ and $\beta_k(t)$ are bounded.

The on-line approximators are used to learn the unknown dynamics comprising modeling errors and system changes due to faults so as to achieve fault accommodation. We may choose radial basis function (RBF) neural networks, B-spline neural networks, Takagi-Sugeno fuzzy systems or other *linear in the parameter* approximators so that the approximations of $\alpha(x)$ and $\beta(x)$ of the actual system are

$$\hat{\alpha}(x) = \theta_\alpha^\top(t)\phi_\alpha(x) \tag{22}$$

$$\hat{\beta}(x) = \theta_\beta^\top(t)\phi_\beta(x) \tag{23}$$

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

$$\alpha(x) = \theta_\alpha^{*\top}\phi_\alpha(x) + \omega_\alpha(x) \tag{24}$$

$$\beta(x) = \theta_\beta^{*\top}\phi_\beta(x) + \omega_\beta(x) \tag{25}$$

where

$$\theta_\alpha^* = \arg\min_{\theta_\alpha \in \Omega_\alpha}\left(\sup_{x \in S_x} |\theta_\alpha^\top\phi_\alpha(x) - \alpha(x)|\right) \tag{26}$$

$$\theta_\beta^* = \arg\min_{\theta_\beta \in \Omega_\beta}\left(\sup_{x \in S_x} |\theta_\beta^\top\phi_\beta(x) - \beta(x)|\right) \tag{27}$$

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

$$|\omega_\alpha(x)| \leq W_\alpha(x) \tag{28}$$

$$|\omega_\beta(x)| \leq W_\beta(x) \tag{29}$$

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

$$\tilde{\theta}_\alpha(t) = \theta_\alpha(t) - \theta_\alpha^* \tag{30}$$

$$\tilde{\theta}_\beta(t) = \theta_\beta(t) - \theta_\beta^* \tag{31}$$

We view fault tolerant 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

$$\frac{Y_m(s)}{R(s)} = \frac{q(s)}{p(s)} = \frac{q_0}{s^d + p_{d-1}s^{d-1} + \cdots + p_0} \tag{32}$$

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

The indirect adaptive control law

$$u = u_{ce} + u_{si} \tag{33}$$

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 $e_s(t) = e^{(d-1)}(t) + k_{d-2}e^{(d-2)}(t) + \cdots + k_1\dot{e}(t) + k_0e(t)$, that is, in the frequency domain, $e_s(s) = L(s)e(s)$ with $L(s) = s^{(d-1)} + k_{d-2}s^{(d-2)} + \cdots + k_1s + k_0$ 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

$$u_{ce} = \frac{1}{\beta_k(t) + \hat{\beta}(x)}(-(\alpha_k(t) + \hat{\alpha}(x)) + \nu(t)) \tag{34}$$

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

$$\nu(t) = y_m^{(d)} + \eta e_s + \bar{e}_s \tag{35}$$

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

$$\begin{aligned} \dot{\theta}_\alpha(t) &= -Q_\alpha^{-1}\phi_\alpha(x)e_s && (36)\\ \dot{\theta}_\beta(t) &= -Q_\beta^{-1}\phi_\beta(x)e_su_{ce} && (37) \end{aligned}$$

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 projection method to ensure this, in particular, to make sure that $\beta_k(t) + \beta(x) \geq \beta_0$. Additionally, the sliding mode control term is defined as

$$u_{si} = \frac{(W_\alpha(x) + W_\beta(x)|u_{ce}|)}{\beta_0}sgn(e_s) \tag{38}$$

3.2 Stability Analysis

It is of particular interest to study the zero dynamics of the system with relative degree $d < n$ to achieve state boundedness. The dynamics for a relative degree d system described by (19,20) may be written in normal form as

$$\begin{aligned}
\dot{\xi}_1 &= \xi_2 && (39)\\
\dot{\xi}_2 &= \xi_3 && (40)\\
&\vdots\\
\dot{\xi}_{d-1} &= \xi_d && (41)\\
\dot{\xi}_d &= \alpha(\xi,\pi)+\beta(\xi,\pi)u && (42)\\
\dot{\pi} &= f_0(\xi,\pi) && (43)
\end{aligned}$$

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

$$z = T(x) = \begin{bmatrix} \psi_1(x) \\ \vdots \\ \psi_d(x) \\ \cdots\cdots \\ \phi_1(x) \\ \vdots \\ \phi_{n-d}(x) \end{bmatrix} = \begin{bmatrix} \psi(x) \\ \cdots\cdots \\ \phi(x) \end{bmatrix} = \begin{bmatrix} \xi \\ \cdots \\ \pi \end{bmatrix} \tag{44}$$

where $\psi_1(x)=h(x)$, $\psi_{i+1}(x)=(\partial\psi_i/\partial x)^\top f(x), i=1,2,\ldots,d-1$ and $(\partial\phi_j/\partial x)^\top g(x)=0, j=1,2,\ldots,n-d$ for any $x\in D_x$, and $T(x)$ is a diffeomorphism on a domain $D_x \subset S_x$ (The map T is invertible and both $T(.)$ and $T^{-1}(.)$ are continuously differentiable); [39] shows the existence of $\phi(x)$ and $T(x)$. The normal form decomposes the system states into an external part ξ and an internal part π. The external part is stabilized by the control u (which we will show later), while the internal part is made unobservable by the same control. Note that by having $\xi=0$ in the inner part we obtain the "zero dynamics" of the system

$$\dot{\pi} = f_0(0,\pi) \tag{45}$$

In particular, if the origin ($\xi=0,\pi=0$) is an equilibrium point, the exponential stability of zero dynamics may be studied around the origin. Actually, the zero dynamics can be characterized in the original x-coordinates [39]. Notice that keeping the output identically zero ($y=0$) gives $\xi=0$, which implies that the solution of the state equation must be confined to the set

$$z^* = \{x \in S_x : \psi_1(x)=\psi_2(x)=\cdots=\psi_d(x)=0\} \tag{46}$$

and keeping the output identically zero also gives

$$u = u^*(x) = \frac{-\alpha(\xi,\pi)+k\xi}{\beta(\xi,\pi)} \tag{47}$$

so that the zero dynamics in the original x-coordinates is

$$\dot{x} = f^*(x) = f(x) + g(x)u|_{x \in z^*, u = u^*(x)} \tag{48}$$

Consider the model of the engine

$$\begin{aligned} \dot{x}_1 &= f_1(x,c,p) + g_1(x,c,p)u & (49) \\ \dot{x}_2 &= f_2(x,c,p) + g_2(x,c,p)u & (50) \\ y &= x_1 & (51) \end{aligned}$$

