Lyapunov-based Integration of a Data Recording Algorithm in Adaptive Control

Girish Chowdhary, Eric N. Johnson
Georgia Institute of Technology, Atlanta, GA, 30318

Recently we have developed concurrent learning model reference adaptive controllers which use recorded data concurrently with current data and can guarantee exponential stability of the closed loop without requiring persistency of excitation. The rate of convergence of these controllers is dependent on the quality of the recorded data, particularly the minimum singular value of the matrix containing the recorded states. In this paper, we use a Lyapunov framework to integrate a singular value maximizing data recording algorithm with concurrent learning adaptive controllers.

I. Introduction

Modern aerospace vehicles are expected to operate reliably in presence of uncertainties. This has necessitated that modern autopilots guarantee stability of aircraft in presence of significant modeling uncertainties. Adaptive flight control is a widely studied methodology for this purpose. For example, Calise, Johnson, Kannan and others have developed model reference adaptive controllers for both fixed wing and rotary wing Unmanned Aerial Systems (UAS). Cao, Yang, Hovakimen, and others have developed the $L_1$ adaptive control method. Lavertsky, Nguyen and others have extended direct adaptive control methods to fault tolerant control and developed techniques in composite/hybrid adaptation.

We recently developed a concurrent learning adaptive controller, which uses recorded data concurrently with current data for adaptation. We have shown that concurrent learning adaptive controllers can guarantee exponential closed loop stability without requiring persistency of excitation if the plant uncertainty can be parameterized using a set of known basis functions. Furthermore, we have shown that when the structure of the uncertainty is unknown, concurrent learning neuro-adaptive controllers can guarantee uniform ultimate boundedness of the tracking error in a neighborhood of zero and the uniform ultimate boundedness of the neural network weights in a compact neighborhood of the ideal weights. Concurrent learning formalizes the intuitive concept that if recorded data from when the system states were exciting is used concurrently for adaptation, then the system states do not need to be persistently exciting for guaranteeing exponential stability.

In concurrent learning, an algorithm is often used to specifically select and record data to be used for concurrent adaptation in a history-stack. In our previous work we have shown that in the case of a static history-stack, the convergence was directly proportional to the minimum singular value of the recorded data. We have also shown that the data recording algorithms greatly affect the rate of convergence, and have compared three different data recording algorithms. In this paper, we extend the convergence results to the case when the history-stack is time varying as a result of online recording and removal of data points. Such removal and addition of data points results in switching in the closed loop dynamics. We establish the global exponential stability of the integrated closed loop through using Lyapunov like arguments.

We begin with a brief introduction of approximate model inversion based Model Reference Adaptive Control (MRAC) in Section II. Section III discusses the concurrent learning adaptive law and a singular value maximizing data recording algorithm. The stability properties of the integrated closed loop for the case of structured uncertainty are discussed in section IV, and for the case of unstructured uncertainty are discussed in section V. Numerical simulations are used to validate the claims in Section VI. The paper is concluded in section VII.

---

* Research Engineer II, School of Aerospace Engineering, AIAA member
† Lockheed Martin Professor of Avionics Integration, School of Aerospace Engineering, AIAA member

Copyright © 2011 by Girish Chowdhary and Eric Johnson. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission.

American Institute of Aeronautics and Astronautics
II. Model Reference Adaptive Control

This section discusses the formulation of Model Reference Adaptive Control using approximate model inversion. Let $D_x \in \mathbb{R}^n$ be compact, and let $x(t) \in D_x$ be the known state vector, let $\delta \in \mathbb{R}^k$ denote the control input, and consider the following system:

$$\dot{x} = f(x(t), \delta(t)), \quad (1)$$

where the function $f$ is assumed to be continuously differentiable in $x \in D_x$, and control input $\delta$ is assumed to be bounded and piecewise continuous. The conditions for the existence and the uniqueness of the solution to 1 are assumed to be met.

