Predictor-Model-Based Least-Squares Model-Reference Adaptive Control with Chebyshev Orthogonal Polynomial Approximation

Nhan Nguyen*
NASA Ames Research Center, Moffett Field, CA 94035

This paper presents a model-reference adaptive control approach for systems with unstructured uncertainty based on two least-squares parameter estimation methods: gradient-based method and recursive least-squares method. The unstructured uncertainty is approximated by Chebyshev orthogonal polynomial basis functions. The use of orthogonal basis functions improves the function approximation significantly and enables better convergence of parameter estimates. The adaptation of least-squares adaptive control is driven by the plant modeling error which manifestes itself as the tracking error, but not vice versa. A predictor model is introduced to estimate the plant modeling error without the need for differentiation of the state signals which can introduce noise. The least-squares gradient adaptive control achieves superior parameter convergence as compared to the standard model-reference adaptive control. Flight control simulations were conducted with four adaptive controllers: least-squares gradient adaptive control, recursive least-squares adaptive control, standard model-reference adaptive control, and neural-network adaptive control. The results show that the recursive least-squares adaptive control achieves better robustness as measured by a time-delay margin while the least-squares gradient adaptive control achieves better tracking performance than both the standard model-reference adaptive control and neural-network adaptive control.

I. Introduction

In many physical applications, there is no clear certainty about the structure between the input and output of a process. This uncertainty is called unstructured. In systems with unstructured uncertainty, the transfer function between the input and output is usually not known. Let $y(t) \in \mathbb{R}$ be the output with an unknown transfer function, expressed as

$$y = f(x) \tag{1}$$

where $x(t) \in \mathscr{D} \subset \mathbb{R}^p$ and $f(x) \in \mathbb{R}$ is an unknown function but assumed to be bounded function in x.

When the structure of the uncertainty is unknown, function approximation is usually employed to estimate the unknown function. In recent years, neural networks have gained a lot of attention in function approximation theory in connection with adaptive control. Multi-layer neural networks have the capability of approximating an unknown function to within an arbitrary value of the approximation error. The universal approximation theorems for sigmoidal neural networks by Cybenko[1] and for radial basis functions by Micchelli[2] provide a theoretical justification of function approximation using neural networks. The use of multi-layer neural networks can create an additional complexity in the back propagation gradient-based training rules.

Polynomial approximation is a well-known regression technique for function approximation. In theory, as the degree of an approximating polynomial increases, the approximation error is expected to decrease. However, increasing the degree of the approximating polynomial beyond a theoretical limit could lead to oscillations in the approximating polynomial due to over-parametrization. Regularization techniques to constrain parameters have been developed to prevent over-parametrization.[3]

In this paper, we explore the use of a special class of polynomials, known as Chebyshev orthogonal polynomials, as basis functions for function approximation in an adaptive control setting. The Chebyshev orthogonal polynomials have been shown to provide the "best" approximation of a function over any other types of polynomials.[4] The use of the Chebyshev orthogonal polynomials in the context of adaptive control with unstructured uncertainty is developed

*Research Scientist, Intelligent Systems Division, AIAA Associate Fellow, nhan.t.nguyen@nasa.gov

in this paper. The adaptive control approach is based on a least-squares gradient method and a recursive least-squares (RLS) method. Least-squares adaptive control has been well-studied by numerous authors.[5–7,9–15] Lai and Wei established consistency of recursive extended least-squares and proposed simple modifications of the Åström-Wittenmark self-tuning regulator.[5] Artificial neural network based adaptive control using least-squares methods is presented by Suykens et al.[6] Guo studied a recursive least-squares algorithm with slowly decreasing weights for linear stochastic systems which is found to have self-convergence property.[7] Nguyen presented a hybrid adaptive control approach which combines a direct model-reference adaptive control to reduce the tracking error and an indirect recursive least-squares parameter estimation to reduce the modeling error simultaneously.[9–11] Zou et al. studied model-reference adaptive control using the Chebyshev orthogonal polynomials for attitude control of spacecraft.[12] Chowdhary and Johnson developed a least squares modification to adaptive control problems where the uncertainty can be linearly parametrized and the modified weight training law uses an estimate of the ideal weights formed online by solving a least squares problem using recorded and current data concurrently.[13–15]

A different approach is taken as compared to these prior study whereby adaptation is driven by the plant modeling error associated with the least-squares approach. In systems with uncertainty, the plant modeling error manifests itself as the tracking error, but not vice versa. Thus, least-squares adaptive control addressing the plant modeling error can be more effective than conventional model-reference adaptive control which is designed to reduced the tracking error. In the context of adaptive control, least-squares adaptive control using Chebyshev orthogonal polynomials as basis functions represents a novel approach that can be considered as an alternative design to the conventional model-reference adaptive control.

A previous study demonstrates the effectiveness of least-squares model-reference adaptive control.[8] One of the drawbacks of least-squares adaptive control is that the adaptive laws in general require the time derivatives of the state signals. Differentiation of any time signals can introduce a significant noise source which could affect the closed-loop stability of a control system. This paper addresses this drawback by introducing a predictor model to estimate the plant modeling error without the need for differentiation of the state signals. Simulations demonstrates that using the predictor-model-based plant modeling error for adaptation as opposed to the true plant modeling error is highly effective.

II. Polynomial Approximation

Any sufficiently smooth function $f(x) \in \mathscr{C}$ can be expanded as a Taylor's series about some $x = x_0$

$$f(x) = f(x_0) + \nabla f_x(x_0)(x - x_0) + \frac{1}{2}(x - x_0)^\top \nabla^2 f_x(x_0)(x - x_0) + \cdots \tag{2}$$

Then $f(x)$ can be represented as

$$f(x) = \Theta^{*\top}\Phi(x) + \varepsilon(x) \tag{3}$$

where $\Theta^* \in \mathbb{R}^m \times \mathbb{R}^n$ is a matrix of constant but unknown coefficients, $\Phi(x) \in \mathbb{R}^q$ is a vector of regressors in terms of monomials of x

$$\Phi(x) = \begin{bmatrix} 1 & x_1 & x_2 & \dots & x_p & x_1^2 & x_1x_2 & \dots & x_p^2 & \dots & x_1^q & x_1x_2^{q-1} & \dots & x_p^q \end{bmatrix} \tag{4}$$

and $\varepsilon(x)$ is a function approximation error or truncation error which depends on x.

$f(x)$ is then approximated by a polynomial of qth degree

$$\hat{f}(x) = p_q(x) = \Theta^\top\Phi(x) \tag{5}$$

where $\Theta \in \mathbb{R}^m \times \mathbb{R}^n$ is the estimate of Θ^*.

The coefficients Θ can be computed using various least-squares methods such as the batch least-squares, least-squares gradient method, or recursive least-squares (RLS) method. Note that since $\Theta^\top\Phi(x)$ is an approximation of an unknown function $f(x)$, the approximation error will not be asymptotic regardless whether or not $\Phi(x)$ is a persistently exciting (PE) signal.

The Weierstrass theorem[4] states that given any sufficiently smooth function $f(x) \in \mathscr{C}[a,b]$ and $\varepsilon_0 > 0$, there exist a polynomial $p_q(x)$ for some sufficiently large q such that $\left\|f(x) - p_q(x)\right\|_\infty < \varepsilon_0$. This means that any sufficiently smooth function can be approximated by a polynomial of qth degree. Then the function approximation error could be made sufficiently small on a compact domain of x such that $\sup_{x\in\mathscr{D}} \|\varepsilon(x)\| \le \varepsilon_0$ for all $x \in \mathscr{D} \subset \mathbb{R}^n$.

There are several types of polynomial approximation of a function. The regular polynomial regression using monomials as basis functions is frequently used for function approximation. Orthogonality is a property of a function that belongs to a metric space endowed with an inner product. Given two functions $g(x)$ and $h(x)$, then $g(x)$ and $h(x)$ are orthogonal to each other if their inner product is zero. That is

$$\langle g(x), h(x) \rangle = 0 \tag{6}$$

where $\langle .,. \rangle$ is an inner product operator that takes on two real-valued functions and returns a constant. The inner product has a concrete mathematical definition depending on the class of functions and the type of metric space.

A regular polynomial does not possess this orthogonality property. In contrast, certain classes of polynomials such as Chebyshev polynomials and Legendre polynomials are known as orthogonal polynomials. One advantage of an orthogonal polynomial over a regular polynomial is that an orthogonal polynomial can provide a better function approximation than a regular polynomial of the same degree. Because the Chebyshev orthogonal polynomials are considered as being nearly optimal polynomials[4] that can accurately approximate functions, neural networks based on the Chebyshev orthogonal polynomials have been developed.[16]

The Chebyshev orthogonal polynomials are special polynomial functions that are associated with the solution of a class of Sturm-Liouville differential equations

$$\left(1-x^2\right)\frac{d^2y}{dx^2} - x\frac{dy}{dx} + n^2 y = 0 \tag{7}$$

for $x \in [-1, 1]$.

This differential equation is known as the Chebyshev differential equation of the first kind. The Chebyshev differential equation of the second kind is of the form

$$\left(1-x^2\right)\frac{d^2y}{dx^2} - 3x\frac{dy}{dx} + n(n+2)y = 0 \tag{8}$$

The Chebyshev orthogonal polynomials of the first kind are given by a generating function

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \tag{9}$$

where $T_0(x) = 1$ and $T_1(x) = x$.

The first several terms of the Chebyshev orthogonal polynomials are given as follows:

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_2(x) &= 2x^2 - 1 \\ T_3(x) &= 4x^3 - 3x \\ T_4(x) &= 8x^4 - 8x^2 + 1 \\ &\vdots \end{aligned} \tag{10}$$

A solution of a Sturm-Liouville differential equation constitutes an orthogonal basis that spans any Hilbert space which is a complete, inner product space with a weighted inner product definition

$$\langle g(x), h(x) \rangle = \int_a^b w(x)\, g(x)\, h(x)\, dx \tag{11}$$

The Chebyshev polynomials are orthogonal with respect to a weighting function

$$w(x) = \frac{1}{\sqrt{1-x^2}} \tag{12}$$

such that

$$\langle T_n(x), T_m(x) \rangle = \int_{-1}^{1} \frac{T_n(x)\, T_m(x)}{\sqrt{1-x^2}} dx = \begin{cases} 0 & n \neq m \\ \pi & n = m = 0 \\ \frac{\pi}{2} & n = m \end{cases} \tag{13}$$

Any subspace $\mathscr{S}$ in an inner product space $\mathscr{C}$ has an orthogonal complement $\mathscr{S}^{\perp}$ such that their bases completely span the inner product space $\mathscr{C}$. Therefore

$$\mathscr{C} = \mathscr{S} \oplus \mathscr{S}^{\perp} \tag{14}$$

Since the Chebyshev polynomials are orthogonal, they form a complete basis for an real-valued function. This implies that any unknown, non-singular function $f(x)$ can be approximated by a Chebyshev polynomial of degree n.

$$\hat{f}(x) = \theta_0 T_0(x) + \theta_1 T_1(x) + \cdots + \theta_n T_n(x) = \sum_{i=1}^{n} \theta_i T_i(x) = \Theta^{\top} \Phi(x) \tag{15}$$

where the polynomial coefficients are approximated by

$$\theta_i = \int_{-1}^{1} \frac{f(x) T_i(x)}{\sqrt{1-x^2}} dx \left(\int_{-1}^{1} \frac{T_i^2(x)}{\sqrt{1-x^2}} dx \right)^{-1} \tag{16}$$

for $x \in [-1, 1]$.