To simplify the stability analysis we change the variables $x_1 = \bar{x}_1 + x_1^0, x_2 = \bar{x}_2 + x_2^0, u = \bar{u} + u^0, y = \bar{y} + x_1^0$ to transform the engine model to the form of

$$\begin{aligned} \dot{\bar{x}}_1 &= [f_1(\bar{x}+x^0,c,p) + g_1(\bar{x}+x^0,c,p)u^0] + g_1(\bar{x}+x^0,c,p)\bar{u} & (52) \\ \dot{\bar{x}}_2 &= [f_2(\bar{x}+x^0,c,p) + g_2(\bar{x}+x^0,c,p)u^0] + g_2(\bar{x}+x^0,c,p)\bar{u} & (53) \end{aligned}$$

We assume that the above system has an open-loop equilibrium at the origin, that is, $f_1(x^0,c,p) + g_1(x^0,c,p)u^0 = 0$ and $f_2(x^0,c,p) + g_2(x^0,c,p)u^0 = 0$. The derivative of the output is given by

$$\dot{\bar{y}} = [f_1(\bar{x}+x^0,c,p) + g_1(\bar{x}+x^0,c,p)u^0] + g_1(\bar{x}+x^0,c,p)\bar{u} \tag{54}$$

Since we know $g_1(\bar{x}+x^0,c,p) > 0$ for all x, c, p, the engine has relative degree 1. To characterize the zero dynamics, restrict x to

$$z^* = \{x \in S_x : \psi_1(x) = \xi_1 = \bar{x}_1 = 0\} \tag{55}$$

and take

$$u = u^*(x) = \frac{-f_1(\bar{x}+x^0,c,p) - g_1(\bar{x}+x^0,c,p)u^0 + k\bar{x}_1}{g_1(\bar{x}+x^0,c,p)} \tag{56}$$

we have

$$\begin{aligned} \dot{\bar{x}}_2 &= f_0(0,\bar{x}_2) \\ &= [f_2(x_1^0,\bar{x}_2+x_2^0,c,p) + g_2(x_1^0,\bar{x}_2+x_2^0,c,p)u^0] + \frac{g_2(x_1^0,\bar{x}_2+x_2^0,c,p)}{g_1(x_1^0,\bar{x}_2+x_2^0,c,p)}[-f_1(x_1^0,\bar{x}_2+x_2^0,c,p) \\ &\quad -g_1(x_1^0,\bar{x}_2+x_2^0,c,p)u^0] \end{aligned}$$

Now we can study the zero dynamics of the system from the above equation. Notice that

$$\begin{aligned}
\underline{f}_1(x_1^0, \bar{x}_2 + x_2^0, c_i, p_i) &= \frac{\sum_{j=1}^{R}[a_{j,0}^1(c_i,p_i) + a_{j,1}^1(c_i,p_i)x_1^0 + a_{j,2}^1(c_i,p_i)(\bar{x}_2 + x_2^0)]\underline{\mu}_j(x_1^0)}{\sum_{j=1}^{R}\underline{\mu}_j(x_1^0)} && (57)\\
&= \underline{f}_1(x_1^0, x_2^0, c_i, p_i) + \beta_i^1(x_1^0, c_i, p_i)\bar{x}_2 && (58)\\
f_1(x_1^0, \bar{x}_2 + x_2^0, c, p) &= \frac{\sum_{i=1}^{N}[\underline{f}_1(x_1^0, \bar{x}_2 + x_2^0, c_i, p_i)]\mu_i(c,p)}{\sum_{i=1}^{N}\mu_i(c,p)} && (59)\\
&= f_1(x_1^0, x_2^0, c, p) + \beta_1(x_1^0, c, p)\bar{x}_2 && (60)\\
\underline{g}_1(x_1^0, \bar{x}_2 + x_2^0, c_i, p_i) &= \frac{\sum_{j=1}^{R} a_{j,3}^1(c_i,p_i)\underline{\mu}_j(x_1^0)}{\sum_{j=1}^{R}\underline{\mu}_j(x_1^0)} && (61)\\
&= \underline{g}_1(x_1^0, c_i, p_i) = \alpha_i^1(x_1^0, c_i, p_i) && (62)\\
g_1(x_1^0, \bar{x}_2 + x_2^0, c, p) &= \frac{\sum_{i=1}^{N}[\underline{g}_1(x_1^0, \bar{x}_2 + x_2^0, c_i, p_i)]\mu_i(c,p)}{\sum_{i=1}^{N}\mu_i(c,p)} && (63)\\
&= g_1(x_1^0, c, p) = \alpha_1(x_1^0, c, p) && (64)
\end{aligned}$$

and, similarly,

$$\begin{aligned}
f_2(x_1^0, \bar{x}_2 + x_2^0, c, p) &= f_2(x_1^0, x_2^0, c, p) + \beta_2(x_1^0, c, p)\bar{x}_2 && (65)\\
g_2(x_1^0, \bar{x}_2 + x_2^0, c, p) &= g_2(x_1^0, c, p) = \alpha_2(x_1^0, c, p) && (66)
\end{aligned}$$

so that we have

$$\begin{aligned}
\dot{\bar{x}}_2 &= f_0(0, \bar{x}_2)\\
&= [f_2(x_1^0, x_2^0, c, p) + \beta_2(x_1^0, c, p)\bar{x}_2 + g_2(x_1^0, c, p)u^0] + \frac{\alpha_2(x_1^0, c, p)}{\alpha_1(x_1^0, c, p)}[-f_1(x_1^0, x_2^0, c, p)\\
&\quad -\beta_1(x_1^0, c, p)\bar{x}_2 - g_1(x_1^0, c, p)u^0]\\
&= \beta_2(x_1^0, c, p)\bar{x}_2 - \frac{\alpha_2(x_1^0, c, p)}{\alpha_1(x_1^0, c, p)}\beta_1(x_1^0, c, p)\bar{x}_2\\
&= \left(\beta_2(x_1^0, c, p) - \frac{\alpha_2(x_1^0, c, p)}{\alpha_1(x_1^0, c, p)}\beta_1(x_1^0, c, p)\right)\bar{x}_2
\end{aligned}$$

Since we know that $a_{j,3}^1(c_i,p_i) > a_{j,3}^2(c_i,p_i) > 0$ and $a_{j,2}^2(c_i,p_i) < a_{j,2}^1(c_i,p_i) < 0$ for any $i = 1, 2, \ldots, N, j = 1, 2, \ldots, R$, we have

$$\frac{\sum_{j=1}^{R} a_{j,3}^1(c_i,p_i)\underline{\mu}_j(x_1^0)}{\sum_{j=1}^{R}\underline{\mu}_j(x_1^0)} > \frac{\sum_{j=1}^{R} a_{j,3}^2(c_i,p_i)\underline{\mu}_j(x_1^0)}{\sum_{j=1}^{R}\underline{\mu}_j(x_1^0)} > 0$$

that is $\alpha_i^1(x_1^0, c_i, p_i) > \alpha_i^2(x_1^0, c_i, p_i) > 0$ and

$$\frac{\sum_{i=1}^{N}\alpha_i^1(x_1^0, c_i, p_i)\mu_i(c,p)}{\sum_{i=1}^{N}\mu_i(c,p)} > \frac{\sum_{i=1}^{N}\alpha_i^2(x_1^0, c_i, p_i)\mu_i(c,p)}{\sum_{i=1}^{N}\mu_i(c,p)} > 0$$

