Least-Squares Adaptive Control Using Chebyshev Orthogonal Polynomials

Nhan Nguyen*
NASA Ames Research Center, Moffett Field, CA 94035
John Burken†
NASA Dryden Flight Research Center, Edwards, CA 93523
Abraham Ishihara‡
Carnegie Mellon University Silicon Valley, Moffett Field, CA 94035

This paper presents a new adaptive control approach using Chebyshev orthogonal polynomials as basis functions in a least-squares functional approximation. The use of orthogonal basis functions improves the function approximation significantly and enables better convergence of parameter estimates. Flight control simulations demonstrate the effectiveness of the proposed adaptive control approach.

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 theorem for sigmoidal neural networks by Cybenko[1] and the Micchelli's theorem[2] for radial basis functions 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. The Chebyshev polynomials have been shown to provide the "best" approximation of a function over any other types of polynomials.[4] The use of the Chebyshev polynomials in the context of adaptive control with unstructured uncertainty is demonstrated in this paper. Simulation results demonstrate a significant improvement in the effectiveness of Chebyshev polynomials in the adaptive control setting over a regular polynomial regression.

*AIAA Associate Fellow, nhan.t.nguyen@nasa.gov
†AIAA Member, john.burken@nasa.gov
‡abe.ishihara@west.cmu.edu

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 which depends on x.

$f(x)$ is then approximated by a polynomial of q-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 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 persistently exciting.

The Weierstrass's 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 q-th degree. Then the function approximation error could be made sufficiently small on a compact domain of x such that $\sup_{x\in\mathscr{D}} \|\varepsilon(x)\| \leq \varepsilon_0$ 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 with a constant. The inner product has a concrete mathematical definition depending on the class of functions.

A regular polynomial does not possess this orthogonality property. In contrast, certain classes of polynomials such as Chebyshev polynomials and Legendre polynomials are orthogonal polynomials. One advantage of an orthogonal polynomial over a regular polynomial is that a low-degree orthogonal polynomial can provide a good function approximation with minimal loss of accuracy as compared to a higher-degree regular polynomial.

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

III. Least-Squares Estimation

The input-output transfer function of a system is given by a set of measurement data of $y(t) \in \mathbb{R}^n$ as a function of some independent variable $x(t) \in \mathbb{R}^p$, expressed as data pairs (x_i, y_i), $i = 1, 2, \ldots, N$. Furthermore, suppose the transfer function 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

$$J(\Theta) = \frac{1}{2}\sum_{i=1}^{N} \varepsilon_i^\top \varepsilon_i \tag{20}$$

When $J(\Theta)$ is minimized, the approximation error is also minimized. Then $\hat{y}$ approximates y in a least-squares sense. Thus, the parameter identification problem is posed as a functional minimization.

The necessary condition is given by

$$\frac{\partial J}{\partial \Theta} = \nabla J_\Theta(\Theta) = \sum_{i=1}^{N} \frac{\partial \varepsilon_i}{\partial \Theta^\top} \varepsilon_i^\top = \sum_{i=1}^{N} \Phi(x_i)\left[\Phi^\top(x_i)\Theta - y_i^\top\right] = 0 \tag{21}$$

where ∇J_Θ is called a gradient of J with respect to Θ.

Thus, Θ can be found by solving the following least-squares regression equation

$$\Theta = A^{-1}B \tag{22}$$

where

$$A = \sum_{i=1}^{N} \Phi(x_i)\Phi^\top(x_i) \tag{23}$$

$$B = \sum_{i=1}^{N} \Phi(x_i) y_i^\top \tag{24}$$

and A^{-1} is assumed to exist.

<u>Example:</u> Suppose $y \in \mathbb{R}$ is a scalar variable which can be approximated as a p-th degree polynomial in terms of $x \in \mathbb{R}$ as

$$y = \theta_0 + \theta_1 x + \cdots + \theta_p x^p = \sum_{j=0}^{p} \theta_j x^j = \Theta^\top \Phi(x)$$

where $\Theta^\top = \begin{bmatrix} \theta_0 & \theta_1 & \dots & \theta_p \end{bmatrix}$ and $\Phi(x) = \begin{bmatrix} 1 & x & \dots & x^p \end{bmatrix}^\top$.

The least-squares regression equation is expressed as

$$A\Theta = B \tag{25}$$

where

$$A = \sum_{i=1}^{N} \Phi(x_i)\Phi^\top(x_i) = \begin{bmatrix} \sum_{i=1}^{N} 1 \\ \sum_{i=1}^{N} x_i \\ \vdots \\ \sum_{i=1}^{N} x_i^p \end{bmatrix} \begin{bmatrix} 1 & x_i & \cdots & x_i^p \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} 1 & \sum_{i=1}^{N} x_i & \cdots & \sum_{i=1}^{N} x_i^p \\ \sum_{i=1}^{N} x_i & \sum_{i=1}^{N} x_i^2 & \cdots & \sum_{i=1}^{N} x_i^{p+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{N} x_i^p & \sum_{i=1}^{N} x_i^{p+1} & \cdots & \sum_{i=1}^{N} x_i^{2p} \end{bmatrix} = \left\{ \sum_{i=1}^{N} x_i^{j+k} \right\}_{jk} \tag{26}$$

$$B = \sum_{i=1}^{N} \Phi(x_i) y_i^\top = \begin{bmatrix} \sum_{i=1}^{N} 1 \\ \sum_{i=1}^{N} x_i \\ \vdots \\ \sum_{i=1}^{N} x_i^p \end{bmatrix} y_i = \begin{bmatrix} \sum_{i=1}^{N} y_i \\ \sum_{i=1}^{N} x_i y_i \\ \vdots \\ \sum_{i=1}^{N} x_i^p y_i \end{bmatrix} = \left\{ \sum_{i=1}^{N} x_i^j y_i \right\}_j \tag{27}$$

This least-squares regression method is essentially a polynomial curve-fitting technique. For example, let $p = 2$, then the quadratic curve-fitting coefficients can be found with

$$A = \begin{bmatrix} N & \sum_{i=1}^{N} x_i & \sum_{i=1}^{N} x_i^2 \\ \sum_{i=1}^{N} x_i & \sum_{i=1}^{N} x_i^2 & \sum_{i=1}^{N} x_i^3 \\ \sum_{i=1}^{N} x_i^2 & \sum_{i=1}^{N} x_i^3 & \sum_{i=1}^{N} x_i^4 \end{bmatrix}, B = \begin{bmatrix} \sum_{i=1}^{N} y_i \\ \sum_{i=1}^{N} x_i y_i \\ \sum_{i=1}^{N} x_i^2 y_i \end{bmatrix} \tag{28}$$

When all available data are used in the least-squares regression method, it is sometimes called a batch least-squares method. This is usually when there are sufficient data over a given time interval and the estimates of the unknown coefficients are not needed immediately at each time step.

A. Convex Optimization and Least-Squares Gradient Method

When the estimates of the unknown coefficients are needed at each time step, Θ can be estimated recursively using each pair of data (x_i, y_i) at each time step.