For higher dimensions when $x \in \mathbb{R}^n$, multi-dimensional Chebyshev orthogonal polynomials can be obtained.[16]

III. Least-Squares Estimation

A. Least-Squares Gradient Method

The input-output transfer function of a system between $x(t)$ and $y(t)$ can be linearly parametrized as

$$y = \Theta^{*\top} \Phi(x) \tag{17}$$

where $\Theta^* \in \mathbb{R}^m \times \mathbb{R}^n$ is a matrix of constant but unknown coefficients and $\Phi(x) \in \mathbb{R}^m$ is a bounded regressor (or basis function) vector and is assumed to be known.

Let $\hat{y}(t)$ be an estimate of $y(t)$ such that

$$\hat{y} = \Theta^{\top} \Phi(x) \tag{18}$$

where Θ is an estimation of Θ^*.

Formulating an approximation error $\varepsilon(t)$ as

$$\varepsilon = \hat{y} - y = \Theta^{\top} \Phi(x) - y \tag{19}$$

Consider the following cost function

$$J(\Theta) = \frac{1}{2} \varepsilon^{\top} \varepsilon \tag{20}$$

The gradient of the cost function with respect to Θ is given by

$$\frac{\partial J}{\partial \Theta} = \nabla J_{\Theta}(\Theta) = \left(\frac{\partial \varepsilon}{\partial \Theta^{\top}} \right) \varepsilon^{\top} = \Phi(x) \varepsilon^{\top} \tag{21}$$

Now consider the minimization of $J(\Theta)$. Θ^* is said to be a global minimum of J if

$$J(\Theta^*) \leq J(\Theta) \tag{22}$$

This implies that $\nabla J_{\Theta}(\Theta^*) = 0$ and $\nabla^2 J_{\Theta}(\Theta^*) \geq 0$ since $J(\Theta)$ is twice-differentiable with respect to Θ.

Utilizing Taylor's series expansion, one writes

$$\nabla J_{\Theta}(\Theta^*) = \nabla J_{\Theta}(\Theta^* + \Delta\Theta) + \nabla^2 J_{\Theta}(\Theta^* + \Delta\Theta) \Delta\Theta + \underbrace{\mathscr{O}\left(\Delta\Theta^{\top} \Delta\Theta\right)}_{\approx 0} \tag{23}$$

Since $\nabla J_{\Theta}(\Theta^*) = 0$, $\nabla J_{\Theta}(\Theta^* + \Delta\Theta) = \nabla J_{\Theta}(\Theta)$, and $\nabla^2 J_{\Theta}(\Theta^* + \Delta\Theta) = \nabla^2 J_{\Theta}(\Theta)$, then

$$\Delta\Theta = -\left[\nabla^2 J_{\Theta}(\Theta)\right]^{-1} \nabla J_{\Theta}(\Theta) \tag{24}$$

Equation (24) can be written in discrete-time form as

$$\Theta_{i+1} = \Theta_i - \left[\nabla^2 J_\Theta(\Theta_i)\right]^{-1} \nabla J_\Theta(\Theta_i) \tag{25}$$

This is known as a second-order gradient or Newton's method for convex optimization. It is noted that the inverse of the Hessian matrix is generally numerically intensive. So a first-order approximation can be made by recognizing that $\nabla^2 J_\Theta(\Theta) \approx \nabla^2 J_\Theta(\Theta^*) = \varepsilon \geq 0$, where ε is a small positive parameter, when Θ is in the neighborhood of Θ^*. In continuous time, the least-squares gradient method is expressed as

$$\dot{\Theta} = -\Gamma \nabla J_\Theta(\Theta) \tag{26}$$

where $\Gamma = \Gamma^\top > 0 \in \mathbb{R}^m \times \mathbb{R}^m$ is a positive definite adaptation rate matrix.

Returning to the minimization of $J(\Theta)$ to estimate Θ^*, the differential form of the least-squares estimation of Θ can be expressed using the gradient method as

$$\dot{\Theta} = -\Gamma \nabla J_\Theta(\Theta) = -\Gamma \Phi(x) \varepsilon^\top \tag{27}$$

B. Persistent Excitation and Parameter Convergence

Let $\tilde{\Theta}(t) = \Theta(t) - \Theta^*$ be the estimation error, then

$$\varepsilon = \Theta^\top \Phi(x) \Theta - y = \tilde{\Theta}^\top \Phi(x) \tag{28}$$

The least-squares gradient method can be written as

$$\dot{\tilde{\Theta}} = \dot{\Theta} = -\Gamma \Phi(x) \Phi^\top(x) \tilde{\Theta} \tag{29}$$

Now, choose a Lyapunov candidate function

$$V\left(\tilde{\Theta}\right) = \text{trace}\left(\tilde{\Theta}^\top \Gamma^{-1} \tilde{\Theta}\right) \tag{30}$$

Then

$$\dot{V}\left(\tilde{\Theta}\right) = 2\text{trace}\left(\tilde{\Theta}^\top \Gamma^{-1} \dot{\tilde{\Theta}}\right) = -2\text{trace}\left(\tilde{\Theta}^\top \Phi(x) \Phi^\top(x) \tilde{\Theta}\right) = -2\Phi^\top(x) \tilde{\Theta} \tilde{\Theta}^\top \Phi(x) = -2\varepsilon^\top \varepsilon = -2\|\varepsilon\|^2 \leq 0 \tag{31}$$

Note that $\dot{V}\left(\tilde{\Theta}\right)$ can only be negative semi-definite because $\dot{V}\left(\tilde{\Theta}\right)$ can be zero when $\Phi(x) = 0$ independent of $\tilde{\Theta}$. One can establish that $V\left(\tilde{\Theta}\right)$ has a finite limit as $t \to \infty$ since

$$V(t \to \infty) = V(t_0) - 2\int_{t_0}^{\infty} \|\varepsilon\|^2 dt < \infty \tag{32}$$

which implies

$$2\int_{t_0}^{\infty} \|\varepsilon\|^2 dt = V(t_0) - V(t \to \infty) < \infty \tag{33}$$

Therefore, $\varepsilon(t) \in \mathscr{L}_2 \cap \mathscr{L}_\infty$. Moreover, since $\Phi(x) \in \mathscr{L}_\infty$ by the problem statement, then $\tilde{\Theta}(t) \in \mathscr{L}_\infty$, but there is no assurance that $\tilde{\Theta}(t) \to 0$ as $t \to \infty$ which implies parameter convergence.

One cannot conclude that $\dot{V}\left(\tilde{\Theta}\right)$ is uniformly continuous since

$$\ddot{V}\left(\tilde{\Theta}\right) = -4\varepsilon^\top \dot{\varepsilon} = -4\varepsilon^\top \left[\dot{\Theta}^\top \Phi(x) + \Theta^\top \dot{\Phi}(x) - \dot{y}\right] \tag{34}$$

is not necessarily bounded because there is no other condition placed on $\Phi(x)$ except for $\Phi(x) \in \mathscr{L}_\infty$. For $\dot{V}\left(\tilde{\Theta}\right)$ to be uniformly continuous, an additional condition that $\dot{\Phi}(x) \in \mathscr{L}_\infty$ is required. Then, using the Barbalat's lemma, one can conclude that $\dot{V}\left(\tilde{\Theta}\right) \to 0$ or $\varepsilon(t) \to 0$ which also implies that $\dot{\Theta}(t) \to 0$ as $t \to \infty$. Note that from Eq. (29) $\dot{\Theta}(t) \to 0$ does not necessarily imply that $\tilde{\Theta}(t) \to 0$ since $\Phi(x)\Phi^\top(x)$ can also tend to zero instead of $\tilde{\Theta}(t)$.

So, one can only show that the approximation error $\varepsilon(t)$ can tend to zero if $\dot{\Phi}(x) \in \mathscr{L}_\infty$, but not necessarily the estimation error $\tilde{\Theta}(t)$ since $\Phi(x)$ can be a zero signal at some time interval. To examine the issue of parameter convergence, suppose $\Phi(x)$ satisfies the following condition:

$$\frac{1}{T}\int_{t}^{t+T} \Phi(x) \Phi^\top(x) d\tau \geq \alpha I \tag{35}$$

for all $t \geq t_0$, $T > t_0$, and $\alpha > 0$.

Then, Eq. (29) has a solution

$$\tilde{\Theta}(t) = \tilde{\Theta}(t_0) e^{-\Gamma\alpha(t-t_0)} \tag{36}$$

Equation (35 is the well-known persistent excitation (PE) condition which essentially requires an input signal to be persistently exciting (PE), that is, a signal that does not go to zero after some finite time when parameter convergence has not been reached. Another interpretation of the persistent excitation condition is that for parameter identification to converge exponentially, an input signal must be sufficiently rich to excite all system modes associated with the parameters to be identified. It should be noted that while persistent excitation is needed for parameter convergence, in practice, input signals that are persistently exciting can lead to unwanted consequences such as exciting unknown or unmodeled dynamics that can exacerbate stability of a dynamical system.

Another observation to be made is that if $x(t)$ is a state variable of a closed-loop system, one cannot assume that the persistent excitation condition can be satisfied. Suppose $x(t)$ is not determined by a feedback process, then the estimation error is exponentially stable and bounded by

$$\left\|\tilde{\Theta}(t)\right\| \leq \left\|\tilde{\Theta}(t_0)\right\| e^{-\gamma\alpha(t-t_0)}, \forall t \in [t_1, t_1+T], t_1 > t_0 \tag{37}$$

where $\gamma = \lambda_{min}(\Gamma)$ is the smallest eigenvalue of Γ. Thus, $\tilde{\Theta}(t)$ is exponentially stable with $\tilde{\Theta}(t) \to 0$ as $t \to \infty$. Hence, the parameter convergence is established. It follows that the approximation error is also asymptotically stable with $\varepsilon(t) \to 0$ as $t \to \infty$.

C. Recursive Least-Squares

Consider the following cost function

$$J(\Theta) = \frac{1}{2}\int_{t_0}^{t} \varepsilon^\top \varepsilon d\tau \tag{38}$$

which is the continuous-time version of the cost function for batch least-squares.