that is $\alpha_1(x_1^0, c, p) > \alpha_2(x_1^0, c, p) > 0$ so that $0 < \frac{\alpha_2(x_1^0,c,p)}{\alpha_1(x_1^0,c,p)} < 1$. Also, notice that

$$\begin{aligned}
& \beta_i^2(x_1^0, c, p) - \frac{\alpha_2(x_1^0, c, p)}{\alpha_1(x_1^0, c, p)} \beta_i^1(x_1^0, c, p) \\
= \;& \frac{\sum_{j=1}^R a_{j,2}^2(c_i, p_i) \underline{\mu}_j(x_1^0)}{\sum_{j=1}^R \underline{\mu}_j(x_1^0)} - \frac{\alpha_2(x_1^0, c, p)}{\alpha_1(x_1^0, c, p)} \frac{\sum_{j=1}^R a_{j,2}^1(c_i, p_i) \underline{\mu}_j(x_1^0)}{\sum_{j=1}^R \underline{\mu}_j(x_1^0)} \\
= \;& \frac{\sum_{j=1}^R \left[a_{j,2}^2(c_i, p_i) - \frac{\alpha_2(x_1^0,c,p)}{\alpha_1(x_1^0,c,p)} a_{j,2}^1(c_i, p_i) \right] \underline{\mu}_j(x_1^0)}{\sum_{j=1}^R \underline{\mu}_j(x_1^0)} \\
< \;& \frac{\sum_{j=1}^R \left[a_{j,2}^2(c_i, p_i) - a_{j,2}^1(c_i, p_i) \right] \underline{\mu}_j(x_1^0)}{\sum_{j=1}^R \underline{\mu}_j(x_1^0)} \\
< \;& 0
\end{aligned}$$

so that

$$\begin{aligned}
& \beta_2(x_1^0, c, p) - \frac{\alpha_2(x_1^0, c, p)}{\alpha_1(x_1^0, c, p)} \beta_1(x_1^0, c, p) \\
= \;& \frac{\sum_{i=1}^N \beta_i^2(x_1^0, c, p) \mu_i(c, p)}{\sum_{i=1}^N \mu_i(c, p)} - \frac{\alpha_2(x_1^0, c, p)}{\alpha_1(x_1^0, c, p)} \frac{\sum_{i=1}^N \beta_i^1(x_1^0, c, p) \mu_i(c, p)}{\sum_{i=1}^N \mu_i(c, p)} \\
= \;& \frac{\sum_{i=1}^N [\beta_i^2(x_1^0, c, p) - \frac{\alpha_2(x_1^0,c,p)}{\alpha_1(x_1^0,c,p)} \beta_i^1(x_1^0, c, p)] \mu_i(c, p)}{\sum_{i=1}^N \mu_i(c, p)} \\
< \;& 0
\end{aligned}$$

Therefore, the origin of $\dot{\bar{x}}_2 = f_0(0, \bar{x}_2)$ is exponentially stable.

To show that the external part ξ is stabilized by the control u, consider the following Lyapunov function candidate [38]

$$V_i = \frac{1}{2} e_s^2 + \frac{1}{2} \tilde{\theta}_\alpha^\top Q_\alpha \tilde{\theta}_\alpha + \frac{1}{2} \tilde{\theta}_\beta^\top Q_\beta \tilde{\theta}_\beta \tag{67}$$

which quantifies both errors in tracking and in parameter estimation. Using vector derivatives and following [38], the time derivative of (67) is

$$\dot{V}_i = e_s \dot{e}_s + \tilde{\theta}_\alpha^\top Q_\alpha \dot{\tilde{\theta}}_\alpha + \tilde{\theta}_\beta^\top Q_\beta \dot{\tilde{\theta}}_\beta \tag{68}$$

Note that $\bar{e}_s(t) = \dot{e}_s(t) - e^{(d)}(t)$ and the d^{th} derivative of the output error is $e^{(d)} = y_m^{(d)} - y^{(d)}$ so that

$$\dot{e}_s(t) = \bar{e}_s(t) + y_m^{(d)} - y^{(d)} \tag{69}$$

and from (21), (33), (35) and (34)

$$
\begin{aligned}
\dot{e}_s(t) &= \bar{e}_s(t) + [\nu(t) - \eta e_s - \bar{e}_s] - [(\alpha_k(t) + \alpha(x)) + (\beta_k(t) + \beta(x))(u_{ce} + u_{si})] \\
&= -\eta e_s + [\nu(t) - \alpha_k(t) - (\beta_k(t) + \hat{\beta}(x))u_{ce}] - \alpha(x) + (\hat{\beta}(x) - \beta(x))u_{ce} \\
&\quad -(\beta_k(t) + \beta(x))u_{si} \\
&= -\eta e_s + [\nu(t) - \alpha_k(t) + (\alpha_k(t) + \hat{\alpha}(x)) - \nu(t)] - \alpha(x) + (\hat{\beta}(x) - \beta(x))u_{ce} \\
&\quad -(\beta_k(t) + \beta(x))u_{si} \\
&= -\eta e_s + (\hat{\alpha}(x) - \alpha(x)) + (\hat{\beta}(x) - \beta(x))u_{ce} - (\beta_k(t) + \beta(x))u_{si}
\end{aligned}
$$

also from (22,23), (24,25) and (30,31) we have

$$
\dot{e}_s = -\eta e_s + (\tilde{\theta}_\alpha^\top \phi_\alpha(x) - \omega_\alpha(x)) + (\tilde{\theta}_\beta^\top \phi_\beta(x) - \omega_\beta(x))u_{ce} - (\beta_k(t) + \beta(x))u_{si} \tag{70}
$$

Substitute the above equation into (68), and assume that the ideal parameters are constant so that $\dot{\tilde{\theta}}_\alpha = \dot{\theta}_\alpha$ and $\dot{\tilde{\theta}}_\beta = \dot{\theta}_\beta$ and substitute (36,37) into (68)

$$
\begin{aligned}
\dot{V}_i &= e_s[-\eta e_s + (\tilde{\theta}_\alpha^\top \phi_\alpha(x) - \omega_\alpha(x)) + (\tilde{\theta}_\beta^\top \phi_\beta(x) - \omega_\beta(x))u_{ce} - (\beta_k(t) + \beta(x))u_{si}] \\
&\quad + \tilde{\theta}_\alpha^\top Q_\alpha[-Q_\alpha^{-1}\phi_\alpha(x)e_s] + \tilde{\theta}_\beta^\top Q_\beta[-Q_\beta^{-1}\phi_\beta(x)e_s u_{ce}] \\
&= -\eta e_s^2 - (\omega_\alpha(x) + \omega_\beta(x)u_{ce})e_s - (\beta_k(t) + \beta(x))u_{si}e_s
\end{aligned}
$$