Consider the following cost function

$$J(\Theta) = \frac{1}{2}\varepsilon^\top \varepsilon \tag{29}$$

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

$$\frac{\partial J}{\partial \Theta} = \nabla J_\Theta(\Theta) = \left(\frac{\partial \varepsilon}{\partial \Theta^\top}\right)\varepsilon^\top = \Phi(x)\,\varepsilon^\top \tag{30}$$

To determine a least-squares estimation based on a given data pair at each time step, the concept of convex optimization is now introduced.

A subset $\mathscr{S}$ is said to be convex if there exist x, y in $\mathscr{S}$ and a constant $\alpha \in [0,1]$ such that $\alpha x + (1-\alpha)y$ is also in $\mathscr{S}$. A function f is said to be convex in a convex set $\mathscr{S}$ if for every x, y in $\mathscr{S}$, then

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y) \tag{31}$$

Note that $J(\Theta)$ is convex since

$$\begin{aligned}\frac{1}{2}[\alpha\varepsilon + (1-\alpha)\varepsilon_1]^\top [\alpha\varepsilon + (1-\alpha)\varepsilon_1] &= \frac{1}{2}\alpha^2\varepsilon^\top\varepsilon + \alpha(1-\alpha)\varepsilon^\top\varepsilon_1 + \frac{1}{2}(1-\alpha)^2\varepsilon_1^\top\varepsilon_1 \\ &= \frac{1}{2}\alpha^2\left(\varepsilon^\top\varepsilon - 2\varepsilon^\top\varepsilon_1\right) + \alpha\varepsilon^\top\varepsilon_1 + \frac{1}{2}(1-\alpha)^2\varepsilon_1^\top\varepsilon_1\end{aligned} \tag{32}$$

but $\alpha^2 \leq \alpha$ and $(1-\alpha)^2 \leq 1-\alpha$ for all $\alpha \in [0,1]$, so

$$\frac{1}{2}\alpha\left(\varepsilon^\top\varepsilon - 2\varepsilon^\top\varepsilon_1\right) + \alpha\varepsilon^\top\varepsilon_1 + \frac{1}{2}(1-\alpha)\varepsilon_1^\top\varepsilon_1 \leq \alpha\frac{1}{2}\varepsilon^\top\varepsilon + (1-\alpha)\frac{1}{2}\varepsilon_1^\top\varepsilon_1 \tag{33}$$

If $f \in \mathscr{C}^1$, i.e., f is differentiable at least once, then

$$f(y) \geq f(x) + (\nabla f(x))^\top (y-x) \tag{34}$$

If $f \in \mathscr{C}^2$, then f is convex if $\nabla^2 f \geq 0$ where $\nabla^2 f$ is called the Hessian of f.

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

$$J(\Theta^*) \leq J(\Theta) \tag{35}$$

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

$$\nabla J_\Theta(\Theta^*) = \nabla J_\Theta(\Theta^* + \Delta\Theta) + \nabla^2 J_\Theta(\Theta^* + \Delta\Theta)\Delta\Theta + \underbrace{\mathscr{O}\left(\Delta\Theta^\top\Delta\Theta\right)}_{\approx 0} \tag{36}$$

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

$$\Delta\Theta = -\left[\nabla^2 J_\Theta(\Theta)\right]^{-1}\nabla J_\Theta(\Theta) \tag{37}$$

Equation (37) can be written in discrete-time form as

$$\Theta_{i+1} = \Theta_i - \left[\nabla^2 J_\Theta(\Theta_i)\right]^{-1}\nabla J_\Theta(\Theta_i) \tag{38}$$

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^*. This approximation leads to the well-known steepest descent or first-order gradient method for convex optimization.

$$\Theta_{i+1} = \Theta_i - \varepsilon\nabla J_\Theta(\Theta_i) \tag{39}$$

Now dividing both sides by Δt and taking the limit as $\Delta t \to 0$ yield

$$\dot{\Theta} = -\Gamma \nabla J_{\Theta}(\Theta) \tag{40}$$

where $\Gamma = \Gamma^{\top} > 0 \in \mathbb{R}^m \times \mathbb{R}^m$ is a positive definite adaptation rate matrix that effectively replaces $\frac{\varepsilon}{\Delta t}$. This is the continuous-time version of the gradient method.

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

$$\dot{\Theta} = -\Gamma \nabla J_{\Theta}(\Theta) = -\Gamma \Phi(x) \varepsilon^{\top} \tag{41}$$

Notice the resemblance of this least-squares gradient method to a model-reference adaptive law, where the approximation error $\varepsilon(t)$ replaces the tracking error $e(t)$.

<u>Example:</u> For the example in Section 6.1, the least-squares gradient method is

$$\dot{\Theta} = \begin{bmatrix} \dot{\theta}_0 \\ \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_p \end{bmatrix} = -\Gamma \left(\begin{bmatrix} 1 & x & \cdots & x^p \\ x & x^2 & \cdots & x^{p+1} \\ \vdots & \vdots & \ddots & \vdots \\ x^p & x^{p+1} & \cdots & x^{2p} \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_p \end{bmatrix} - \begin{bmatrix} y \\ xy \\ \vdots \\ x^p y \end{bmatrix} \right) \tag{42}$$

B. Persistent Excitation and Parameter Convergence

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

$$\varepsilon = \Theta^{\top} \Phi(x) \Theta - y = \tilde{\Theta}^{\top} \Phi(x) \tag{43}$$

The least-squares gradient method can be written as

$$\dot{\tilde{\Theta}} = \dot{\Theta} = -\Gamma \Phi(x) \Phi^{\top}(x) \tilde{\Theta} \tag{44}$$

Now, choose a Lyapunov candidate function

$$V(\tilde{\Theta}) = \text{trace}\left(\tilde{\Theta}^{\top} \Gamma^{-1} \tilde{\Theta}\right) \tag{45}$$

Then

$$\dot{V}(\tilde{\Theta}) = 2\text{trace}\left(\tilde{\Theta}^{\top} \Gamma^{-1} \dot{\tilde{\Theta}}\right) = -2\text{trace}\left(\tilde{\Theta}^{\top} \Phi(x) \Phi^{\top}(x) \tilde{\Theta}\right) = -2\Phi^{\top}(x) \tilde{\Theta} \tilde{\Theta}^{\top} \Phi(x) = -2\varepsilon^{\top} \varepsilon = -2\|\varepsilon\|^2 \leq 0 \tag{46}$$

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

$$V(t \to \infty) = V(t_0) - 2\int_{t_0}^{\infty} \|\varepsilon\|^2 \, dt < \infty \tag{47}$$

which implies

$$2\int_{t_0}^{\infty} \|\varepsilon\|^2 \, dt = V(t_0) - V(t \to \infty) < \infty \tag{48}$$

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 $\dot{V}(\tilde{\Theta})$ is uniformly continuous since

$$\ddot{V}(\tilde{\Theta}) = -4\varepsilon^{\top} \dot{\varepsilon} = -4\varepsilon^{\top} \left[\dot{\Theta}^{\top} \Phi(x) + \Theta^{\top} \dot{\Phi}(x) - \dot{y}\right] \tag{49}$$

is not necessarily bounded because there is no other condition placed on $\Phi(x)$ and $y(t)$ except for $\Phi(x) \in \mathscr{L}_\infty$ and $y(t) \in \mathscr{L}_\infty$. For $\dot{V}(\tilde{\Theta})$ to be uniformly continuous, additional conditions that $\dot{\Phi}(x) \in \mathscr{L}_\infty$ and $\dot{y}(t) \in \mathscr{L}_\infty$ are required. Then, using the Barbalat's lemma, one can conclude that $\dot{V}(\tilde{\Theta}) \to 0$ or $\varepsilon(t) \to 0$ which also implies that $\dot{\Theta}(t) \to 0$ as