The necessary condition is

$$\nabla J_\Theta(\Theta) = \frac{\partial J^\top}{\partial \Theta^\top} = \int_{t_0}^{t} \Phi(x)\left[\Phi^\top(x)\Theta - y^\top\right] d\tau = 0 \tag{39}$$

from which $\Theta(t)$ is obtained as

$$\Theta = \left[\int_{t_0}^{t} \Phi(x)\Phi^\top(x) d\tau\right]^{-1} \int_{t_0}^{t} \Phi(x) y^\top d\tau \tag{40}$$

assuming the inverse of $\int_{t_0}^{t} \Phi(x)\Phi^\top(x) d\tau$ exists. Note that the matrix $\Phi(x)\Phi^\top(x)$ is always singular and is not invertible. However, if the PE condition is satisfied, then $\int_{t_0}^{t} \Phi(x)\Phi^\top(x) d\tau$ is invertible.

Introducing a matrix $R(t) = R^\top(t) > 0 \in \mathbb{R}^m \times \mathbb{R}^m$ where

$$R = \left[\int_{t_0}^{t} \Phi(x)\Phi^\top(x) d\tau\right]^{-1} \tag{41}$$

Then

$$R^{-1}\Theta = \int_{t_0}^{t} \Phi(x) y^\top d\tau \tag{42}$$

Upon differentiation, this yields

$$R^{-1}\dot{\Theta} + \frac{dR^{-1}}{dt}\Theta = \Phi(x) y^\top \tag{43}$$

From Eq. (41)

$$\frac{dR^{-1}}{dt} = \Phi(x)\Phi^\top(x) \tag{44}$$

Therefore

$$\dot{\Theta} = -R\Phi(x)\left[\Phi^\top(x)\Theta - y^\top\right] = -R\Phi(x)\varepsilon^\top \tag{45}$$

Now since $RR^{-1} = I$, then

$$\dot{R}R^{-1} + R\frac{dR^{-1}}{dt} = 0 \tag{46}$$

Thus

$$\dot{R} = -R\Phi(x)\Phi^{\top}(x)R \tag{47}$$

Both Eqs. (48) and (47) constitute the well-known recursive least-squares (RLS) parameter identification method.[17] The matrix $R(t)$ is called the covariance matrix and the RLS formula is similar to the Kalman filter where Eq. (47) is a differential Riccati equation for a zero-order plant model. Comparing Eq. (27) with Eq. (48), $R(t)$ plays a role of Γ as a time-varying adaptation rate matrix and Eq. (47) is effectively an adaptive law for the time-varying adaptation rate matrix.

Let $\tilde{\Theta}(t) = \Theta(t) - \Theta^*$ be the estimation error. Since $\varepsilon = \tilde{\Theta}^{\top}\Phi(x)$, then

$$\dot{\tilde{\Theta}} = -R\Phi(x)\Phi^{\top}(x)\tilde{\Theta} \tag{48}$$

Choose a Lyapunov candidate function

$$V\left(\tilde{\Theta}\right) = \text{trace}\left(\tilde{\Theta}^{\top}R^{-1}\tilde{\Theta}\right) \tag{49}$$

Then

$$\begin{aligned}\dot{V}\left(\tilde{\Theta}\right) &= \text{trace}\left(2\tilde{\Theta}^{\top}R^{-1}\dot{\tilde{\Theta}} + \tilde{\Theta}^{\top}\frac{dR^{-1}}{dt}\tilde{\Theta}\right) = \text{trace}\left(-2\tilde{\Theta}^{\top}\Phi(x)\Phi^{\top}(x)\tilde{\Theta} + \tilde{\Theta}^{\top}\Phi(x)\Phi^{\top}(x)\tilde{\Theta}\right) \\ &= -\text{trace}\left(\tilde{\Theta}^{\top}\Phi(x)\Phi^{\top}(x)\tilde{\Theta}\right) = -\varepsilon^{\top}\varepsilon = -\|\varepsilon\|^2 \leq 0\end{aligned} \tag{50}$$

One can establish that $V\left(\tilde{\Theta}\right)$ has a finite limit as $t \to \infty$ since

$$V(t \to \infty) = V(t_0) - \int_{t_0}^{\infty} \|\varepsilon\|^2 dt < \infty \tag{51}$$

Therefore, $\varepsilon(t) \in \mathscr{L}_2 \cap \mathscr{L}_\infty$. Since $\Phi(x) \in \mathscr{L}_\infty$ by the problem statement, then $\tilde{\Theta}(t) \in \mathscr{L}_\infty$, but there is no guarantee that $\tilde{\Theta}(t) \to 0$ as $t \to \infty$ which implies parameter convergence, unless $\Phi(x)$ is PE.

Note that $\dot{V}\left(\tilde{\Theta}\right)$ is not necessarily uniformly continuous since this would require that $\ddot{V}\left(\tilde{\Theta}\right)$ is bounded. Evaluating $\ddot{V}\left(\tilde{\Theta}\right)$ as

$$\begin{aligned}\ddot{V}\left(\tilde{\Theta}\right) &= -2\varepsilon^{\top}\dot{\varepsilon} = -2\varepsilon^{\top}\left[\dot{\tilde{\Theta}}^{\top}\Phi(x) + \tilde{\Theta}^{\top}\dot{\Phi}(x)\right] = -2\varepsilon^{\top}\left[-\tilde{\Theta}^{\top}\Phi(x)\Phi^{\top}(x)R\Phi(x) + \tilde{\Theta}^{\top}\dot{\Phi}(x)\right] \\ &= -2\varepsilon^{\top}\left\{-\tilde{\Theta}^{\top}\Phi(x)\Phi^{\top}(x)\left[\int_{t_0}^{t}\Phi(x)\Phi^{\top}(x)d\tau\right]^{-1}\Phi(x) + \tilde{\Theta}^{\top}\dot{\Phi}(x)\right\}\end{aligned} \tag{52}$$

Therefore, $\ddot{V}\left(\tilde{\Theta}\right)$ is bounded if the following conditions are imposed:

- $\dot{\Phi}(x) \in \mathscr{L}_\infty$
- $\left[\int_{t_0}^{t}\Phi(x)\Phi^{\top}(x)d\tau\right]^{-1}$ is invertible which implies $\Phi(x)$ is PE

If these conditions are satisfied, then using the Barbalat's lemma, it can be shown that $\varepsilon(t) \to 0$ as $t \to \infty$. In addition, $\tilde{\Theta}(t) \to 0$ as $t \to \infty$ and the parameter convergence is achieved.

Comparing the RLS method with the least-squares gradient method, the former seeks to minimize an integral quadratic cost function of the approximation error, whereas the latter minimizes an instantaneous cost function. If an instantaneous cost function is minimized, then its integral cost function is also minimized, but the converse is not necessarily true. Therefore, the least-squares gradient method generally performs better than the RLS method for function approximation.

IV. Least-Squares Model-Reference Adaptive Control with Unstructured Uncertainty

Consider a nonlinear system with match unstructured uncertainty

$$\dot{x} = Ax + B\left[u + f(x)\right] \tag{53}$$

where $x(t) \in \mathbb{R}^n$ is a state vector, $u(t) \in \mathbb{R}^p$ is a control vector with $p \geq n$, $A \in \mathbb{R}^{n\times n}$ is unknown and $B \in \mathbb{R}^{n\times p}$ is known and the pair (A,B) is assumed to be controllable, and $f(x) \in \mathbb{R}^p$ is a matched unstructured uncertainty which can be approximated as

$$f(x) = \Theta^{*\top}\Phi(x) + \delta(x) \tag{54}$$

where $\Theta^* \in \mathbb{R}^{q\times p}$ is a constant unknown weight matrix, $\Phi(x) \in \mathbb{R}^q$ is a vector of orthogonal basis functions, and $\delta(x)$ is a truncation error which, according to the Weierstrass Theorem can be made small on a compact domain of x such that $\sup_{x\in\mathscr{D}} \|\delta(x)\| \leq \delta_0$ for all $x \in \mathscr{D} \subset \mathbb{R}^n$ by increasing the number of orthogonal basis functions. Note that, in general, $\Phi(x)$ can be any bounded regressor function. However, with the choice of Chebyshev orthogonal polynomials, these regressor functions are also true basis functions with their endowed orthogonality properties. Basis functions provide a better approximation of an unstructured uncertainty than non-basis regressor functions.

Alternatively, an unstructured uncertainty can also be approximated by a neural network

$$f(x) = \Theta^{*\top}\Phi\left(W^{*\top}\bar{x}\right) + \delta(x) \tag{55}$$

where $\Theta^* \in \mathbb{R}^{q+1\times p}$ and $W^* \in \mathbb{R}^{n+1}$ are unknown constant ideal weight matrices, $\bar{x} = \begin{bmatrix} 1 & x \end{bmatrix}^\top \in \mathbb{R}^{n+1}$, $\Phi\left(W^{*\top}\bar{x}\right) \in \mathbb{R}^{q+1}$.

A reference model is specified by

$$\dot{x}_m = A_m x_m + B_m r \tag{56}$$

where $x_m(t) \in \mathbb{R}^n$ is a reference state vector, A_m is Hurwitz, and $r(t) \in \mathscr{L}_\infty$ is a bounded command vector.

Note that since $p \geq n$, then B is invertible and is defined by a right pseudo-inverse.

$$B^{-1} = B^\top\left(BB^\top\right)^{-1} \tag{57}$$

To accommodate the uncertainty, an adaptive controller is designed as

$$u = K_x(t)x + K_r r - \Theta^\top(t)\Phi(x) \tag{58}$$

Assuming that there exist K_x^* and K_r that satisfy the following model matching conditions:

$$A_m = A + BK_x^* \tag{59}$$

$$B_m = BK_r \tag{60}$$

If $\hat{A}(t)$ is an estimate of A, then

$$K_x(t) = B^{-1}\left[A_m - \hat{A}(t)\right] \tag{61}$$

Then the closed-loop plant model is given by

$$\dot{x} = \left(A + A_m - \hat{A}\right)x + B_m r + B\bar{u} + B\left(\Theta^{*\top} - \Theta^\top\right)\Phi(x) \tag{62}$$

where

$$\bar{u} = K_x x + K_r r \tag{63}$$

If $\hat{A}(t) \to A$, and $\Theta(t) \to \Theta^*$, then $\dot{x}(t)$ converges to

$$\dot{x}_d = A_m x + B_m r = \hat{A}x + B\bar{u} \tag{64}$$

which follows the reference model if $x(t) \to x_m(t)$.

Let $\tilde{A} = \hat{A} - A$, $\tilde{K}_x = K_x^* - K_x$, and $\tilde{\Theta}(t) = \Theta(t) - \Theta^*$ be the estimation errors. The tracking error equation is defined as

$$\dot{e} = \dot{x}_m - \dot{x} = A_m e - B\tilde{K}_x x + B\tilde{\Theta}^\top\Phi(x) - B\delta(x) \tag{65}$$

Formulating a plant modeling error as

$$\varepsilon = \dot{x}_d - \dot{x} = \tilde{A}x + B\tilde{\Theta}^\top \Phi(x) - B\delta(x) \tag{66}$$

then the tracking error equation can be expressed in terms of the plant modeling error as

$$\dot{e} = \dot{x}_m - \dot{x} = \dot{x}_m - \dot{x}_d + \dot{x}_d - \dot{x} = A_m e + \varepsilon \tag{67}$$

Thus, we see that the tracking error is a manifestation of the modeling error since the tracking error dynamics are influenced by the modeling error. The converse, however, is not true.