Notice that we did not consider projection modification to the update laws above. Clearly, since $\theta_\alpha^* \in \Omega_\alpha$ and $\theta_\beta^* \in \Omega_\beta$, when the projection is in effect it always results in smaller parameter errors that will decrease $\dot{V}_i$ so that

$$
\dot{V}_i \le -\eta e_s^2 - (\omega_\alpha(x) + \omega_\beta(x)u_{ce})e_s - (\beta_k(t) + \beta(x))u_{si}e_s \tag{71}
$$

Substitute (38) into the above equation and also notice that $-(\omega_\alpha(x)+\omega_\beta(x)u_{ce})e_s \le (|\omega_\alpha(x)| + |\omega_\beta(x)u_{ce}|)|e_s| \le (W_\alpha(x) + W_\beta(x)|u_{ce}|)|e_s|$ and $\beta_k(t) + \beta(x) \ge \beta_0$ and $|e_s| = e_s sgn(e_s)$ (except at $e_s = 0$)

$$
\begin{aligned}
\dot{V}_i &\le -\eta e_s^2 + (W_\alpha(x) + W_\beta(x)|u_{ce}|)|e_s| - (\beta_k(t) + \beta(x))\left(\frac{(W_\alpha(x) + W_\beta(x)|u_{ce}|)}{\beta_0} sgn(e_s)\right) e_s \\
&\le -\eta e_s^2 + (W_\alpha(x) + W_\beta(x)|u_{ce}|)|e_s| - \frac{(\beta_k(t) + \beta(x))}{\beta_0}(W_\alpha(x) + W_\beta(x)|u_{ce}|)|e_s| \\
&\le -\eta e_s^2
\end{aligned}
$$

Thus, $\dot{V}_i$ is negative semidefinite which means V_i is a nonincreasing function of time, that is, the measure of the tracking error e_s is bounded. Notice that $e_s(s) = L(s)e(s)$ and $L(s)$ is a stable function with the degree of $d-1$ we known that the tracking error and its derivatives $e, \dot{e}, \ldots, e^{(d-1)}$ are bounded. Since the reference trajectory y_m and its derivatives $\dot{y}_m, \ldots, y_m^{(d)}$ are assumed to be bounded, the system output y and its derivatives $\dot{y}, \ldots, y^{(d)}$ are bounded. Hence, ξ is bounded and thus x is bounded.

Besides, the fact that $\dot{V}_i$ is negative semidefinite also implies that parameter estimations θ_α

and θ_β are bounded (noting (30,31) and the boundedness of θ_α^* and θ_β^*.) Therefore, the boundedness of $\hat{\alpha}(x)$, $\hat{\beta}(x)$, $\alpha_k(t)$, and $\beta_k(t)$ assures that u_{ce} and u_{si} and hence u are bounded.

To show asymptotic stability of the output, note that

$$\int_0^\infty \eta e_s^2 dt \leq -\int_0^\infty \dot{V}_i dt = V_i(0) - V_i(\infty) \tag{72}$$

this establishes that $e_s \in L_2$ ($L_2 = \{z(t) : \int_0^\infty z^2(t) < \infty\}$) since $V_i(0)$ and $V_i(\infty)$ are bounded. Since e_s and $\dot{e}_s$ are bounded and $e_s \in L_2$, by Barbalat's Lemma we have asymptotic stability of e_s (i.e., $\lim_{t\to\infty} e_s = 0$), which implies asymptotic stability of the tracking error e (i.e., $\lim_{t\to\infty} e = 0$).

3.3 Direct Adaptive control

In addition to the assumptions we made in the indirect adaptive control case, we require $\beta_k(t) = \alpha_k(t) = 0$ for all $t \geq 0$, and that there exist positive constants β_0 and β_1 such that $0 < \beta_0 \leq \beta(x) \leq \beta_1$. Also, we assume that we can specify some function $B(x) \geq 0$ such that $|\dot{\beta}(x)| = \left| \left(\frac{\partial \beta}{\partial x}\right)^\top \dot{x} \right| \leq B(x)$ for all $x \in S_x$. We know that there exists some ideal controller

$$u^* = \frac{1}{\beta(x)}(-\alpha(x) + \nu(t)) \tag{73}$$

where $\nu(t)$ is defined the same as in the indirect adaptive control case. Let

$$u^* = \theta_u^{*\top} \phi_u(x, \nu) + u_k(t) + \omega_u(x, \nu) \tag{74}$$

where u_k is a known part of the controller (e.g., one that was designed for the nominal system without fault) and

$$\theta_u^* = \arg \min_{\theta_u \in \Omega_u} \left(\sup_{x \in S_x, \nu \in S_\nu} |\theta_u^\top \phi_u(x, \nu) - (u^* - u_k)| \right) \tag{75}$$

so that $\omega_u(x, \nu)$ is the approximation error. We assume that $|\omega_u(x, \nu)| \leq W_u(x, \nu)$, where $W_u(x, \nu)$ is a known bound on the error in representing the ideal controller. The approximation of the ideal controller may be represented by

$$\hat{u} = \theta_u^\top \phi_u(x, \nu) + u_k(t) \tag{76}$$

where the parameter vector $\theta_u(t)$ is updated on line and the parameter error is

$$\tilde{\theta}_u(t) = \theta_u(t) - \theta_u^* \tag{77}$$

Consider the direct adaptive control law

$$u = \hat{u} + u_{sd} \tag{78}$$

which is the sum of the approximation to the ideal control law and a sliding mode control term

$$u_{sd} = \left(\frac{B(x)|e_s|}{2\beta_0^2} + W_u(x,\nu) \right) sgn(e_s) \tag{79}$$

and we use the update law

$$\dot{\theta}_u(t) = Q_u^{-1}\phi_u(x,\nu)e_s(t) \tag{80}$$

where Q_u is positive definite and diagonal. We also use a projection method to ensure that $\theta_u \in \Omega_u$.