$t \to \infty$. Note that from Eq. (44) $\dot{\Theta}(t) \to 0$ 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 up to this point, 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 for a moment, Eq. (44) is a scalar equation whose solution is

$$\tilde{\Theta}(t) = \exp\left[-\Gamma \int_{t_0}^{t} \Phi(x)\Phi^{\top}(x)\, d\tau\right] \tilde{\Theta}(t_0) \tag{50}$$

Note that $x(t)$ is an independent variable as a function of t. Then for $\tilde{\Theta}(t)$ to be exponentially stable which implies an exponential parameter convergence, the following condition is required

$$\frac{1}{T}\int_{t}^{t+T} \Phi(x)\Phi^{\top}(x)\, d\tau \geq \alpha I \tag{51}$$

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

This condition is called a 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 easily be satisfied. This can be explained as follows: suppose a parameter identification is used for adaptation, then closed-loop stability usually implies parameter convergence to the ideal values of the unknown parameters. However, parameter convergence requires persistent excitation which depends on $x(t)$ which in turn depends on parameter convergence. This is a circular argument and, therefore, it is difficult to assert the PE condition. However, if $x(t)$ is an independent variable, then the persistent excitation condition can be assumed to be satisfied. Suppose that this is the case, then the estimation error is given by

$$\left|\tilde{\Theta}(t)\right| \leq \left|\tilde{\Theta}(t_0)\right| e^{-\gamma\alpha t}, \forall t \in [t_1, t_1+T], t_1 > t_0 \tag{52}$$

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 (but not necessarily exponentially stable because of $\Phi(x)$) with $\varepsilon(t) \to 0$ as $t \to \infty$.

C. Recursive Least-Squares

Consider the following cost function

$$J(\Theta) = \frac{1}{2}\int_{t_0}^{t} \varepsilon^{\top}\varepsilon\, d\tau \tag{53}$$

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

The necessary condition is

$$\nabla J_{\Theta}(\Theta) = \frac{\partial J^{\top}}{\partial \Theta^{\top}} = \int_{t_0}^{t} \Phi(x)\left[\Phi^{\top}(x)\Theta - y^{\top}\right] d\tau = 0 \tag{54}$$

from which Θ is obtained as

$$\Theta = \left[\int_{t_0}^{t} \Phi(x)\Phi^{\top}(x)\, d\tau\right]^{-1} \int_{t_0}^{t} \Phi(x) y^{\top} d\tau \tag{55}$$

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

$$R = \left[\int_{t_0}^{t} \Phi(x)\Phi^{\top}(x)\, d\tau\right]^{-1} \tag{56}$$

Then

$$R^{-1}\Theta = \int_{t_0}^{t} \Phi(x) y^\top d\tau \tag{57}$$

Upon differentiation, this yields

$$R^{-1}\dot{\Theta} + \frac{dR^{-1}}{dt}\Theta = \Phi(x) y^\top \tag{58}$$

From Eq. (56)

$$\frac{dR^{-1}}{dt} = \Phi(x)\Phi^\top(x) \tag{59}$$

Therefore

$$\dot{\Theta} = -R\Phi(x)\left[\Phi^\top(x)\Theta - y^\top\right] = -R\Phi(x)\varepsilon^\top \tag{60}$$

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

$$\dot{R}R^{-1} + R\frac{dR^{-1}}{dt} = 0 \tag{61}$$

Thus

$$\dot{R} = -R\Phi(x)\Phi^\top(x)R \tag{62}$$

Both Eqs. (63) and (62) constitute the well-known recursive least-squares (RLS) parameter identification method.[5] The matrix R is called the covariance matrix and the RLS formula is similar to the Kalman filter where Eq. (62) is a differential Riccati equation for a zero-order plant model. Comparing Eq. (41) with Eq. (63), R plays a role of Γ as a time-varying adaptation rate matrix and Eq. (62) 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

$$\dot{\tilde{\Theta}} = -R\Phi(x)\Phi^\top(x)\tilde{\Theta} \tag{63}$$

Choose a Lyapunov candidate function

$$V\left(\tilde{\Theta}\right) = \text{trace}\left(\tilde{\Theta}^\top R^{-1}\tilde{\Theta}\right) \tag{64}$$

Then

$$\begin{aligned}\dot{V}\left(\tilde{\Theta}\right) &= \text{trace}\left(2\tilde{\Theta}^\top R^{-1}\dot{\tilde{\Theta}} + \tilde{\Theta}^\top\frac{dR^{-1}}{dt}\tilde{\Theta}\right) = \text{trace}\left(-2\tilde{\Theta}^\top\Phi(x)\Phi^\top(x)\tilde{\Theta} + \tilde{\Theta}^\top\Phi(x)\Phi^\top(x)\tilde{\Theta}\right)\\ &= -\text{trace}\left(\tilde{\Theta}^\top\Phi(x)\Phi^\top(x)\tilde{\Theta}\right) = -\varepsilon^\top\varepsilon = -\|\varepsilon\|^2 \le 0\end{aligned} \tag{65}$$

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

$$V(t \to \infty) = V(t_0) - \int_{t_0}^{\infty} \|\varepsilon\|^2 dt < \infty \tag{66}$$

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

$$\begin{aligned}\ddot{V}\left(\tilde{\Theta}\right) &= -2\varepsilon^\top\dot{\varepsilon} = -2\varepsilon^\top\left[\dot{\tilde{\Theta}}^\top\Phi(x) + \tilde{\Theta}^\top\dot{\Phi}(x)\right] = -2\varepsilon^\top\left[-\tilde{\Theta}^\top\Phi(x)\Phi^\top(x)R\Phi(x) + \tilde{\Theta}^\top\dot{\Phi}(x)\right]\\ &= -2\varepsilon^\top\left[-\tilde{\Theta}^\top\Phi(x)\Phi^\top(x)\left[\int_{t_0}^{t}\Phi(x)\Phi^\top(x)d\tau\right]^{-1}\Phi(x) + \tilde{\Theta}^\top\dot{\Phi}(x)\right]\end{aligned} \tag{67}$$

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.

Note that there are various versions of the RLS method. One popular version is the RLS method with normalization where the adaptive law for R is modified as follows:

$$\dot{R} = -\frac{R\Phi(x)\Phi^{\top}(x)R}{1+n^2} \tag{68}$$

where $1+n^2 = 1+\Phi^{\top}(x)R\Phi(x)$ is called a normalization factor.