The weight update laws for $\hat{A}(t)$ and $\Theta(t)$ can be established by the least-squares gradient adaptive law

$$\dot{\Theta} = -\Gamma_\Theta \Phi(x) \varepsilon^\top B \tag{68}$$

$$\dot{\hat{A}}^\top = -\Gamma_A x \varepsilon^\top \tag{69}$$

or alternatively by the RLS adaptive laws

$$\dot{\Theta} = -R_A \Phi(x) \varepsilon^\top B \tag{70}$$

$$\dot{\hat{A}}^\top = -R_A x \varepsilon^\top \tag{71}$$

$$\dot{R}_\Theta = -\eta R_\Theta \Phi(x) \Phi^\top(x) R_\Theta \tag{72}$$

$$\dot{R}_A = -\eta R_A x x^\top R_A \tag{73}$$

Note that the signal $\dot{x}(t)$ is required in the computation of the modeling error. In many applications, this signal is not necessarily available. Therefore, in these situations, the signal $\dot{x}(t)$ needs to be estimated. One method of estimating $\dot{x}(t)$ is to use a backward finite-difference method[9,10]

$$\dot{\hat{x}}(t) = \frac{x(t) - x(t - \Delta t)}{\Delta t} \tag{74}$$

to estimate $\dot{x}(t)$ at a current time step using the current and past state information.

Another approach is to sample data sufficiently and then use them to construct an at least C^1 smooth time function using a cubic or B-spline method. This curve is then differentiated at the spline knots to find the estimated derivative values. This derivative estimate would tend to result in lower noise than the backward finite-difference method.[9,10] In either case, the derivative computation will introduce a noise source. Least-squares is generally robust to noise, especially if the noise source is a normal Gaussian distribution. Therefore, least-squares adaptive control should be generally more effective than model-reference adaptive control in the presence of noisy measurements. A possibility to suppress noise source is to use a low-pass filter to smooth out the reconstructed signal from numerical differentiation.

An alternative approach of estimating the signal $\dot{x}$ without numerical differentiation is via a prediction method. Consider a predictor model of the plant as

$$\dot{\hat{x}} = A_m \hat{x} + \left(\hat{A} - A_m\right) x + B\left[u + \Theta^\top \Phi(x)\right] \tag{75}$$

If $\Theta(t) \to \Theta^*$, then $\dot{\hat{x}}(t) \to \dot{x}(t)$. Thus, $\dot{\hat{x}}(t)$ is an estimate of $\dot{x}(t)$. The estimate $\dot{\hat{x}}(t)$ only depends on the information of the current state, control, and adaptive parameter. The predictor model thus can be used to provide the estimate of the signal $\dot{x}(t)$ without differentiation.

As the adaptation proceeds with sufficiently rich signals, the predictor model will converge to the plant model with $\hat{x}(t) \to x(t) + e_p(t)$ as $t \to \infty$, where $e_p(t) = \hat{x}(t) - x(t)$ is the predictor error whose dynamics are described by

$$\dot{e}_p = \dot{\hat{x}} - \dot{x} = A_m e_p + \varepsilon \tag{76}$$

The plant modeling error based on the predictor model is established by

$$\varepsilon_p = \dot{x}_d - \dot{\hat{x}} = \dot{x}_d - \dot{x} - \dot{e}_p = -A_m e_p \tag{77}$$

If the plant modeling error based on the predictor model converges to zero; i.e., $\varepsilon_p(t) \to 0$, then the predictor error also converges to zero, $e_p(t) \to 0$. This immediately leads to the convergence of the true plant modeling error to zero; $\varepsilon(t) \to 0$ as $t \to \infty$. Therefore, in lieu of the true plant modeling error, the plant modeling error based on the predictor

model can be used in the least-squares adaptive laws. Thus, the predictor-model-based least-squares gradient adaptive laws are given by

$$\dot{\Theta} = -\Gamma_{\Theta}\Phi(x)\,\varepsilon_p^{\top}B \tag{78}$$

$$\dot{\hat{A}}^{\top} = -\Gamma_A x\varepsilon_p^{\top} \tag{79}$$

and the predictor-model-based RLS adaptive laws are given by

$$\dot{\Theta} = -R_A\Phi(x)\,\varepsilon_p^{\top}B \tag{80}$$

$$\dot{\hat{A}}^{\top} = -R_A x\varepsilon_p^{\top} \tag{81}$$

$$\dot{R}_{\Theta} = -\eta R_{\Theta}\Phi(x)\,\Phi^{\top}(x)\,R_{\Theta} \tag{82}$$

$$\dot{R}_A = -\eta R_A x x^{\top} R_A \tag{83}$$

The least-squares adaptive control can be shown to be stable and result in bounded tracking error for $0 \le \eta < 2$. Toward this end, the estimation error equation for the RLS adaptive laws is expressed as

$$\dot{\tilde{\Omega}} = -R\Psi(x)\left[\Psi^{\top}(x)\,\tilde{\Omega} - \delta^{\top}B^{\top}\right] \tag{84}$$

where $R = \mathrm{diag}\,(R_{\Theta}, R_A)$, $\tilde{\Omega}^{\top} = \begin{bmatrix} \tilde{A} & B\tilde{\Theta}^{\top} \end{bmatrix}$, and $\Psi(x) = \begin{bmatrix} x^{\top} & \Phi^{\top}(x) \end{bmatrix}^{\top}$.

Choose a Lyapunov candidate function

$$V\left(e,\tilde{\Omega}\right) = e^{\top}Pe + \mathrm{trace}\left(\tilde{\Omega}^{\top}R^{-1}\tilde{\Omega}\right) \tag{85}$$

Then $\dot{V}\left(e,\tilde{\Theta}\right)$ is evaluated as

$$\begin{aligned}
\dot{V}\left(e,\tilde{\Omega}\right) &= -e^{\top}Qe + 2e^{\top}P\varepsilon + \mathrm{trace}\left(2\tilde{\Omega}^{\top}R^{-1}\dot{\tilde{\Omega}} + \tilde{\Omega}^{\top}\frac{dR^{-1}}{dt}\tilde{\Omega}\right) \\
&= -e^{\top}Qe + 2e^{\top}P\varepsilon + \mathrm{trace}\left(-2\tilde{\Omega}^{\top}\Psi(x)\left[\Psi^{\top}(x)\,\tilde{\Omega} - \delta^{\top}B^{\top}\right] + \eta\tilde{\Omega}^{\top}\Psi(x)\,\Psi^{\top}(x)\,\tilde{\Omega}\right) \\
&= -e^{\top}Qe + 2e^{\top}PB\left[\tilde{\Omega}^{\top}\Psi(x) - B\delta\right] - \Psi^{\top}(x)\,\tilde{\Omega}\,(2-\eta)\,\tilde{\Omega}^{\top}\Phi(x) + 2\delta^{\top}B^{\top}\tilde{\Omega}^{\top}\Psi(x) \\
&\le -\lambda_{min}(Q)\,\|e\|^2 + 2\,\|e\|\,\|PB\|\,\delta_0 + 2\,\|e\|\,\|PB\|\,\|\Psi(x)\|\,\left\|\tilde{\Omega}\right\| - (2-\eta)\,\|\Psi(x)\|^2\,\left\|\tilde{\Omega}\right\|^2 \\
&\qquad + 2\,\|\Psi(x)\|\,\left\|\tilde{\Omega}\right\|\,\|B\|\,\delta_0
\end{aligned} \tag{86}$$

Let $c_1 = \lambda_{min}(Q)$, $c_2 = \frac{\|PB\|\left(\delta_0 + \|\Psi(x)\|\left\|\tilde{\Omega}\right\|\right)}{c_1}$, $c_3 = 2-\eta$, and $c_4 = \frac{\|B\|\delta_0}{c_3}$. Then, completing the squares yields

$$\dot{V}\left(e,\tilde{\Omega}\right) \le -c_1\left(\|e\| - c_2\right)^2 + c_1c_2^2 - c_3\left[\|\Psi(x)\|\left\|\tilde{\Omega}\right\| - c_4\right]^2 + c_3c_4^2 \tag{87}$$

Therefore, $\dot{V}\left(e,\tilde{\Theta}\right) \le 0$ implies

$$\|e\| \ge c_2 + \sqrt{c_2 + \frac{c_3c_4^2}{c_1}} = \sigma \tag{88}$$

or

$$\|\Psi(x)\|\left\|\tilde{\Omega}\right\| \ge c_4 + \sqrt{c_4 + \frac{c_1c_2^2}{c_3}} = \kappa \tag{89}$$

where c_2 is redefined as $c_2 = \frac{\|PB\|(\delta_0+\kappa)}{c_1}$.

Then, $\|e\|$ is bounded. This implies $\|x\| \le \|x_m\| + \|e\|$ is also bounded. Therefore, $\|\Psi(x)\| \le \Psi_0$ is bounded. Then the system is uniformly ultimately bounded with the following ultimate bounds

$$\|e\| \le \sqrt{\frac{\lambda_{max}(P)\,\sigma^2\Psi_0^2 + \lambda_{max}\left(\Gamma^{-1}\right)\kappa^2}{\lambda_{min}(P)\,\Psi_0^2}} \tag{90}$$

$$\left\|\tilde{\Psi}\right\| \leq \sqrt{\frac{\lambda_{max}(P)\,\sigma^2\Psi_0^2 + \lambda_{max}(\Gamma^{-1})\,\kappa^2}{\lambda_{min}(\Gamma^{-1})\,\Psi_0^2}} \tag{91}$$

Example: Consider a first-order scalar system with unstructured uncertainty

$$\dot{x} = ax + b\left[u + f(x)\right] \tag{92}$$

where a and $f(x)$ are unknown, but $b = 2$. For simulation purpose, $a = 1$ and $f(x) = 0.2\left(\sin 2x + \cos 4x + e^{-x^2}\right)$.

The reference model is given by

$$\dot{x}_m = a_m x_m + b_m r \tag{93}$$

where $a_m = -1$, $b_m = 1$, and $r(t) = \sin t$.

Since $f(x)$ is unknown, a regular polynomial of qth degree is used to approximate $f(x)$ as

$$f(x) = a_0 + a_1 x + \cdots + a_q x^q + \delta(x) \tag{94}$$

where a_i, $i = 0, 1, \ldots, q$ are constant unknown coefficients.