Since the exact model 1 is usually not available or not invertible, we introduce an approximate inversion model $\hat{f}(x, \delta)$ which can be inverted with respect to $\delta$ to determine the control input:

$$\delta = \hat{f}^{-1}(x, \nu). \quad (2)$$

Where $\nu$ is the pseudo control input, which represents the desired model output $\dot{x}$ and is expected to be approximately achieved by $\delta$. Hence, the pseudo control input is the output of the approximate inversion model:

$$\nu = \hat{f}(x, \delta). \quad (3)$$

This approximation results in a model error of the form:

$$\dot{x} = \nu(x, \delta) + \Delta(x, \delta) \quad (4)$$

where the model error $\Delta : \mathbb{R}^{n+k} \rightarrow \mathbb{R}^n$ is given by:

$$\Delta(x, \delta) = f(x, \delta) - \hat{f}(x, \delta). \quad (5)$$

A reference model can be designed that characterizes the desired response of the system:

$$\dot{x}_{rm} = f_{rm}(x_{rm}, r(t)), \quad (6)$$

where $f_{rm}(x_{rm}(t), r(t))$ denote the reference model dynamics which are assumed to be continuously differentiable in $x$ for all $x \in D_x \subset \mathbb{R}^n$. The command $r(t)$ is assumed to be bounded and piecewise continuous, furthermore, it is assumed that all requirements for guaranteeing the existence of a unique solution to 6 are satisfied. It is also assumed that the reference model states remain bounded for a bounded $r(t)$.

A tracking control law consisting of a linear feedback part $u_{pd} = Kx$, a linear feedforward part $u_{crm} = \dot{x}_{rm}$, and an adaptive part $u_{ad}(x)$ is proposed to have the following form:

$$u = u_{crm} + u_{pd} - u_{ad}. \quad (7)$$

Define the tracking error $e$ as $e(t) = x_{rm}(t) - x(t)$, then, letting $A = -K$ the tracking error dynamics are found to be:4,11

$$\dot{e} = Ae + [u_{ad}(x, \delta) - \Delta(x, \delta)]. \quad (8)$$

The baseline full state feedback controller $u_{pd} = Kx$ is assumed to be designed such that $A$ is a Hurwitz matrix. Hence for any positive definite matrix $Q \in \mathbb{R}^{n \times n}$, a positive definite solution $P \in \mathbb{R}^{n \times n}$ exists to the Lyapunov equation:

$$A^TP + PA + Q = 0. \quad (9)$$

Letting $z = [x, \delta] \in \mathbb{R}^{n+k}$, the following two cases for characterizing the uncertainty $\Delta(x)$ are considered:

**Case I: Structured Uncertainty:** Let $z = (x, \delta)$ and consider the case where it is known that the uncertainty is linearly parameterized with a known nonlinear basis $\Phi(z)$ is known. This case is captured through the following assumption:

**Assumption 1.** The uncertainty $\Delta(z)$ can be linearly parameterized, that is, there exist a matrix of constants $W^* \in \mathbb{R}^{n \times m}$ and a vector of continuously differentiable functions $\Phi(z) = [\phi_1(z), \phi_2(z), \ldots, \phi_m(z)]^T$ such that

$$\Delta(z) = W^*\Phi(z). \quad (10)$$
In this case letting $W$ denote the estimate $W^*$ the adaptive law can be written as

$$u_{ad}(z) = W^T \Phi(z).$$

(11)

A large class of aircraft plants can be modeled in this manner.

**Case II: Unstructured Uncertainty:** Consider the case when it is only known that the uncertainty $\Delta(z)$ is continuous and defined over a compact domain $D \subseteq \mathbb{R}^{n+k}$. In this case, a Radial Basis Function (RBF) Neural Network (NN) can be used as the adaptive element. In this case the adaptive element takes the following form

$$u_{ad}(z) = W^T \sigma(z),$$

(12)

where $W \in \mathbb{R}^{n \times l}$ and $\sigma(z) = [\sigma_1(z), \sigma_2(z), \sigma_3(z), ..., \sigma_l(z)]^T$ is a vector of known radial basis functions. For $i = 2, 3, ..., l$ let $c_i$ denote the RBF centroid and $\mu_i$ denote the RBF width then for each $i$ the radial basis functions are given as

$$\sigma_i(x) = e^{-\|z-c_i\|^2/\mu_i}.$$

(13)

Appealing to the universal approximation property of RBF NN (see reference 17 or 18) we have that given a fixed number of radial basis functions $l$ there exists ideal weights $W^* \in \mathbb{R}^{n \times l}$ and a real number $\bar{\varepsilon}(z)$ such that the following approximation holds for all $x \in D$ where $D$ is compact:

$$\Delta(x) = W^{*T} \sigma(z) + \bar{\varepsilon}(z),$$

(14)

and $\bar{\varepsilon} = \sup_{z \in D} \|\bar{\varepsilon}(z)\|$ can be made arbitrarily small when sufficient number of radial basis functions are used.

Figure 1 depicts the control architecture for MRAC control discussed in this section.