The time derivative of the Lyapunov candidate function for the RLS method with normalization is

$$\begin{aligned}\dot{V}\left(\tilde{\Theta}\right) &= \text{trace}\left(-2\tilde{\Theta}^{\top}\Phi(x)\Phi^{\top}(x)\tilde{\Theta} + \frac{\tilde{\Theta}^{\top}\Phi(x)\Phi^{\top}(x)\tilde{\Theta}}{1+n^2}\right) = -\text{trace}\left(\tilde{\Theta}^{\top}\Phi(x)\Phi^{\top}(x)\tilde{\Theta}\left(1+2n^2\right)\right) \\ &= -\varepsilon^{\top}\varepsilon\left(1+2n^2\right) = -\|\varepsilon\|^2\left(1+2n^2\right) \le 0\end{aligned} \tag{69}$$

Note that $\dot{V}\left(\tilde{\Theta}\right)$ is more negative with than without normalization. Therefore, the effect of normalization is to make the adaptive law for R more stable, but the parameter convergence is slower.

Another popular version is the RLS method with forgetting factor and normalization which is given by without derivation

$$\dot{R} = \beta R - \frac{R\Phi(x)\Phi^{\top}(x)R}{1+n^2} \tag{70}$$

where $0 \le \beta \le 1$ is called a forgetting factor.

IV. Adaptive Control with Unstructured Uncertainty

The RLS method has been used in adaptive control applications and demonstrated to be highly effective in estimating parametric uncertainty in adaptive control. In a previous study, a hybrid direct-indirect adaptive control with the RLS was developed. Other techniques based on the RLS have been recently developed such as the RLS-based modification in adaptive control.[7] In this study, the approach is considered to be a direct adaptive control method.

Consider a second-order SISO system

$$\ddot{y} + a\dot{y} + cy = b\left[u + f(y)\right] \tag{71}$$

where a, b, and c are known and $f(y)$ is an unstructured uncertainty

The state-space form of the system is

$$\dot{x} = Ax + B\left[u + f(y)\right] \tag{72}$$

where $x = \begin{bmatrix} y & \dot{y} \end{bmatrix}^{\top} \in \mathbb{R}^2$ and

$$A = \begin{bmatrix} 0 & 1 \\ c & a \end{bmatrix},\ B = \begin{bmatrix} 0 \\ b \end{bmatrix} \tag{73}$$

A reference model is given by

$$\dot{x}_m = A_m x + B_m r \tag{74}$$

where $x_m \in \mathbb{R}^2$, $r \in \mathbb{R}$, and

$$A_m = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix},\ B_m = \begin{bmatrix} 0 \\ b_m \end{bmatrix} \tag{75}$$

with $\zeta > 0$ and $\omega_n > 0$.

The nominal system is designed to track the reference model with a nominal controller

$$\bar{u} = -K_x x + k_r r \tag{76}$$

where $A - BK_x = A_m$ and $Bk_r = B_m$.

$$K_x = \left(B^{\top}B\right)^{-1}B^{\top}\left(A - A_m\right) \tag{77}$$

$$k_r = \left(B^\top B\right)^{-1} B^\top B_m \tag{78}$$

Note that $\left(B^\top B\right)^{-1} B^\top$ is the left pseudo-inverse of B and is equal to

$$\left(B^\top B\right)^{-1} B^\top = \frac{1}{b^2} B^\top \tag{79}$$

To accommodate the uncertainty, an adaptive controller is included in the controller

$$u = \bar{u} + u_{ad} \tag{80}$$

where the adaptive controller

$$u_{ad} = -\Theta^\top \Phi(y) \tag{81}$$

is designed to estimate the unstructured uncertainty which can be expressed as

$$f(y) = \Theta^{*\top} \Phi(y) - \varepsilon(y) \tag{82}$$

where $\Theta^* \in \mathbb{R}^p$ is an unknown constant ideal weight matrix, $\Phi(y) \in \mathbb{R}^p$ is a vector of known bounded regressors or basis functions, and $\varepsilon(y) \in \mathbb{R}$ is an approximation error.

In general, $\Phi(y)$ can be any bounded regressor function. However, with the choice of Chebyshev 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

$$f(y) = \Theta^{*\top} \Phi\left(W^{*\top} \bar{y}\right) \tag{83}$$

where $\Theta^* \in \mathbb{R}^{p+1}$ and $W^* \in \mathbb{R}^2$ are unknown constant ideal weight matrices, $\bar{y} = \begin{bmatrix} 1 & y \end{bmatrix}^\top \in \mathbb{R}^{n+1}$, $\Phi\left(W^{*\top}\bar{y}\right) \in \mathbb{R}^{p+1}$.

Invoking the Weierstrass's theorem, $\varepsilon(y)$ can be made sufficiently small in a compact domain of $y(t)$ such that $\sup_{y \in \mathscr{D}} \|\varepsilon(y)\| \leq \varepsilon_0 \forall y \in \mathscr{D} \subset \mathbb{R}$ by a suitable selection of $\Phi(y)$.

Define a desired plant model as

$$\dot{x}_d = A_m x + B_m r \tag{84}$$

Then formulating a plant modeling error as

$$\bar{\varepsilon} = \dot{x}_d - \dot{x} = Ax + B\bar{u} - \dot{x} = B\tilde{\Theta}^\top \Phi(y) + B\varepsilon \tag{85}$$

The RLS adaptive law for estimating Θ is given by

$$\dot{\Theta} = -R\Phi(y)\bar{\varepsilon}^\top B\left(B^\top B\right)^{-1} \tag{86}$$

$$\dot{R} = -\eta R\Phi(y)\Phi^\top(y) R \tag{87}$$

where $0 \leq \eta \leq 1$.

The estimation error equation for the RLS adaptive law is then obtained as

$$\dot{\tilde{\Theta}} = -R\Phi(y)\left[\Phi^\top(y)\tilde{\Theta}B^\top + \varepsilon^\top B^\top\right] B\left(B^\top B\right)^{-1} = -R\Phi(y)\left[\Phi^\top(y)\tilde{\Theta} + \varepsilon^\top\right] \tag{88}$$

The tracking error equation can be expressed in terms of the approximation error as

$$\dot{e} = \dot{x}_m - \dot{x} = \dot{x}_m - \dot{x}_d + \dot{x}_d - \dot{x} = A_m e + \bar{\varepsilon} = A_m e + B\tilde{\Theta}^\top \Phi(y) + B\varepsilon \tag{89}$$

Now, choose a Lyapunov candidate function

$$V\left(e, \tilde{\Theta}\right) = e^\top P e + \operatorname{trace}\left(\tilde{\Theta}^\top R^{-1} \tilde{\Theta}\right) \tag{90}$$