Consider the following Lyapunov function candidate

$$V_d = \frac{1}{2\beta(x)}e_s^2 + \frac{1}{2}\tilde{\theta}_u^\top Q_u \tilde{\theta}_u \tag{81}$$

taking the time derivative

$$\dot{V}_d = \frac{e_s}{\beta(x)}\dot{e}_s - \frac{\dot{\beta}(x)e_s^2}{2\beta^2(x)} + \tilde{\theta}_u^\top Q_u \dot{\tilde{\theta}}_u \tag{82}$$

Note that $\bar{e}_s(t) = \dot{e}_s(t) - e^{(d)}(t)$ and the d^{th} derivative of the output error is $e^{(d)} = y_m^{(d)} - y^{(d)}$ so that

$$\dot{e}_s(t) = \bar{e}_s(t) + y_m^{(d)} - y^{(d)} \tag{83}$$

and from (21), (35), (78) and (73) and by assuming $\alpha_k(t) = \beta_k(t) = 0$

$$\begin{aligned}
\dot{e}_s(t) &= \bar{e}_s(t) + [\nu(t) - \eta e_s - \bar{e}_s] - [(\alpha_k(t) + \alpha(x)) + (\beta_k(t) + \beta(x))(\hat{u} + u_{sd})] \\
&= -\eta e_s + [\nu(t) - \alpha(x) - \beta(x)u^*] - \beta(x)(\hat{u} - u^*) - \beta(x)u_{sd} \\
&= -\eta e_s - \beta(x)(\hat{u} - u^*) - \beta(x)u_{sd}
\end{aligned}$$

also from (76), (73) and (77) we have

$$\dot{e}_s(t) = -\eta e_s - \beta(x)(\tilde{\theta}_u^\top \phi_u(x,\nu) - \omega_u(x,\nu)) - \beta(x)u_{sd} \tag{84}$$

Substitute the above equation into (82), and assume that the ideal parameters are constant so that $\dot{\tilde{\theta}}_u = \dot{\theta}_u$ and substitute (80) into (82)

$$\begin{aligned}
\dot{V}_d &= \frac{e_s}{\beta(x)}[-\eta e_s - \beta(x)(\tilde{\theta}_u^\top \phi_u(x,\nu) - \omega_u(x,\nu)) - \beta(x)u_{sd}] - \frac{\dot{\beta}(x)e_s^2}{2\beta^2(x)} + \tilde{\theta}_u^\top Q_u[Q_u^{-1}\phi_u(x,\nu)e_s(t)] \\
&= -\frac{\eta e_s^2}{\beta(x)} - \left(\tilde{\theta}_u^\top \phi_u - \omega_u(x,\nu) + \frac{\dot{\beta}(x)e_s}{2\beta^2(x)} - \tilde{\theta}_u^\top \phi_u \right) e_s - e_s u_{sd} \\
&= -\frac{\eta e_s^2}{\beta(x)} - \left(\frac{\dot{\beta}(x)e_s}{2\beta^2(x)} - \omega_u(x,\nu) \right) e_s - e_s u_{sd}
\end{aligned}$$

After we consider the projection modification to the update law we have

$$\dot{V}_d \leq -\frac{\eta e_s^2}{\beta(x)} - \left(\frac{\dot{\beta}(x)e_s}{2\beta^2(x)} - \omega_u(x,\nu)\right) e_s - e_s u_{sd} \tag{85}$$

Substitute (79) into the above equation and notice that

$$-\left(\frac{\dot{\beta}(x)e_s}{2\beta^2(x)} - \omega_u(x,\nu)\right) e_s \leq \left(\frac{|\dot{\beta}(x)||e_s|}{2\beta^2(x)} + |\omega_u(x,\nu)|\right)|e_s| \leq \left(\frac{B(x)|e_s|}{2\beta_0^2(x)} + W_u(x,\nu)\right)|e_s|$$

and $0 < \beta_0 \leq \beta(x) \leq \beta_1$ we have

$$\dot{V}_d \leq -\frac{\eta e_s^2}{\beta(x)} \tag{86}$$

so that V_d is a nonincreasing function of time. This gives the same type of stability result that we obtained in the indirect case.

4 Component Level Model Simulation

To study the effectiveness of the proposed fault tolerant control methods, we apply them to the component level model simulation, which we treat as the real application on the XTE46 engine. Actually, we first applied them to the design model but in the interest of brevity we do not show those (slightly better) results here.

4.1 Indirect Adaptive Control

Consider the engine in the form of

$$\begin{aligned} \dot{y} &= f_1(x,c,p) + g_1(x,c,p)u & (87)\\ &= [\alpha_k(c,p_0,t) + \alpha(x,c,p)] + [\beta_k(c,p_0,t) + \beta(x,c,p)]u & (88) \end{aligned}$$

where $y = x_1$. Since $g_1(x,c,p) > 0$ for any x, c, p, the relative degree of the engine is 1. Furthermore, 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. 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 due to fault effects. They will be approximated by two Takagi-Sugeno fuzzy systems $\hat{\alpha}$ and $\hat{\beta}$ with $R = 11$ rules for each. The inputs to the fuzzy systems include two state variables and the parameters are updated on line to capture the unknown 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 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 $\bar{e}_s(t) = 0$).

Note that the sliding mode control term can introduce a high frequency signal to the plant which may excite unmodeled dynamics. To avoid this, we use a "smoothed" sliding mode control term

$$u_{si} = \frac{(W_\alpha(x) + W_\beta(x)|u_{ce}|)}{\beta_0} sat(e_s/\epsilon) \tag{89}$$

where $\epsilon > 0$ and

$$sat(x) = \begin{cases} 1, & \text{if } x \geq 1 \\ x, & \text{if } -1 < x < 1 \\ -1, & \text{if } x \leq -1 \end{cases} \tag{90}$$

By using this smoothed control action the tracking error will converge asymptotically to an ϵ-neighborhood of $e = 0$ [38]. Taking into account of the engine dynamics, the model uncertainty is described by $W_\alpha = 0.01$ and $W_\beta = 0.01$, the adaptation gains are $Q_\alpha^{-1} = 5e-8$ and $Q_\beta^{-1} = 1e-17$, and the design parameters are chosen to be $\eta = 1$ and $\epsilon = 10$.

We let the CLM 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. The reference trajectory is defined by passing a square wave through a linear reference model $\frac{Y_m(s)}{R(s)} = \frac{3}{s+3}$. The control performance for an engine without any fault is shown in Figure 3 and Figure 4 shows the control result for an engine with large fan fault. The indirect adaptive controller is able to quickly control the engine even in the presence of a large fan fault. This is because the controller can take advantage of the nominal model to have *a priori* knowledge of the engine, and its adaption scheme can let fuzzy approximators learn the profile of faults so that the control action can be modified to accommodate the fault. Notice that there is some small oscillation caused by parameter updates and there is a small ripple for the large fan fault case which implies that there may exist some high frequency dynamics that can not be learned by the approximator.

We also studied the performance for sine inputs (and many other cases). This time we let the CLM run at a different operating condition (ALT = 16000, XM = 0.72, DTAMB = 0, and PC = 49), the initial engine variation be $p_0 = [0.2\%, 0.1\%, 0.3\%, 0.1\%, -0.1\%, 0, -0.3\%, 0.2\%, -0.1\%, 0.1\%]$, and the engine deterioration index be 0.3. The control performance for an engine with a small fan fault and a medium compressor hub fault is shown in Figure 5. Note that in the first 5 seconds the adaptation scheme operates actively to learn the characteristics of the large fan fault. Afterwards, there is little adaptation and the control is quite smooth.