The controller is designed as

$$u = k_x(t)\,x + k_r r - \Theta^\top(t)\,\Phi(x) \tag{95}$$

where $k_x(t)$ and $\Theta(t)$ are computed by the least-squares gradient adaptive laws as

$$\dot{k}_x = \gamma x \varepsilon b \tag{96}$$

$$\dot{\Theta} = -\Gamma\Phi(x)\,\varepsilon b \tag{97}$$

with $k_x(0) = 0$, $\Theta(0) = 0$, $\gamma = 1$, and $\Gamma = I$, where

$$\varepsilon = \hat{a}x + b\bar{u} - \dot{x} \tag{98}$$

$$\hat{a} = a_m - bk_x \tag{99}$$

$$\bar{u} = k_x x + k_r r \tag{100}$$

The tracking error and parameter convergence for $q = 1, 2, 3, 4$ are shown in Figs. 1 and 2. Note that the tracking error improves for $q \geq 2$. Even the tracking error improves, the function $f(x)$ does not seem to be well approximated as shown in Fig. 3. This is also due to the poor convergence of k_x and Θ.

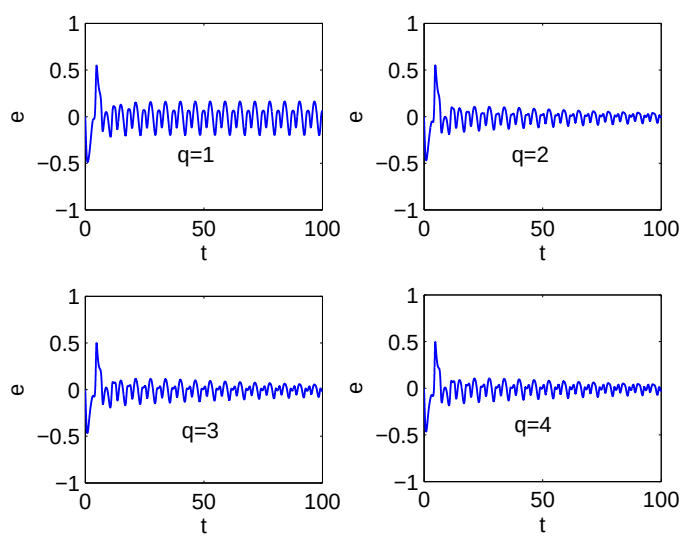

Fig. 1 - Tracking Error due to Least-Squares Gradient Method with Regular Polynomial Approximation

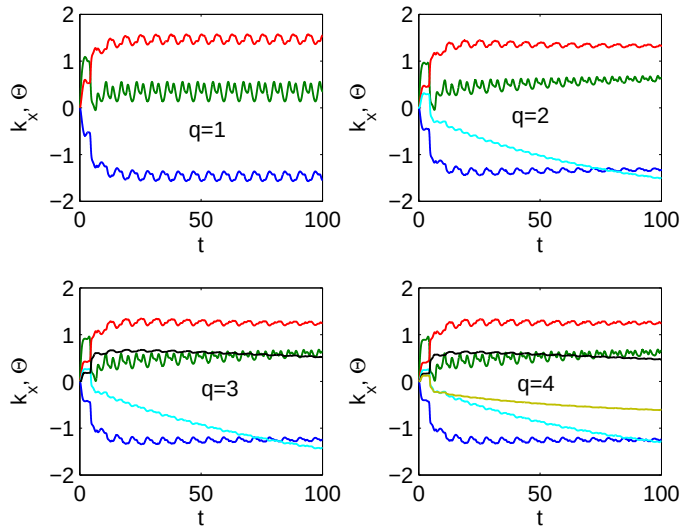

Fig. 2 - k_x and Θ due to Least-Squares Gradient Method with Regular Polynomial Approximation

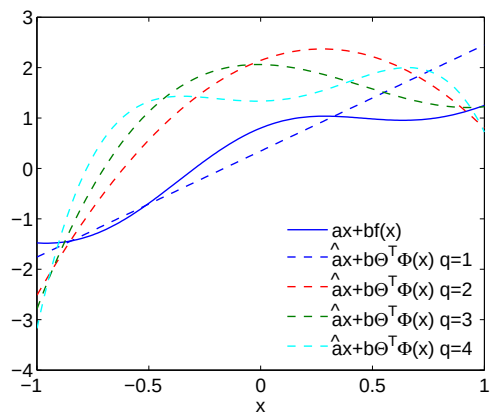

Fig. 3 - Function Approximation at $t = 100$ due to Least-Squares Gradient Method with Regular Polynomial Approximation

Now suppose the Chebyshev orthogonal polynomials are used instead. Then

$$f(x) = a_0 + a_1 T_1(x) + \cdots + a_q T_q(x) + \delta(x) \tag{101}$$

The simulation results are as shown in Figs. 4, 5, and 6. For $q = 1$, the result is the same as the regular polynomial. However, it can be seen that the tracking error significantly reduces for $q = 2$ with the Chebyshev orthogonal polynomial and is even smaller than that for $q = 4$ with the regular polynomial. For $q = 4$, the Chebyshev orthogonal polynomial approximation results in a very small tracking error, as shown in Fig. 4. The unknown function is very well approximated by a 4th-degree Chebyshev orthogonal polynomial, as shown in Fig. 6.

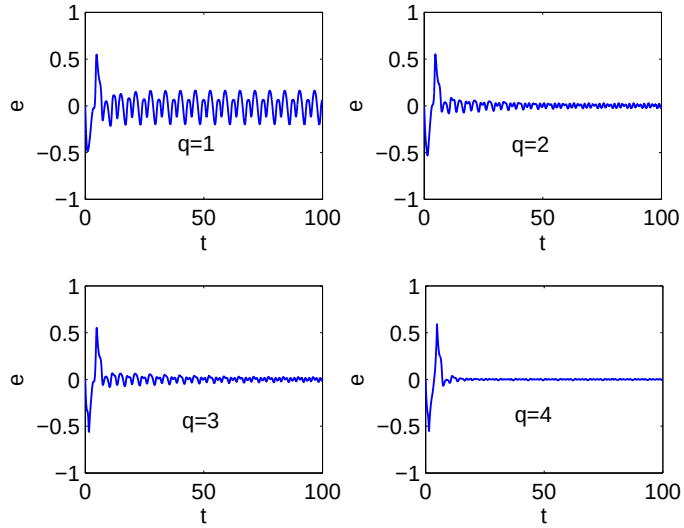

Fig. 4 - Tracking Error due to Least-Squares Gradient Method with Chebyshev Orthogonal Polynomial Approximation

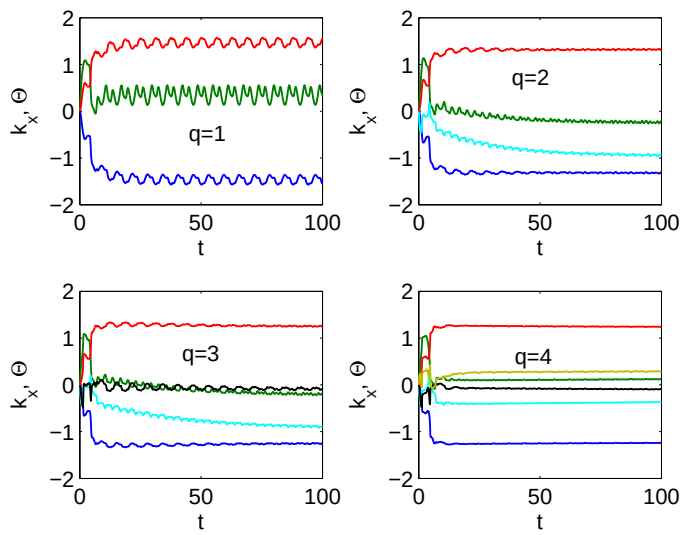

Fig. 5 - k_x and Θ due to Least-Squares Gradient Method with Chebyshev Orthogonal Polynomial Approximation

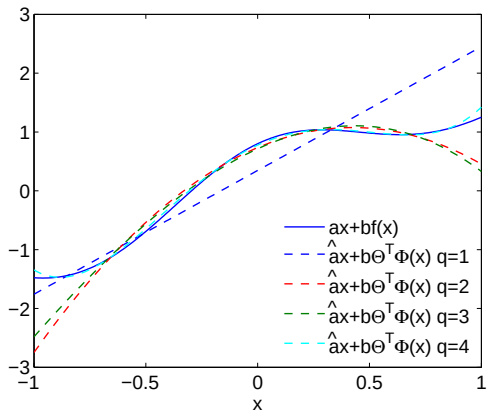

Fig. 6 - Function Approximation at $t = 100$ due to Least-Squares Gradient Method with Chebyshev Orthogonal Polynomial Approximation

In contrast, let us compare the least-squares gradient adaptive control with the standard MRAC using the Chebyshev orthogonal polynomial basis functions. The MRAC update laws are given by

$$\dot{k}_x = \gamma x e b \tag{102}$$

$$\dot{\Theta} = -\Gamma \Phi(x) e b \tag{103}$$

where $e = x_m - x$.

Even though the Chebyshev orthogonal polynomial basis functions are used, the tracking error with the standard MRAC is poorer than that with the least-squares gradient adaptive control, as shown in Fig. 7. Figure 8 illustrates parameter convergence of MRAC. The parameters $k(t)$ and $\Theta(t)$ are more oscillatory. The function approximation by the standard MRAC is less accurate than that by the least-squares gradient adaptive control, as shown in Fig. 9. Furthermore, for systems with unstructured uncertainty, MRAC is known to be non-robust since the parameter estimation error is not necessarily bounded.[17] Therefore, robust modification or a projection method must be used to ensure that the parameter estimation error is bounded.[17]