Then $\dot{V}$ is

$$\begin{aligned}\dot{V}\left(e,\tilde{\Theta}\right) &= -e^{\top}Qe+2e^{\top}P\bar{\varepsilon}+\text{trace}\left(2\tilde{\Theta}^{\top}R^{-1}\dot{\tilde{\Theta}}+\tilde{\Theta}^{\top}\frac{dR^{-1}}{dt}\tilde{\Theta}\right)\\ &= -e^{\top}Qe+2e^{\top}P\bar{\varepsilon}+\text{trace}\left(-2\tilde{\Theta}^{\top}\Phi(y)\left[\Phi^{\top}(y)\tilde{\Theta}+\varepsilon^{\top}\right]+\eta\tilde{\Theta}^{\top}\Phi(y)\Phi^{\top}(y)\tilde{\Theta}\right)\\ &= -e^{\top}Qe+2e^{\top}P\bar{\varepsilon}-(2-\eta)\Phi^{\top}(y)\tilde{\Theta}\tilde{\Theta}^{\top}\Phi(y)-2\varepsilon^{\top}\tilde{\Theta}^{\top}\Phi(y)\\ &\leq -e^{\top}Qe+2e^{\top}P\bar{\varepsilon}-\Phi^{\top}(y)\tilde{\Theta}\tilde{\Theta}^{\top}\Phi(y)-2\varepsilon^{\top}\tilde{\Theta}^{\top}\Phi(y)\end{aligned} \tag{91}$$

Note that

$$\frac{\|\bar{\varepsilon}\|^2}{\|B\|^2}=\Phi^{\top}(y)\tilde{\Theta}\tilde{\Theta}^{\top}\Phi(y)+2\varepsilon^{\top}\tilde{\Theta}^{\top}\Phi(y)+\varepsilon^{\top}\varepsilon \tag{92}$$

Therefore

$$\dot{V}\left(e,\tilde{\Theta}\right)\leq -\lambda_{min}(Q)\|e\|^2+2\|P\|\|e\|\|\bar{\varepsilon}\|-\frac{\|\bar{\varepsilon}\|^2}{\|B\|^2}+\varepsilon_0^2 \tag{93}$$

where $\varepsilon_0=\sup_{y\in\mathscr{D}}\|\varepsilon(y)\|$.

Define a compact set $\mathscr{S}$

$$\mathscr{S}=\left\{(e,\bar{\varepsilon})\,|\lambda_{min}(Q)\|e\|^2-2\lambda_{max}(P)\|e\|\|\bar{\varepsilon}\|+\frac{\|\bar{\varepsilon}\|^2}{\|B\|^2}\leq\varepsilon_0^2\right\} \tag{94}$$

$\dot{V}\left(e,\tilde{\Theta}\right)>0$ inside of $\mathscr{S}$, but $\dot{V}\left(e,\tilde{\Theta}\right)\leq 0$ outside $\mathscr{S}$. Therefore, $e(t)\in\mathscr{L}_\infty$ and $\bar{\varepsilon}(t)\in\mathscr{L}_\infty$. Define $\bar{\varepsilon}_0=\sup_{x\in\mathscr{D}}\|\bar{\varepsilon}(x)\|$, then

$$\|e\|\geq r=\frac{\lambda_{max}(P)\bar{\varepsilon}_0+\sqrt{\lambda_{max}^2(P)\bar{\varepsilon}_0^2+\lambda_{min}(Q)\left(\varepsilon_0^2-\bar{\varepsilon}_0^2/\|B\|^2\right)}}{\lambda_{min}(Q)} \tag{95}$$

It can be shown that $e(t)$ is uniformly ultimately bounded with an ultimate bound

$$\rho=\sqrt{\frac{\lambda_{max}(P)}{\lambda_{min}(P)}}r \tag{96}$$

<u>Example:</u> Consider a first-order scalar system with unstructured uncertainty

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

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

$$\dot{x}_m=a_mx_m+b_mr$$

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

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

$$f(x)=a_0+a_1x+\cdots+a_qx^q-\varepsilon(x)=\Theta^{*\top}\Phi(x)-\varepsilon(x)$$

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

The controller is designed as

$$u=k_x(t)x+k_rr-\Theta^{\top}(t)\Phi(x)$$

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

$$\dot{k}_x=\frac{\gamma x\bar{\varepsilon}}{b}$$

$$\dot{\Theta}=-\frac{\Gamma\Phi(x)\bar{\varepsilon}}{b}$$

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

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

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

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

The tracking error for $q = 1,2,3,4$ are shown in the following plots. 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.

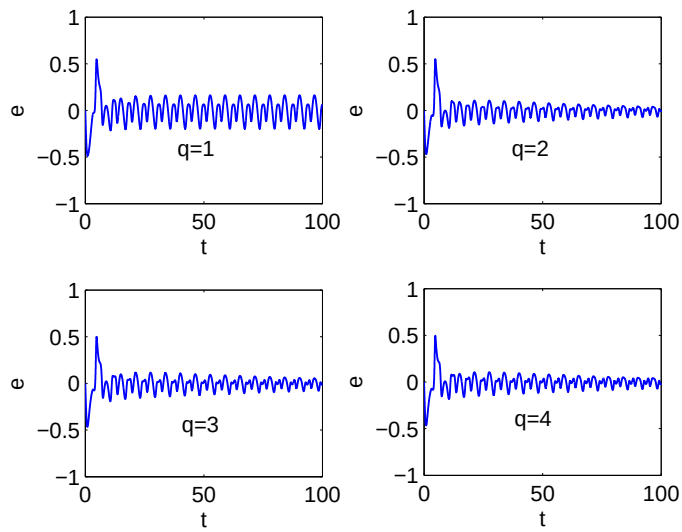

Fig. 1 - Tracking Error due to Least-Squares Gradient Method with regular Polynomial Approximation

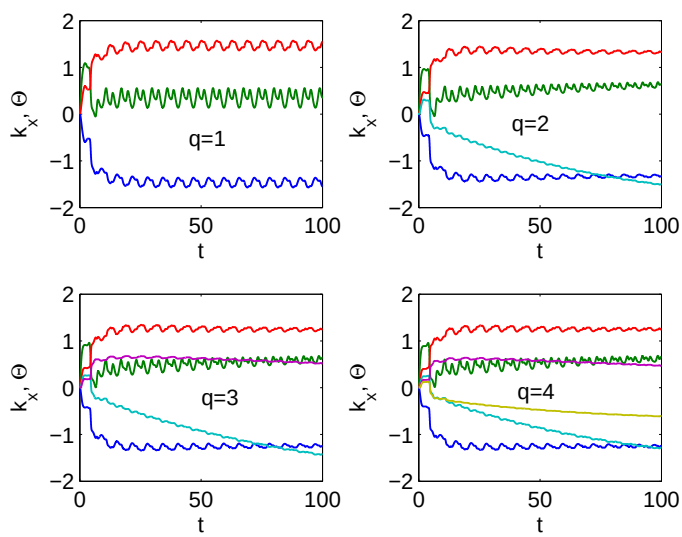

Fig. 2 - k_x and Θ due to Least-Squares Gradient Method with regular Polynomial Approximation

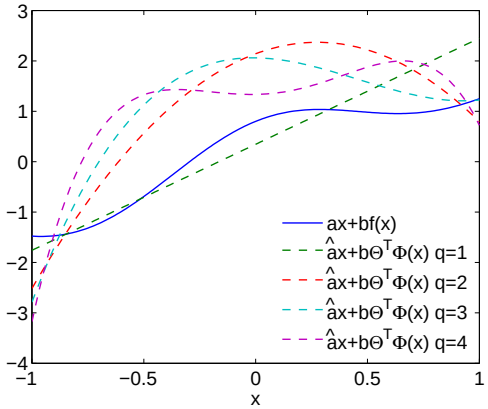

Fig. 3 - Function Approximation at $t = 100$ due to Least-Squares Gradient Method with regular Polynomial

Now suppose the Chebyshev orthogonal polynomials are used instead. Then

$$f(x) = a_0 + a_1 T_1(x) + \cdots + a_q T_q(x) - \varepsilon(x) = \Theta^{*\top}\Phi(x) - \varepsilon(x)$$

The simulation results are as shown. 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 polynomial and is even smaller than that for $q = 4$ with the regular polynomial. For $q = 4$, the Chebyshev polynomial approximation results in a very small tracking error. The unknown function is very well approximated by a 4th-degree Chebyshev polynomial as shown in Fig. 6.