4.2 Direct Adaptive Control

We will now show how to apply the direct adaptive control scheme to the fault tolerant engine control problem. For direct adaptive control scheme, the nominal engine model cannot be used. Instead, we define the known controller to be a proportional-integral (PI) controller

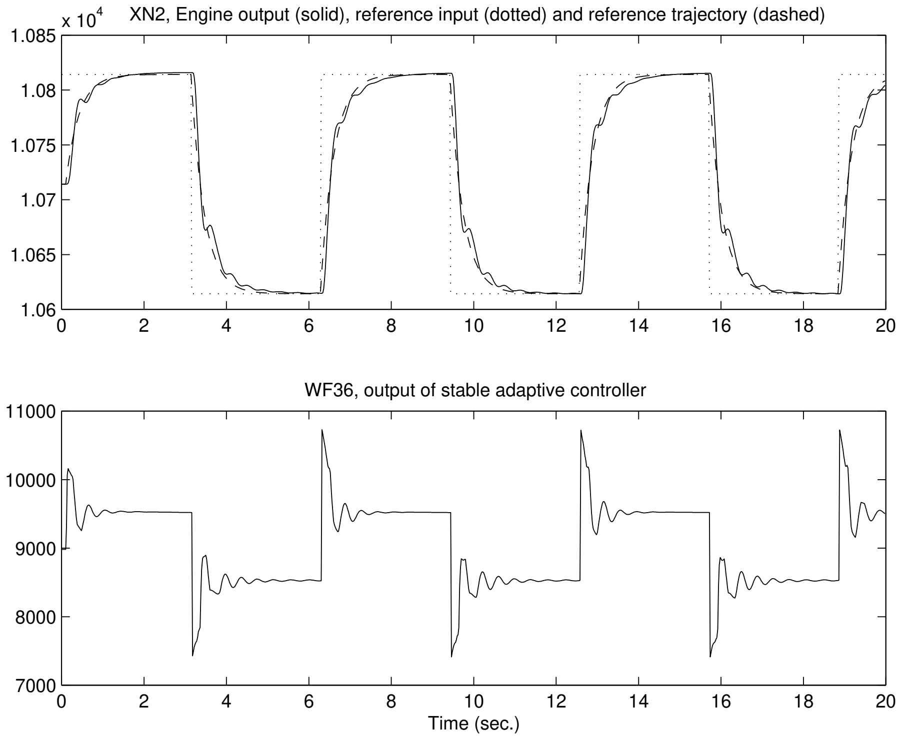

Figure 3: Performance of indirect adaptive controller for the engine without fault.

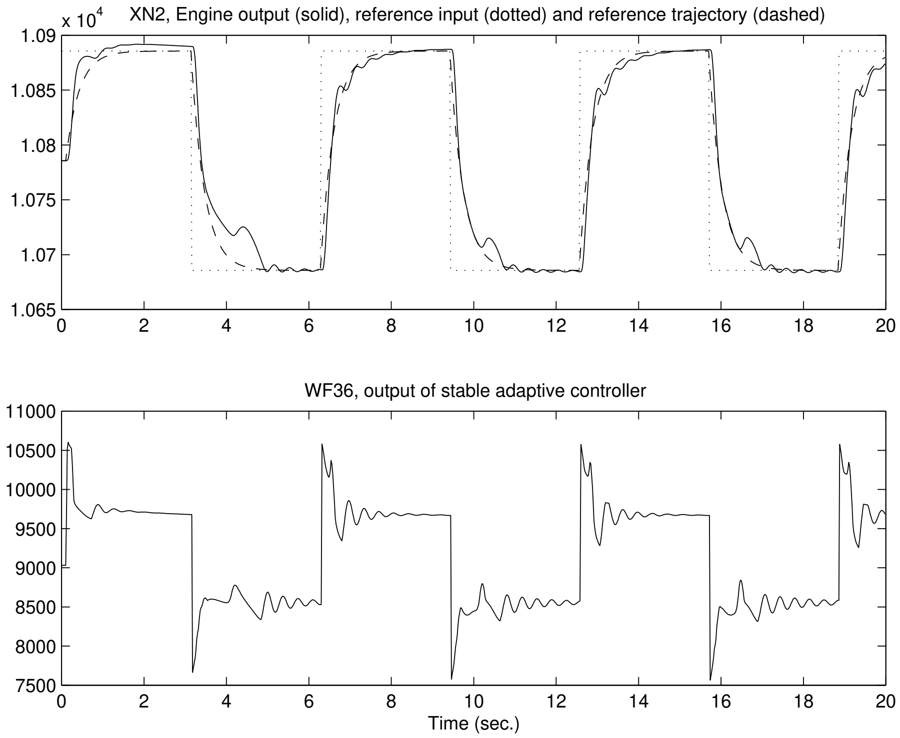

Figure 4: Performance of indirect adaptive controller for the engine with large fan fault.

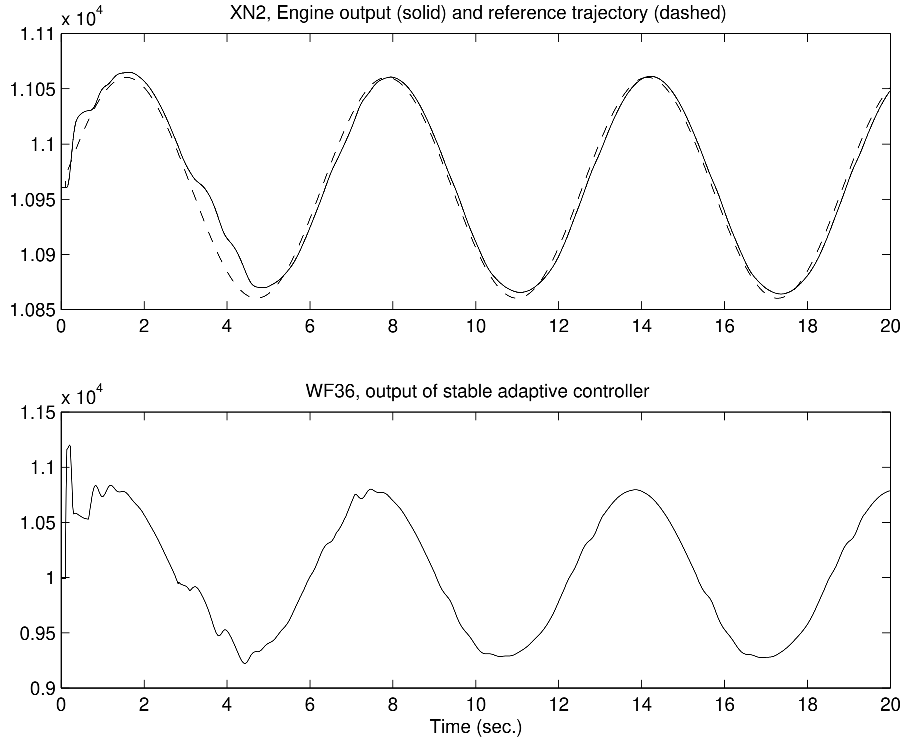

Figure 5: Performance of indirect adaptive controller for the engine with small fan fault and medium compressor hub fault (sine input).