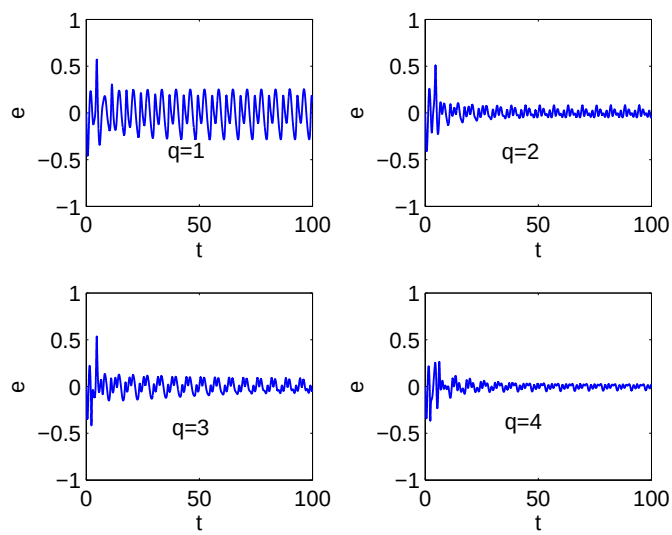

Fig. 7 - Tracking Error due to MRAC with Chebyshev Polynomial Approximation

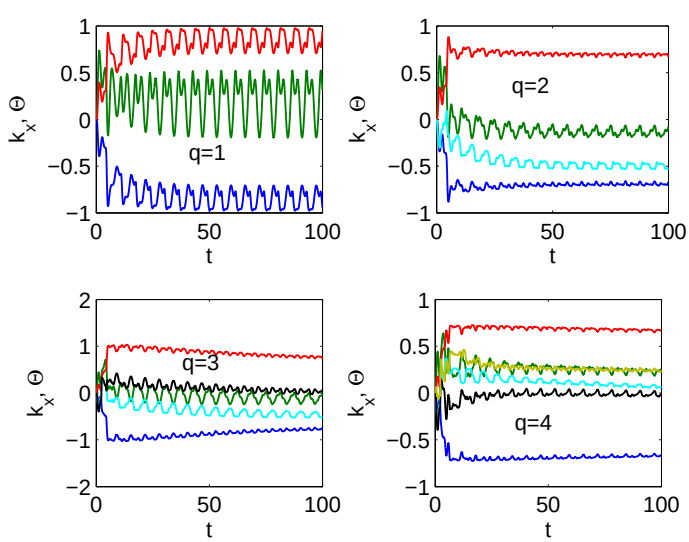

Fig. 8 - k_x and Θ due to MRAC with Chebyshev Polynomial Approximation

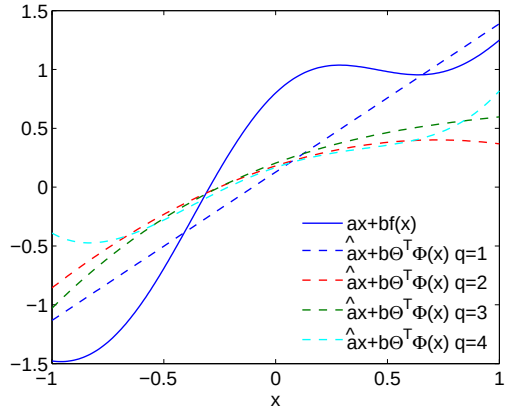

Fig. 9 - Function Approximation at $t = 100$ due to MRAC with Chebyshev Polynomial Approximation

V. Flight Control Application

To demonstrate the effectiveness of the least-squares adaptive control, an adaptive flight control design for pitch attitude command is developed for the NASA Generic Transport Model (GTM), which is a notional medium-range, twin-engine transport aircraft originally developed by NASA Langley Research Center.[21]

Consider the short-period dynamics of an aircraft with unstructured uncertainty $f(\alpha)$ as a function of the angle of attack due to nonlinear aerodynamics

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\theta} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \frac{Z_\alpha}{\bar{u}} & 0 & 1 \\ 0 & 0 & 1 \\ M_\alpha + \frac{M_{\dot{\alpha}} Z_\alpha}{\bar{u}} & 0 & M_q + M_{\dot{\alpha}} \end{bmatrix} \begin{bmatrix} \alpha \\ \theta \\ q \end{bmatrix} + \begin{bmatrix} \frac{Z_{\delta_e}}{\bar{u}} \\ 0 \\ M_{\delta_e} + \frac{M_{\dot{\alpha}} Z_{\delta_e}}{\bar{u}} \end{bmatrix} [\delta_e + f(\alpha)] \tag{104}$$

where Z_α, Z_{δ_e}, M_α, $M_{\dot{\alpha}}$, M_q, and M_{δ_e} are stability and control derivatives; and $\bar{u}$ is the trim airspeed.

A pitch attitude controller is designed to track a desired second-order pitch attitude dynamics according to

$$\ddot{\theta}_m + 2\zeta\omega_n\dot{\theta}_m + \omega_n^2\theta_m = \omega_n^2\theta_c \tag{105}$$

where $\omega_n = 1.5$ rad/sec and $\zeta = 0.85$ are the desired natural frequency and damping ratio of the pitch attitude response, and θ_c is the pitch attitude command..

The pitch rate equation is written as

$$\ddot{\theta} - \left(M_\alpha + \frac{M_{\dot{\alpha}} Z_\alpha}{\bar{u}}\right)\alpha - (M_q + M_{\dot{\alpha}})\dot{\theta} = \left(M_{\delta_e} + \frac{M_{\dot{\alpha}} Z_{\delta_e}}{\bar{u}}\right)[\delta_e + f(\alpha)] \tag{106}$$

The elevator input is designed with the following proportional-derivative (PD) control law

$$\delta_e = -k_\alpha\alpha - k_\theta(\theta - \theta_c) - k_q q - \Theta^\top\Phi(\alpha) = -K_x x + k_\theta\theta_c - \Theta^\top\Phi(\alpha) \tag{107}$$

where $x = \begin{bmatrix} \alpha & \theta & q \end{bmatrix}^\top$, $K_x = \begin{bmatrix} k_\alpha & k_\theta & k_q \end{bmatrix}^\top$, and

$$k_\alpha = \frac{M_\alpha + \frac{M_{\dot{\alpha}} Z_\alpha}{\bar{u}}}{M_{\delta_e} + \frac{M_{\dot{\alpha}} Z_{\delta_e}}{\bar{u}}} \tag{108}$$

$$k_\theta = \frac{\omega_n^2}{M_{\delta_e} + \frac{M_{\dot{\alpha}} Z_{\delta_e}}{\bar{u}}} \tag{109}$$

$$k_q = \frac{2\zeta\omega_n + M_q + M_{\dot{\alpha}}}{M_{\delta_e} + \frac{M_{\dot{\alpha}} Z_{\delta_e}}{\bar{u}}} \tag{110}$$

The numerical model of the short-period dynamics for the GTM at Mach 0.8 and 30,000 ft is given by

$$\underbrace{\begin{bmatrix} \dot{\alpha} \\ \dot{\theta} \\ \dot{q} \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} -0.7018 & 0 & 0.9761 \\ 0 & 0 & 1 \\ -2.6923 & 0 & -0.7322 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \alpha \\ \theta \\ q \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} -0.0573 \\ 0 \\ -3.5352 \end{bmatrix}}_{B} \left[\underbrace{\delta_e}_{u} + f(\alpha) \right]$$

For simulation purpose, the unstructured uncertainty that represents nonlinear aerodynamics is described by

$$f(\alpha) = 0.1\cos\alpha^3 - 0.2\sin 10\alpha - 0.05e^{-\alpha^2} \tag{111}$$

The feedback gain is computed to be $K_x = \begin{bmatrix} 0.7616 & -0.6365 & -0.5142 \end{bmatrix}^\top$. The nominal closed-loop plant is then chosen to be the reference model as

$$\underbrace{\begin{bmatrix} \dot{\alpha}_m \\ \dot{\theta}_m \\ \dot{q}_m \end{bmatrix}}_{\dot{x}_m} = \underbrace{\begin{bmatrix} -0.6582 & -0.0365 & 0.9466 \\ 0 & 0 & 1 \\ 0 & -2.2500 & -2.5500 \end{bmatrix}}_{A_m} \underbrace{\begin{bmatrix} \alpha_m \\ \theta_m \\ q_m \end{bmatrix}}_{x_m} + \underbrace{\begin{bmatrix} 0.0365 \\ 0 \\ 2.2500 \end{bmatrix}}_{B_m} r$$

The plant modeling error is computed as $\varepsilon = \dot{x}_d - \dot{x} = A_m x + B_m r - \dot{x}$ where $\dot{x}$ is estimated by a backward finite-difference method. The uncertainty is modeled with the first four terms of the Chebyshev orthogonal polynomials

$$\Theta^\top \Phi(\alpha) = \theta_0 + \theta_1\alpha + \theta_2\left(2\alpha^2 - 1\right) + \theta_3\left(4\alpha^3 - 3\alpha\right) \tag{112}$$

The least-squares gradient adaptive control and RLS adaptive control are implemented with $\eta = 0.2$ and $\Gamma = R(0) = 10I$. The aircraft longitudinal responses with the nominal controller and with the least-squares gradient and RLS adaptive controllers are as shown in Figs. 10, 11, and 12.

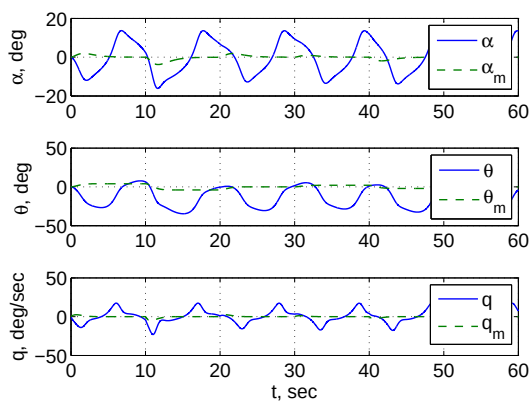

Fig. 10 - Aircraft Response with Nominal Controller

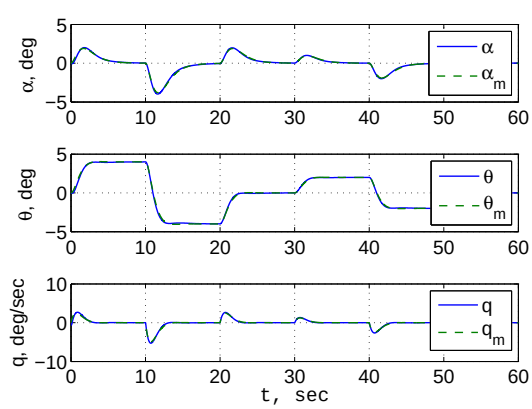

Fig. 11 - Aircraft Response with Least-Squares Gradient Adaptive Control

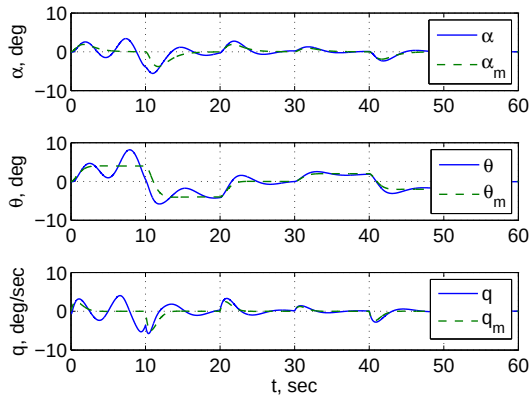

Fig. 12 - Aircraft Response with RLS Adaptive Control ($\eta = 0.2$)

The aircraft response with the nominal controller is seen to be extremely poor with the maximum angle of attack of almost 14° which is well into stall. The benefit of the least-squares gradient adaptive control is clearly demonstrated by the results which show a very good tracking performance. However, the RLS adaptive control does not perform as well. On the other hand, the slow parameter convergence of the RLS adaptive control can improve stability robustness of adaptive control in the presence of time delay or unmodeled dynamics, as will be shown later.

Figures 13 and 14 illustrate the use of the predictor-model-based plant modeling error $\varepsilon_p(t)$ in the least-squares gradient and RLS adaptive control as opposed to the true plant modeling error $\varepsilon(t)$. Both the predictormodel–based least-squares gradient adaptive control and the RLS adaptive control provide very good tracking performance as compared to their counterparts using the true plant modeling error. The responses exhibit some initial transients, but these transients subside rapidly. In particular, the predictor-based RLS adaptive control is observed to provide better tracking than the original RLS adaptive control. Thus, the results indicate that the predictor model can be a viable method for estimating the signal $\dot{x}(t)$ without differentiation which can introduce noise.