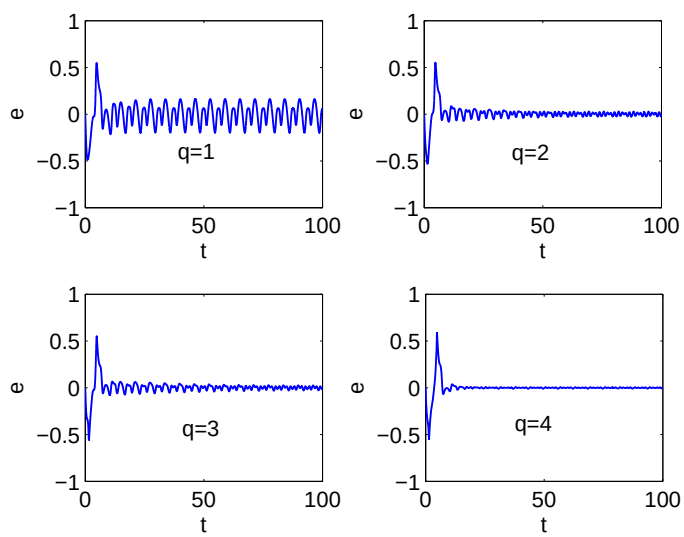

Fig. 4 - Tracking Error due to Least-Squares Gradient Method with Chebyshev Polynomial Approximation

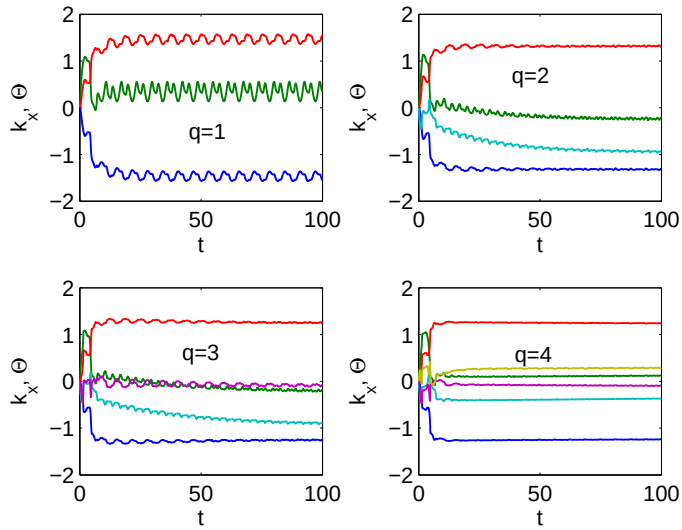

Fig. 5 - k_x and Θ due to Least-Squares Gradient Method with Chebyshev Polynomial Approximation

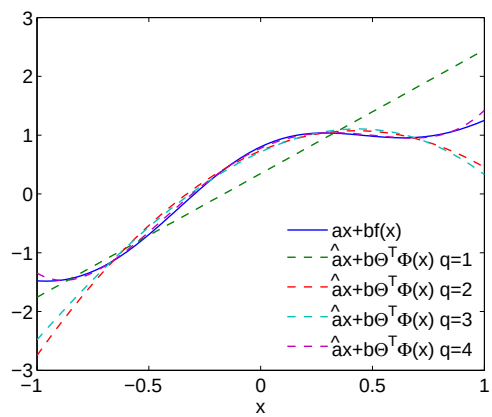

Fig. 6 - Function Approximation at $t = 100$ Least-Squares Gradient Method with Chebyshev Polynomial

In contrast, let us compare the least-squares adaptive control with the standard model-reference adaptive control (MRAC). The MRAC update laws are given by

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

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

where $e = x_m - x$.

Figure 8 illustrates parameter convergence of MRAC. As noted, the tracking error is not as good with the MRAC as with the least-squares gradient method. The parameters $k(t)$ and $\Theta(t)$ are more oscillatory. The function approximation by the MRAC adaptive laws is poorer than that by the least-squares gradient method. Furthermore, for systems with unstructured uncertainty, MRAC is known to be non-robust since the parameter estimation error is not necessarily bounded. Therefore, robust modification or a projection method must be used to ensure that the parameter estimation error is bounded.

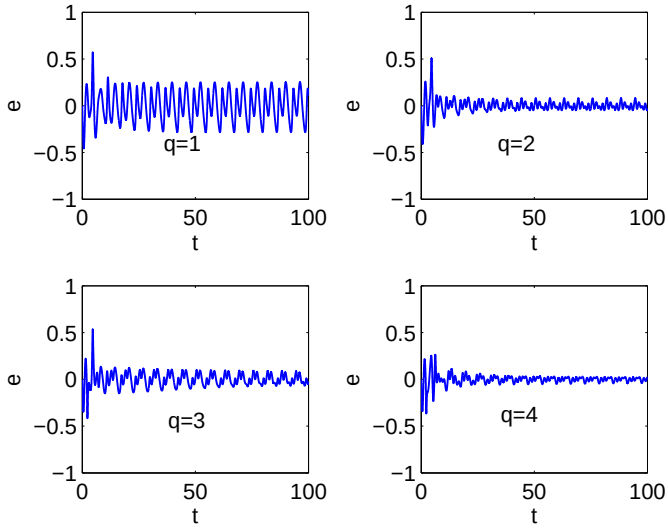

Fig. 7 - Tracking Error due to MRAC with Chebyshev Polynomial Approximation

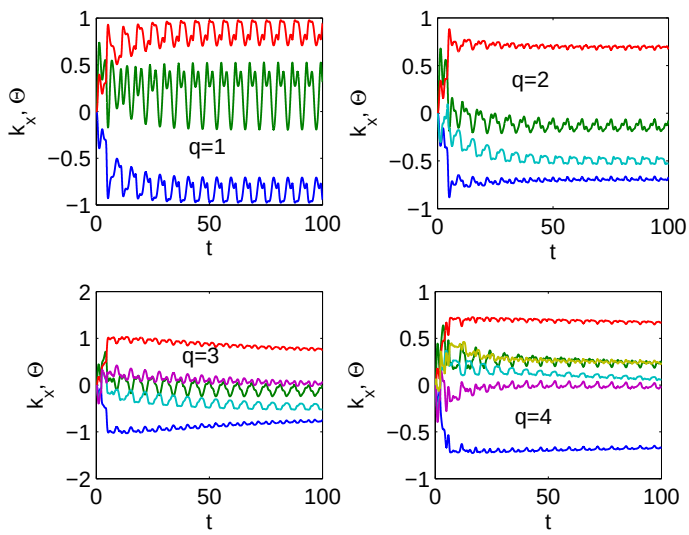

Fig. 8 - k_x and Θ due to MRAC with Chebyshev Polynomial Approximation

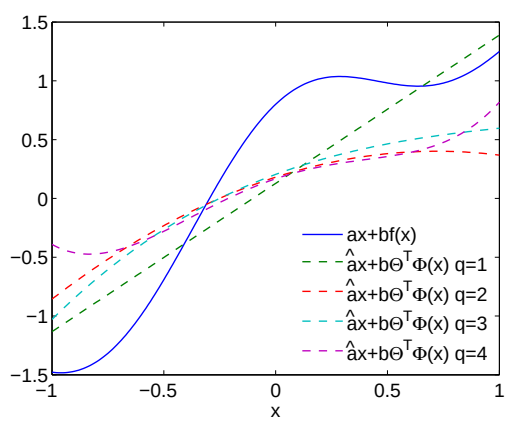

Fig. 9 - Function Approximation at $t = 100$ due to MRAC with Chebyshev Polynomial