($u_k = k_p(e + \frac{1}{T_i}\int edt)$, $k_p = 5, T_i = 0.2$). By studying dynamics of the developed nonlinear model we also know that $g_1(x, c, p) < 0.38$ and its rate of change is smaller than 1.5 so that we can set $\beta_1 = 0.38$ and $B = 1.5$. We also use a smoothed sliding mode control term here and the model uncertainty is described by $W_u = 200$, the adaptation gain is $Q_u^{-1} = 2e-7$ and the design parameters are chosen to be $\eta = 1$ and $\epsilon = 10$.

We try to compare the control result of the indirect adaptive controller with that of the direct adaptive controller, in particular, the large fan fault case as shown in Figure 4. The direct adaptive control result is shown in Figure 6. Note that there is more oscillation in the direct case. This is because the direct adaptive controller can not use *a priori* knowledge of the engine from the nominal engine model. Instead, it uses a known controller (its effect is shown in the figure using dashed line), which is not so good, so that the control action more heavily relies on the adaptation scheme. It is also interesting to compare the control result of the stable adaptive controller with that of a PI controller, tuned by the Ziegler-Nicholes technique. (Note that the PI controller is not gain scheduled since we only compare the performance at one operating point.) There is large overshoot caused by the change of system dynamics and the PI controller, of course, is unable to "learn" it. We may expect more performance deterioration for the PI controller when the fault is more serious, while the fault tolerant engine controller may recover from it. Similar studies have been performed for the sine input case, as shown in Figure 7 (and many other cases). The direct adaptive controller can learn the change of system dynamics quickly. For the PI controller, there is a large residual.

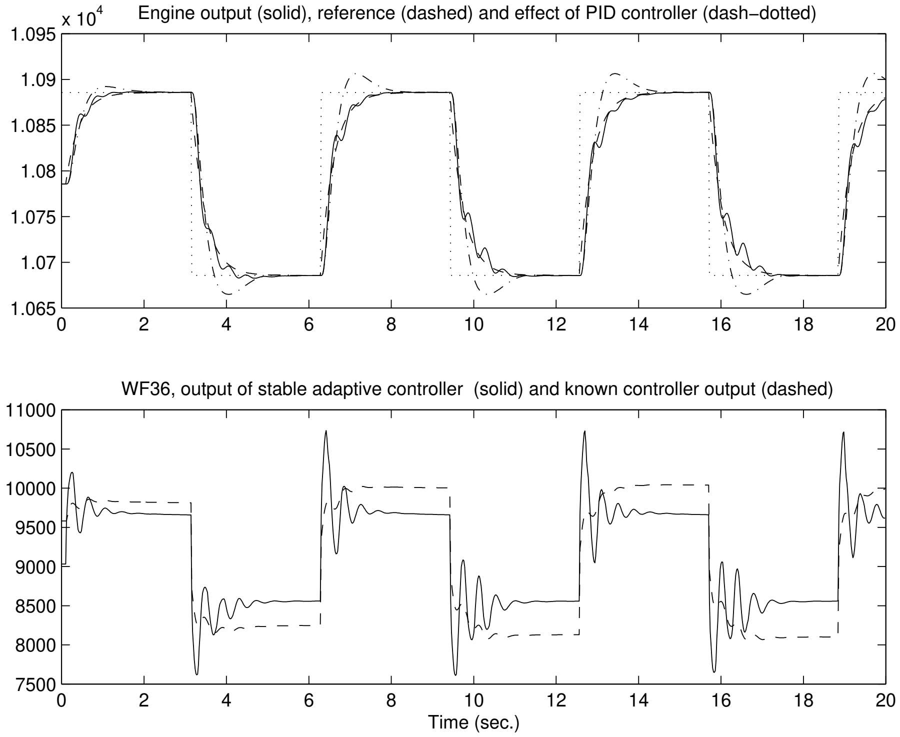

Figure 6: Performance of direct adaptive controller and PI controller for the engine with large fan fault.

5 Conclusion

Stable indirect and direct adaptive controllers are applied to achieve fault tolerant engine control by using Takagi-Sugeno fuzzy systems to "learn" the unknown dynamics caused by faults and to accommodate faults by updating the controller. We prove that both adaptive schemes achieve asymptotic tracking. The performance of the fault tolerant indirect and direct adaptive controller is demonstrated through the component level model simulation of the XTE46 engine.

Acknowledgment: We would like to thank S. Adibhatla at General Electric Aircraft Engines for his assistance and advice on many aspects of this project. Moreover, we would like to thank T.H. Guo at NASA Glenn Research Center for his support.

References

[1] 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.

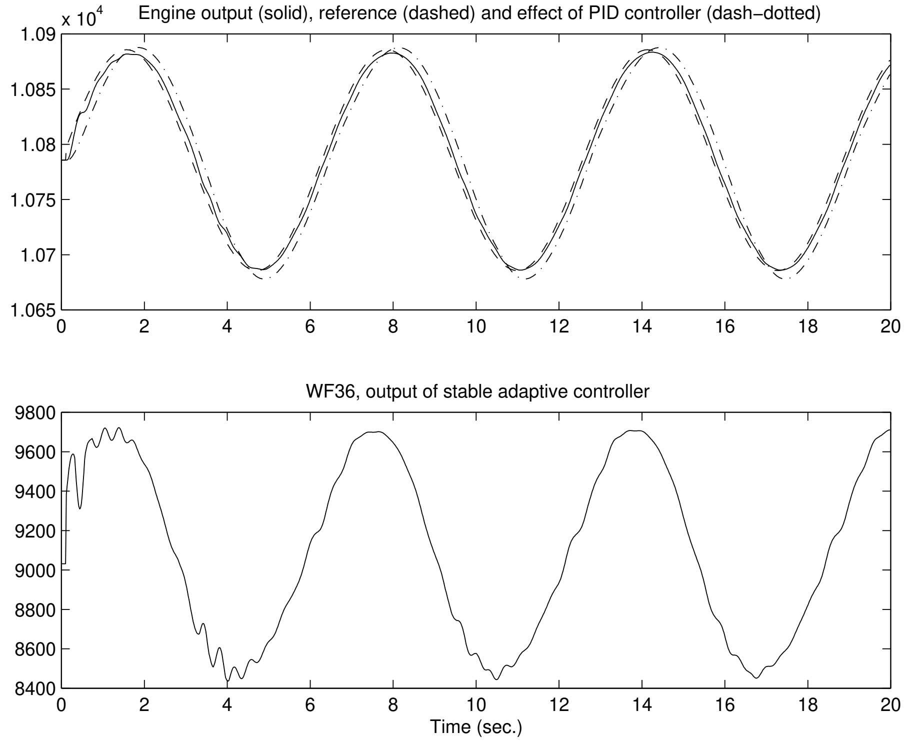

Figure 7: Performance of direct adaptive controller and PI controller for the engine with large fan fault (sine input).