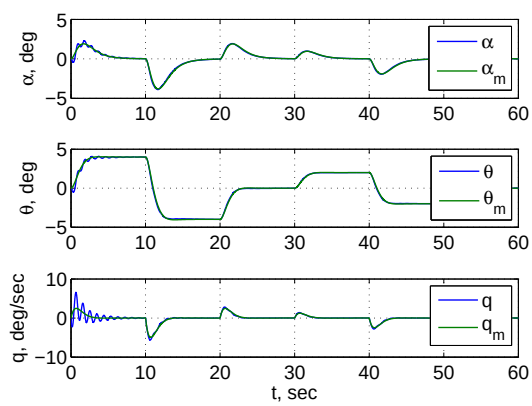

Fig. 13 - Aircraft Response with Least-Squares Gradient Adaptive Control Based on Predictor Model

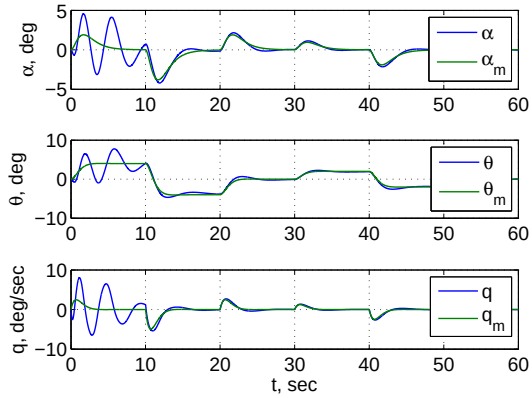

Fig. 14 - Aircraft Response with RLS Adaptive Control ($\eta = 0.2$) Based on Predictor Model

For comparison, the least-squares adaptive control is replaced by the standard MRAC using the same Chebyshev orthogonal polynomials according to

$$\dot{\Theta} = -\Gamma \Phi(\alpha) e^{\top} PB \tag{113}$$

where $e = x_m - x$.

In addition, instead of using the Chebyshev orthogonal polynomials, a two-layer neural network with a sigmoidal activation function is used to approximate the unstructured uncertainty as

$$f(\alpha) = \Theta^{\top} \sigma\left(W^{\top} \bar{\alpha}\right) - \varepsilon(\alpha) \tag{114}$$

where $\bar{\alpha} = \begin{bmatrix} 1 & \alpha \end{bmatrix}^{\top}$ and Φ is the sigmoidal activation function.

The neural network adaptive control is specified by the following adaptive laws

$$\dot{\Theta} = -\Gamma_{\Theta} \Phi\left(W^{\top} \bar{\alpha}\right) e^{\top} PB \tag{115}$$

$$\dot{W} = -\Gamma_W \bar{\alpha} e^{\top} PBV^{\top} \sigma'\left(W^{\top} \bar{\alpha}\right) \tag{116}$$

where $\Theta^{\top} = \begin{bmatrix} V_0 & V^{\top} \end{bmatrix}$.

The aircraft responses with MRAC ($\Gamma = \Gamma_{\Theta} = \Gamma_W = 10I$) using the Chebyshev orthogonal polynomial and the neural network are as shown in Figures 15, 16, and 17.

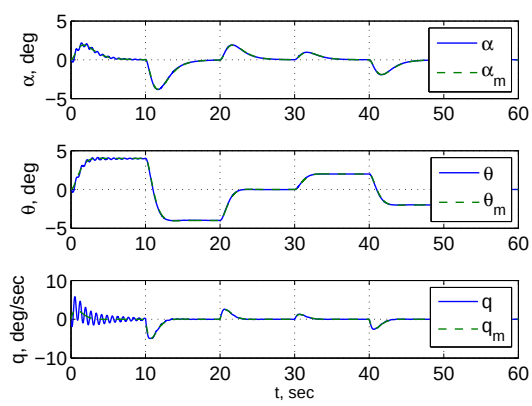

Fig. 15 - Aircraft Response with MRAC

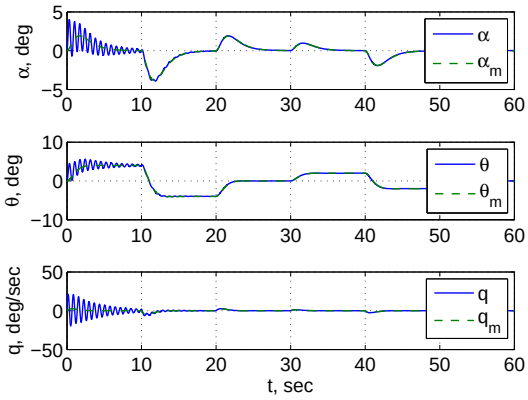

Fig. 16 - Aircraft Response with Neural Network MRAC without Control Saturation

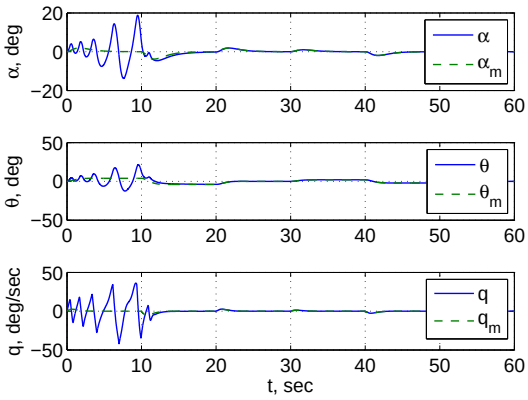

Fig. 17 - Aircraft Response with Neural Network MRAC with Control Saturation

The responses in Figs. 15 and 16 both exhibit initial high frequency oscillations which are indicative of incipient instability, even though the subsequent tracking performance is very good. The neural network MRAC has much more pronounced high frequency oscillations which are due to the weights initialization with random numbers. As a result, a saturation of the elevator is encountered. The aircraft responses due to the neural network MRAC with and without control saturation are as shown in Figures 16 and 17. It is observed that, in the presence of control saturation, the neural network MRAC produces a large angle of attack that exceeds its typical stall limit beyond which the aerodynamic effects can be highly nonlinear and unsteady.

The elevator commands for all four adaptive controllers are shown in Figures 18.

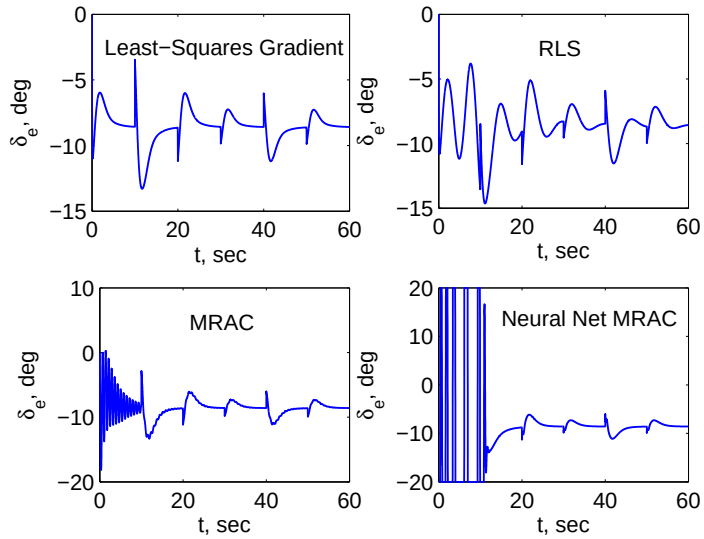

Fig. 18 - Elevator Commands

To illustrate the issue of robustness and show that the RLS adaptive control is actually better able to handle time delay or unmodeled dynamics than the least-squares gradient adaptive control or MRAC, a numerical evidence of the time delay margin is computed for each of the four adaptive controllers. The results are shown in the following table:

Adaptive Law	Numerical Evidence of Time Delay Margin
Least-Squares Gradient	60 ms
RLS with $\eta = 0.2$	260 ms
MRAC	10 ms
Neural Network MRAC (no Saturation)	60 ms

Table 1 - Estimated Time Delay Margins

The RLS adaptive control has the best time delay margin than the other three adaptive controllers. The standard MRAC has very poor robustness which is a well-known fact.[19] Generally, the standard MRAC has to be modified to improve robustness using the projection method or various modification techniques, such as the σ-modification,[22] e-modification,[23] optimal control modification,[24,25] and adaptive loop recovery.[26]

The aircraft responses due to a 60-ms time delay for the least-squares gradient, RLS, and neural network MRAC adaptive control are illustrated in Figures 19, 20, and 21. The aircraft response due to a 10-ms time delay for the MRAC is plotted in Figure 22. As can be seen, the least-squares gradient adaptive control maintains a very good tracking performance even with a 60-ms time delay, but high frequency transients begin to appear. Both the MRAC and neural network MRAC exhibit high frequency oscillations. The RLS adaptive control exhibits low frequency transients as it is much more robust than the other three adaptive controllers. Thus, if the time delay is not too large, the least-squares gradient adaptive control seems to perform the best among all of the adaptive controllers.

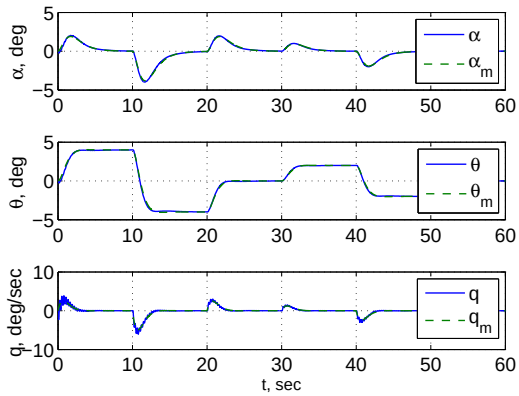

Fig. 19 - Aircraft Response with Least-Squares Gradient with 60-ms Time Delay

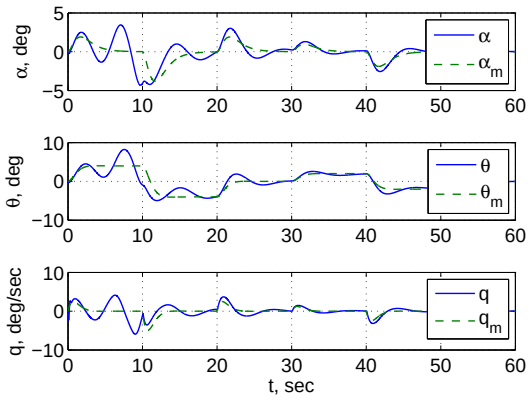

Fig. 20 - Aircraft Response with RLS with 60-ms Time Delay

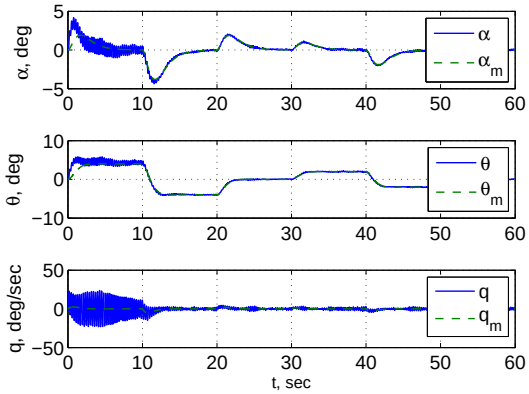

Fig. 21 - Aircraft Response with Neural Network MRAC with 60-ms Time Delay without Control Saturation