V. Flight Control Application

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

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\theta} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \frac{Z_\alpha}{\bar{u}} & 0 & 1 \\ 0 & 0 & 1 \\ M_\alpha + \frac{M_{\dot{\alpha}} Z_\alpha}{\bar{u}} & 0 & M_q + M_{\dot{\alpha}} \end{bmatrix} \begin{bmatrix} \alpha \\ \theta \\ q \end{bmatrix} + \begin{bmatrix} \frac{Z_{\delta_e}}{\bar{u}} \\ 0 \\ M_{\delta_e} + \frac{M_{\dot{\alpha}} Z_{\delta_e}}{\bar{u}} \end{bmatrix} [\delta_e + f(\alpha)] \tag{97}$$

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

$$\ddot{\theta}_m + 2\zeta\omega_n\dot{\theta}_m + \omega_n^2\theta_m = \omega_n^2\theta_c \tag{98}$$

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

$$\ddot{\theta} - \left(M_\alpha + \frac{M_{\dot{\alpha}} Z_\alpha}{\bar{u}}\right)\alpha - (M_q + M_{\dot{\alpha}})\dot{\theta} = \left(M_{\delta_e} + \frac{M_{\dot{\alpha}} Z_{\delta_e}}{\bar{u}}\right)[\delta_e + f(\alpha)] \tag{99}$$

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

$$\delta_e = -k_\alpha\alpha - k_\theta(\theta - \theta_c) - k_q q - \Theta^\top\Phi(\alpha) = -K_x x + k_\theta\theta_c - \Theta^\top\Phi(\alpha) \tag{100}$$

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

$$k_\alpha = \frac{M_\alpha + \frac{M_{\dot{\alpha}} Z_\alpha}{\bar{u}}}{M_{\delta_e} + \frac{M_{\dot{\alpha}} Z_{\delta_e}}{\bar{u}}} \tag{101}$$

$$k_\theta = \frac{\omega_n^2}{M_{\delta_e} + \frac{M_{\dot{\alpha}} Z_{\delta_e}}{\bar{u}}} \tag{102}$$

$$k_q = \frac{2\zeta\omega_n + M_q + M_{\dot{\alpha}}}{M_{\delta_e} + \frac{M_{\dot{\alpha}} Z_{\delta_e}}{\bar{u}}} \tag{103}$$

The numerical model of the short-period dynamics 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

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

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 $\bar{\varepsilon} = \dot{x}_d - \dot{x} = A_m x + B_m r - \dot{x}$, assuming that the angle of attack rate and pitch acceleration are measurable from the vertical acceleration sensors. The uncertainty is modeled with the first four terms of the Chebyshev basis polynomials

$$\Theta^\top \Phi(\alpha) = \theta_1 + \theta_2 \alpha + \theta_3 \left(2\alpha^2 - 1\right) + \theta_4 \left(4\alpha^3 - 3\alpha\right)$$

The RLS parameter estimation is computed by with $\eta = 0$, which effectively is a least-squares gradient method, and $\eta = 0.2$. The covariance matrix is chosen to be $R = 20I$. The aircraft longitudinal responses for $\eta = 0$ and $\eta = 0.2$ are as shown in Figs. 10 and 11.

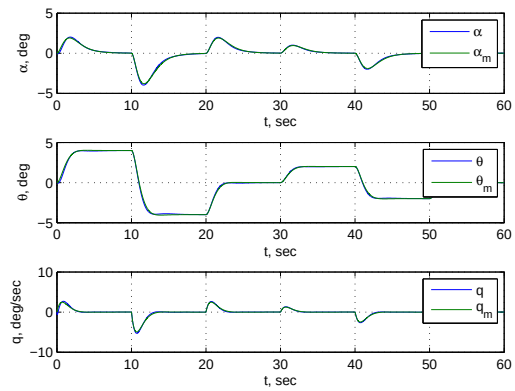

Fig. 10 - Aircraft Response with Least-Squares Gradient Adaptive Control ($\eta = 0$)

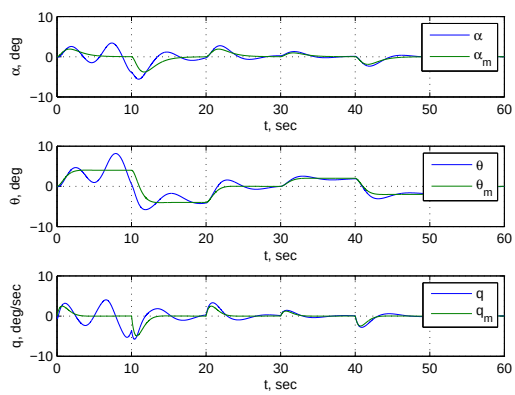

Fig. 11 - Aircraft Response with RLS Adaptive Control ($\eta = 0.2$)

It can be seen that the least-squares gradient method (RLS with $\eta = 0$) results in a very good tracking performance, but the RLS method with $\eta = 0.2$ exhibits poor performance. This is expected as the rate of parameter convergence for the RLS is proportional to $-(2-\eta)\|\bar{\varepsilon}\|^2$ according to the Lyapunov analysis. However, the slow parameter convergence of the RLS can improve stability robustness of adaptive control in the presence of time delay or unmodeled dynamics, as will be shown later.

For comparison, the parameter estimation is computed by the standard MRAC method using the same Chebyshev basis polynomials according to

$$\dot{\Theta} = -\Gamma \Phi(\alpha) e^\top P B \tag{104}$$

where $e = x_m - x$.

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

$$f(\alpha) = \Theta^\top \Phi\left(W^\top \bar{\alpha}\right) \tag{105}$$

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

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

$$\dot{\Theta} = -\Gamma_\Theta \Phi \left(W^\top \bar{\alpha} \right) e^\top PB \tag{106}$$

$$\dot{W} = -\Gamma_W \bar{\alpha} e^\top PBV^\top \sigma' \left(W^\top \bar{\alpha} \right) \tag{107}$$

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 polynomial and the neural network are as shown in Figs. 12 and 13.