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

[3] W. C. Merrill, "Sensor failure detection for jet engines using analytical redundancy," *Journal of Guidance, Control and Dynamics*, vol. 8, no. 6, pp. 673–682, 1985.

[4] W. C. Merrill, J. DeLaat, and W. Bruton, "Advanced detection, isolation, and accommodation of sensor failures - a real-time evaluation," *Journal of Guidance, Control and Dynamics*, vol. 11, no. 6, pp. 517–526, 1988.

[5] W. C. Merrill, J. C. DeLaat, and M. Abdelwahab, "Turbofan engine demonstration of sensor failure-detection," *Journal of Guidance, Control and Dynamics*, vol. 14, no. 2, pp. 337–349, 1991.

[6] A. Duyar and W. C. Merrill, "Fault diagnosis for the space shuttle main engine," *Journal of Guidance, Control and Dynamics*, vol. 15, no. 2, pp. 384–389, 1992.

[7] A. Duyar, V. Eldem, W. C. Merrill, and T. Guo, "Fault detection and diagnosis in propulsion systems - a fault parameter estimation approach," *Journal of Guidance, Control and Dynamics*, vol. 17, no. 1, pp. 104–108, 1994.

[8] R. Patton and J. Chen, "Robust fault detection of jet engine sensor systems using eigenstructure assignment," *Journal of Guidance, Control and Dynamics*, vol. 15, no. 6, pp. 1491–1497, 1992.

[9] R. Patton, J. Chen, and H.Y.Zhang, "Observer-based fault detection and isolation: robustness and applications," *Control Engineering Practice*, vol. 5, no. 5, pp. 671–682, 1997.

[10] R. Patton and J. Chen, "Modelling methods for improving robustness in fault diagnosis of jet engine system," in *Proceeding of the 31st IEEE Conference on Control and Decision*, (Tucson, Arizona), pp. 2330–2335, 1997.

[11] T. Menke and P. Maybeck, "Sensor/actuator failure-detection in the vista F-16 by multiple model adaptive estimation," *IEEE Transactions on Aerospace an Electronic Systems*, vol. 31, no. 4, pp. 1218–1229, 1995.

[12] P. Maybeck and R. Stevens, "Reconfigurable flight control via multiple model adaptive control methods," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 470–479, 1991.

[13] M. Athans, D. Castanon, K. Dunn, C. Greene, W. Lee, N. Sandell, and A. Willsky, "The stochastic control of the F-8C aircraft using a multiple model adaptive control (mmac) method - part 1: equilibrium flight," *IEEE Transactions on Automatic Control*, vol. 22, no. 5, pp. 768–780, 1977.

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

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

[16] Y. Ochi and K. Kanai, "Design of restructurable flight control systems using feedback linearization," *Journal of Guidance, Control and Dynamics*, vol. 14, no. 5, pp. 903–911, 1991.

[17] P. J. Antsaklis and K. M. Passino, eds., *An Introduction to Intelligent and Autonomous Control*. Norwell, MA: Kluwer Academic Publishers, 1993.

[18] R. Stengel, "Toward intelligent flight control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 6, pp. 1699–1717, 1993.

[19] K. Watanabe, I. Matsuura, M. Abe, M. Kubota, and D. Himmelblau, "Incipient fault diagnosis of chemical processes via artificial neural networks," *American Institute of Chemical Engineers Journal*, vol. 35, no. 11, pp. 1803–1812, 1989.

[20] S. Naidu, E. Zafiriou, and T. McAvoy, "Use of neural networks for failure detection in a control system," *IEEE Control Systems Magazine*, vol. 10, pp. 49–55, 1990.

[21] T. Sorsa, H. Koivo, and H. Koivisto, "Neural networks in process fault diagnosis," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 21, no. 4, pp. 815–825, 1991.

[22] T. Sorsa and H. Koivo, "Application of artificial neural networks in process fault diagnosis," *Automatica*, vol. 29, no. 4, pp. 843–849, 1993.

[23] Y. Maki and K. Loparo, "A neural network approach to fault detection and diagnosis in industrial processes," *IEEE Transactions on Control Systems Technology*, vol. 5, no. 6, pp. 529–541, 1997.

[24] S. Kavuri and V. Venkatasubramanian, "Neural network decomposition strategies for large scale fault diagnosis," *International Journal of Control*, vol. 59, no. 3, pp. 767–792, 1994.

[25] M. Polycarpou and A. Vemuri, "Learning methodology for failure detection and accommodation," *IEEE Control Systems Magazine*, vol. 15, no. 3, pp. 16–24, 1995.

[26] M. Polycarpou and A. Helmicki, "Automated fault detection and accommodation - a learning systems approach," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 25, no. 11, pp. 1447–1458, 1995.

[27] A. Vemuri and M. Polycarpou, "Robust nonlinear fault diagnosis in input-output systems," *International Journal of Control*, vol. 68, no. 2, pp. 343–360, 1997.

[28] A. Vemuri and M. Polycarpou, "Neural network based robust fault diagnosis in robotic systems," *IEEE Transactions on Neural Networks*, vol. 8, no. 6, pp. 1410–1420, 1997.

[29] A. Vemuri, M. Polycarpou, and S. Diakourtis, "Neural network based fault detection and accommodation in robotic manipulators," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 2, pp. 342–348, 1998.

[30] M. Demetriou and M. Polycarpou, "Incipient fault diagnosis of dynamical systems using on-line approximators," *IEEE Transactions on Automatic Control*, vol. 43, no. 11, pp. 1612–1617, 1998.

[31] K. M. Passino and P. J. Antsaklis, "Fault detection and identification in an intelligent restructurable controller," *Journal of Intelligent and Robotic Systems*, vol. 1, pp. 145–161, June 1988.

[32] E. G. Laukonen and K. M. Passino, "Training fuzzy systems to perform estimation and identification," *Engineering Applications of Artificial Intelligence*, vol. 8, no. 5, pp. 499–514, 1995.

[33] E. G. Laukonen, K. M. Passino, V. Krishnaswami, G.-C. Luh, and G. Rizzoni, "Fault detection and isolation for an experimental internal combustion engine via fuzzy identification," *IEEE Trans. on Control Systems Technology*, vol. 3, pp. 347–355, Sept. 1995.

[34] 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.

[35] J. Gremling and K. M. Passino, "Genetic adaptive failure estimation," in *Proceedings of the American Control Conference*, (Albuquerque, NM), pp. 908–912, June 1997.

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

[37] K. M. Passino and S. Yurkovich, eds., *Fuzzy Control.* Menlo Park, CA: Addison Wesley Longman, 1998.

[38] J. T. Spooner and K. M. Passino, "Stable adaptive control using fuzzy systems and neural networks," *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 3, pp. 339–359, 1996.

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