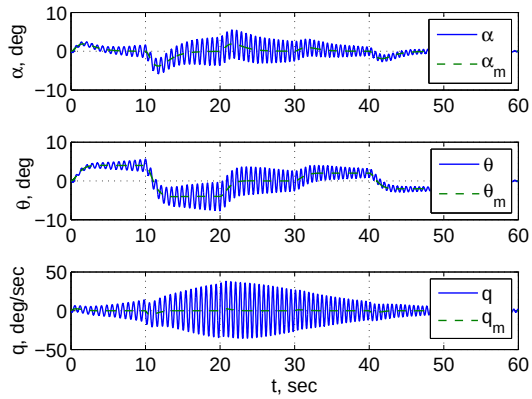

Fig. 22 - Aircraft Response with MRAC with 10-ms Time Delay

VI. Discussion

Modeling error in dynamical systems usually arises from the presence of system uncertainty. In model-reference adaptive control, tracking error is exclusively used as a control signal to drive the adaptation so as to cancel out undesired effects of system uncertainty on closed-loop systems. Philosophically, tracking error is a manifestation of modeling error which in turn is a direct consequence of system uncertainty. Thus, tracking error is a direct consequence of modeling error, but not vice versa. One can separate control and system identification or learning as two related dynamical actions. A control action is generally designed to target the tracking error to ensure that a closed-loop system follows a reference model as closely as possible. The objective of adaptive control is not so much to estimate the uncertainty itself but rather to achieve a desired reference model, regardless whether or not parameter convergence is achieved. The example in Section 4 clearly demonstrates that model-reference adaptive control does a rather poor job on parameter convergence even though the basis functions are Chebyshev polynomials. On the other hand, a learning action or system identification uses the modeling error as a learning signal to estimate the uncertainty directly by least-squares estimation method. The goal of a learning action is first and foremost to achieve parameter convergence, regardless whether or not a desired reference model is achieved.

System identification is usually performed in an open-loop process without directly influencing a control action by feeding the parameter estimates back into a control law. In this study and others, a learning action and control can be combined to provide a very effective adaptation strategy. The learning action can achieve better parameter convergence than model-reference adaptive control by using the modeling error as a learning signal. Using the parameter estimates from learning, least-squares adaptive control can be formulated to cancel out effects of system uncertainty by feedback of the parameter estimates into the control law. Thus, in some way, least-squares adaptive control resembles the separation principle of estimation and control in control theory.

The standard model-reference adaptive control is well-known to be non-robust in the presence of time delay, unmodeled dynamics, and disturbances.[19] Least-squares adaptive control, by the virtue of optimal estimation, is inherently more robust than the standard model-reference adaptive control because of the damping mechanism in the adaptive parameter that naturally arises from the optimal control formulation. To elucidate this point, we can express the least-squares gradient adaptive law (61) as

$$\dot{\Theta} = -\Gamma\Phi(x)\left[\Phi^{\top}(x)\Theta - f^{\top}(x)\right]B^{\top}B \tag{117}$$

One sees that the first term effectively provides a damping mechanism to bound the adaptive parameter $\Theta(t)$. This is very much similar to many robust modification techniques in adaptive control. One such modification, the optimal control modification,[24,25] has a similar damping term as follows:

$$\dot{\Theta} = -\Gamma\Phi(x)\left[-\nu\Phi^{\top}(x)\Theta B^{\top}PA_m^{-1}B + e^{\top}PB\right] \tag{118}$$

where $\nu > 0$ and $PA_m^{-1} < 0$.

Thus, this observation provides a plausible explanation of the improved robustness of least squares adaptive control over the standard model reference adaptive control.

VII. Conclusions

This paper presents an adaptive control method for systems with unstructured uncertainty based on a least-squares gradient method and recursive least-squares method. The adaptive control uses Chebyshev orthogonal polynomials as basis functions to approximate the unstructured uncertainty. The Chebyshev orthogonal polynomials have many desirable features in function approximation and can be shown to be the "best" polynomial function approximation. The least-squares adaptation is driven by a plant modeling error as opposed to the standard tracking error in model-reference adaptive control. Simulations demonstrate the superior performance of Chebyshev orthogonal polynomials in the adaptive control setting over regular polynomials and perhaps even neural networks. The least-squares gradient adaptive control demonstrates to outperform both the recursive least-squares adaptive control and the standard model-reference adaptive control. On the other hand, recursive least-squares adaptive control is shown to be much more robust to time delay and unmodeled dynamics than all the other adaptive control methods being studied. However, this robustness comes at an expense of tracking performance. Thus, in practice, the least-squares gradient method may strive a better balance between tracking performance and robustness.

References

[1]Cybenko, G., "Approximation by Superpositions of a Sigmoidal Function," Mathematics of Control Signals Systems, Vol. 2, pp. 303-314, 1989.

[2]Micchelli, C., "Interpolation of Scattered Data: Distance Matrices and Conditionally Positive Definite Functions," Constructive Approximation, Vol.2, pp. 11-12, 1986.

[3]Moody, J., "The Effective Number of Parameters: An Analysis of Generalization and Regularization in Nonlinear Learning Systems," Advances in Neural Information Processing Systems 4, Morgan Kaufmann Publishers, San Mateo, CA, pp. 847-854, 1992.

[4]Mason, J. and Handscomb, D., *Chebyshev Polynomials*, Chapman and Hall/CRC, 2002.

[5]Lai, T. and Wei, C.-Z. , "Extended Least Squares and Their Applications to Adaptive Control and Prediction in Linear Systems," IEEE Transactions on Automatic Control, Vol. 31, No. 10, pp. 898- 906, 1986.

[6]Suykens, J., Vandewalle, J., and deMoor, B., *Artificial Neural Networks for Modelling and Control of Non-Linear Systems*, Kluwer Academic Publisher, 1996.

[7]Guo, L. , "Self-Convergence of Weighted Least-Squares with Applications to Stochastic Adaptive Control," IEEE Transactions on Automatic Control, Vol. 41, No. 1, pp. 79-89, 1996.

[8]Nguyen, N., Burken, J., Ishihara, A., "Least-Squares Adaptive Control Using Chevbyshev Orthogonal Polynomials," AIAA Infotech@Aerospace Conference, AIAA-2011-1402 , March 2011.

[9]Nguyen, N., Krishnakumar, K., Kaneshige, J., and Nespeca, P., "Dynamics and Adaptive Control for Stability Recovery of Damaged Asymmetric Aircraft," AIAA Guidance, Navigation, and Control Conference, AIAA-2006-6049, 2006.

[10]Nguyen, N., Krishnakumar, K., Kaneshige, J., and Nespeca, P., "Flight Dynamics and Hybrid Adaptive Control of Damaged Aircraft", AIAA Journal of Guidance, Control, and Dynamics, Vol. 31, No. 3, pp. 751-764, 2008.

[11]Nguyen, N., "Hybrid Adaptive Flight Control with Model Inversion Adaptation," *Advances in Flight Control Systems*, ISBN: 978-953-307-218-0, Intech Publishing, 2011.

[12]Zou, A., Kumar, K., and Hou, Z., "Attitude Control of Spacecraft Using Chebyshev Neural Networks," IEEE Transactions on Neural Networks, Vol. 21, No. 9, pp. 1457-1471, 2010.

[13]Chowdhary, G. and Johnson, E., "Least Squares Based modification for Adaptive Control," 49th IEEE Conference on Decision and Control, pp.1767-1772, 2010.

[14]Chowdhary, G. and Johnson, E. , "Recursively Updated Least Squares Based Modification Term for Adaptive Control," American Control Conference (ACC), pp.892-897, 2010.

[15]Chowdhary, G. and Johnson, E. , "Theory and Flight-Test Validation of a Concurrent-Learning Adaptive Controller," Journal of Guidance, Control, and Dynamics, Vol. 34, No. 2, pp. 592-607, 2011.

[16]Lee, T. and Jeng, J., "The Chebyshev-Polynomials-Based Unified Model Neural Networks for Function Approximation," IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, Vol. 28, No. 6, pp. 925-935, 1998.

[17]Ioannu, P. and Sun, J., *Robust Adaptive Control*, Prentice-Hall, 1996.

[18]Nguyen, N., "Bi-Objective Optimal Control Modification Adaptive Control for Systems with Input Uncertainty," AIAA Guidance, Navigation, and Control Conference, AIAA-2012-4615 , 2012.

[19]Rohrs, C., Valavani, L., Athans, M., and Stein, G., "Robustness of Continuous-Time Adaptive Control Algorithms in the Presence of Unmodeled Dynamics," IEEE Transactions on Automatic Control, Vol AC-30, No. 9, pp. 881-889, 1985.

[20]Bosworth, J. and Williams-Hayes, P., "Flight Test Results from the NF-15B IFCS Project with Adaptation to a Simulated Stabilator Failure," AIAA Infotech@Aerospace Conference, AIAA-2007-2818, 2007.

[21]Jordan, T., Langford, W., Belcastro, C., Foster, J., Shah, G., Howland, G., and Kidd, R., "Development of a Dynamically Scaled Generic Transport Model Testbed for Flight Research Experiments," AUVSI Unmanned Unlimited, Arlington, VA, 2004.

[22]Ioannou, P. and Kokotovic, P., "Instability Analysis and Improvement of Robustness of Adaptive Control," Automatica, Vol. 20, No. 5, pp. 583-594, 1984.

[23]Narendra, K. and Annaswamy, A., "A New Adaptive Law for Robust Adaptation Without Persistent Excitation," IEEE Transactions on Automatic Control, Vol. AC-32, No. 2, pp. 134-145, 1987.

[24]Nguyen, N., Krishnakumar, K., and Boskovic, J., "An Optimal Control Modification to Model-Reference Adaptive Control for Fast Adaptation", AIAA Guidance, Navigation, and Control Conference, AIAA 2008-7283, 2008.

[25]Nguyen, N., "Optimal Control Modification for Robust Adaptive Control with Large Adaptive Gain," *Systems & Control Letters*, 61 (2012) pp. 485-494.

[26]Calise, A., Yucelen, T., Muse, J., and Yang, B., "A Loop Recovery Method for Adaptive Control," AIAA Guidance, Navigation, and Control Conference, AIAA-2009-5967, August 2009.