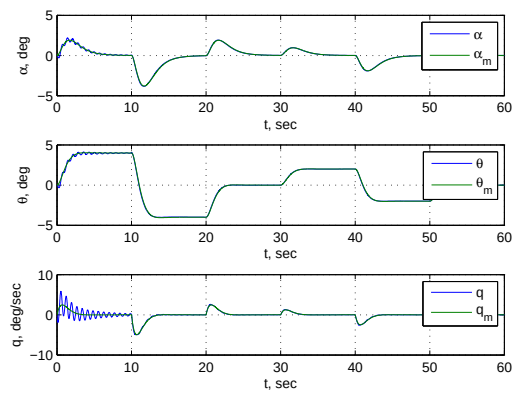

Fig. 12 - Aircraft Response with MRAC

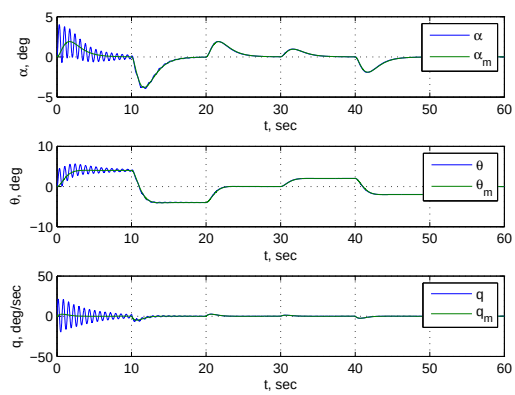

Fig. 13 - Aircraft Response with Neural Network MRAC

The responses both exhibit initial high frequency oscillations which are indicative of incipient instability, even though the subsequent tracking performance is very good. The neural network adaptive control has much more pronounced high frequency oscillations which are due to the weights initialization with random numbers.

To illustrate the issue of robustness and show that the RLS is actually better able to handle a time delay or unmodeled dynamics than the least-squares gradient method or MRAC, a numerical evidence of the time delay margin is computed for each of the four adaptive laws. 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	60 ms

Table 1 - Estimated Time Delay Margins

The RLS method has the best time delay margin than the other methods. The standard MRAC has very poor robustness which is a well-known fact.[8] Generally, the standard MRAC has to be modified to improve robustness using the well-known σ-modification[5] and e-modification[9] or the recently developed optimal control modification[10] and adaptive loop recovery.[11]

The aircraft responses due to a 60-ms time delay for the least-squares gradient method, RLS with $\eta = 0.2$, and neural network MRAC are illustrated in Figs. 14, 15, and 16. The aircraft response due to a 10-ms time delay for the MRAC is plotted in Fig. 17. As can be seen, the least-squares gradient method maintains a very good tracking performance even with a 60-ms time delay. Both the MRAC and neural network MRAC exhibit high frequency oscillations. The RLS method with $\eta = 0.2$ exhibits low frequency transients even though it is much more robust than the other three adaptive laws. Thus, if the time delay is not too large, the least-squares gradient method seems to perform the best among the adaptive laws.

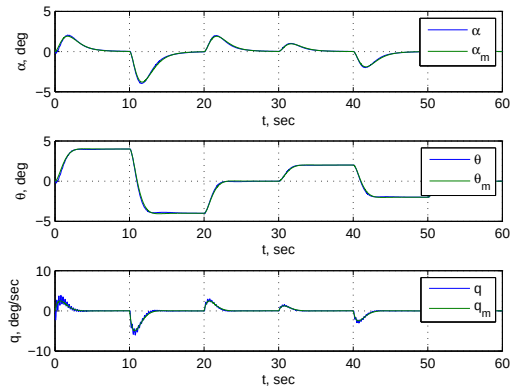

Fig. 14 - Aircraft Response with Least-Squares Gradient Adaptive Control ($\eta = 0$) with 60-ms Time Delay

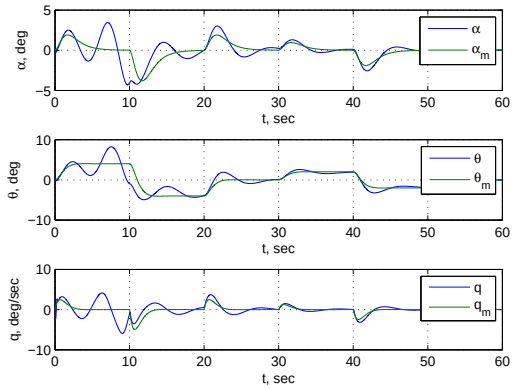

Fig. 15 - Aircraft Response with RLS Adaptive Control ($\eta = 0.2$) with 60-ms Time Delay

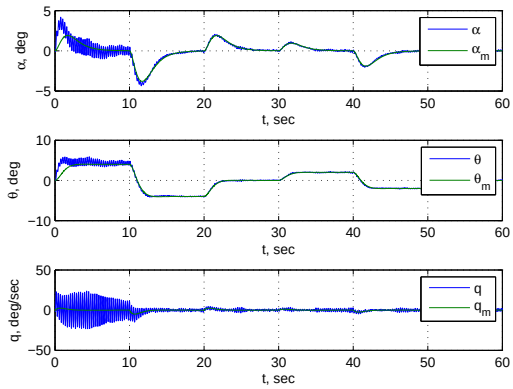

Fig. 16 - Aircraft Response with Neural Network MRAC ($\eta = 0.2$) with 60-ms Time Delay

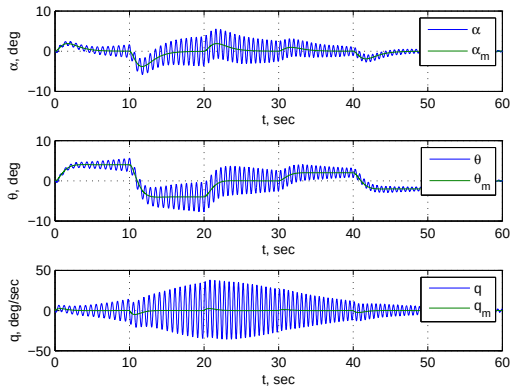

Fig. 17 - Aircraft Response with MRAC ($\eta = 0.2$) with 10-ms Time Delay

VI. Conclusions

This paper presents an adaptive control method for systems with unstructured uncertainty. The adaptive control method uses Chebyshev orthogonal polynomials as basis functions to approximate the unstructured uncertainty. The Chebyshev polynomials have many desirable features in function approximation and can be shown to be the "best" polynomial function approximation. A recursive least-squares adaptive control method is developed for second-order systems using Chebyshev polynomials as basis functions for parameter estimation. The adaptation is driven by a plant modeling error as opposed to the usual tracking error in model-reference adaptive control. Simulations demonstrate the superior performance of Chebyshev polynomials in an adaptive control setting over regular polynomials and even neural networks. The least-squares gradient method demonstrates to outperform both the recursive least-squares adaptive control and the standard, unmodified 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. A., "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. C. and Handscomb, D., *Chebyshev Polynomials*, Chapman and Hall/CRC, 2002.

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

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

[7]Chowdhary, G. and Johnson, E. N., "Recursively Updated Least Squares Based Modification Term for Adaptive Control", American Control Conference, Baltimore, June 2010.

[8]Rohrs, C.E., 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.

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

[10]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.

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