![Figure 1. Neural Network Adaptive Control using Approximate Model Inversion](image)

**A. Baseline Adaptive Law**

For the case of structured uncertainty it is well known that in the presence of persistently exciting input the following adaptive law

$$\dot{W} = -\Gamma_W \Phi(z) e^T P$$

(15)

where $\Gamma_W$ is a positive definite matrix of appropriate dimensions results in ultimate boundedness of the weights and $e(t) \to 0$. Equation 15 will be referred to as the baseline adaptive law. Furthermore, replacing $\Phi(z)$ with $\sigma(z)$ in equation 15 results in the baseline gradient based adaptive law for the case
of unstructured uncertainty (case 2). For this case, the baseline adaptive law guarantees uniform ultimate boundedness of tracking error $e$.\cite{15,22,23}

where $\Gamma_w$ is a positive definite matrix of appropriate dimensions results in ultimate boundedness of the weights and $e(t) \to 0$.\cite{18,21,24} Equation 15 will be referred to as the baseline adaptive law. It can be shown that the baseline adaptive law can be arrived at by minimizing $e^T(t)e(t)$ using a gradient descent methodology,\cite{21,24} hence this adaptive law will also be referred to as gradient based. Furthermore, replacing $\Phi(z)$ with $\sigma(z)$ in equation 15 results in the baseline gradient based adaptive law for the case of unstructured uncertainty (case 2). For this case, the baseline adaptive law guarantees uniform ultimate boundedness of tracking error $e$.\cite{15,22,23} This adaptive law however, does not guarantee the boundedness of adaptive weights in both cases.

In order to ensure that adaptive weights remain bounded, Ioannou et al.\cite{20} suggested the use of a $\sigma$ modification term which adds damping to the adaptive law. Narendra et al.\cite{19} suggested the $e$ modification term which uses the norm of the tracking error to scale the added damping. These modifications have since been commonly used in adaptive control to ensure that the weights do not drift.\cite{18,21,23,25} However, these modifications do little to ensure convergence of adaptive weights, in fact, these modifications are often designed to bound the weights around a neighborhood of the origin. This can be counterproductive in situations where the adaptive element must estimate a steady-state trim value that is far away from origin. Finally, these modifications do little to ensure weight convergence.

If weight convergence does occur, then the linear, exponentially stable term in the tracking error dynamics of equation 8 dominates, and exponential tracking error convergence can be guaranteed. Furthermore, if weight convergence is achieved, then the performance of the system improves over the entire domain of the state space, thus incorporating long term learning and global error parameterization. The convergence of weights $\hat{W}$ to their ideal values $W^*$ can only be guaranteed with the gradient based baseline law of equation 15 if the signal $\Phi(z(t))$ is Persistently Exciting (PE). The persistence of excitation of a bounded vector signal has been defined by Tao as follows:\cite{21}

**Definition 1.** A bounded signal $\Phi(t)$ is defined to be persistently exciting if for all $t > t_0$ there exists $T > 0$ and $\gamma > 0$ such that

$$\int_{t_0}^{t+T} \Phi(t)\Phi^T(t)dt \geq \Gamma \gamma.$$  \hspace{1cm} (16)

Boyd and Sastry have shown that in the framework of MRAC, the persistency of excitation of $\Phi(x(t))$ can be guaranteed through the persistency of excitation of the tracking reference signal $r(t)$.\cite{21,25,26} Constant reference signals are not persistently exciting, nor are exponentially decaying reference signals. In fact Boyd and Sastry have shown that a persistently exciting signal must contain as many spectral lines as the dimension of the basis over the uncertainty over any time interval.\cite{25} In flight control applications, enforcing such persistent excitation in the control system may not be acceptable due to fuel and ride quality restrictions. Furthermore, since most flight controllers are event driven, it is hard to establish online whether the persistency of excitation condition is met.

On examining the baseline adaptive law of equation 15 we note that the law only uses instantaneous information for adaptation. The reliance on only instantaneous data is also reflected in the rank of $\hat{W}$, even though $\hat{W}$ is a matrix, its rank will be at-most one.\cite{4,27} This reliance on only instantaneous data is one reason why the baseline adaptive law must be persistently provided with information in order to guarantee exponential closed loop stability. A concurrent learning adaptive law on the other hand, uses recorded and current data concurrently for adaptation.

### III. Concurrent Adaptation on Current and Recorded Data

Concurrent learning relies on incorporating memory to guarantee exponential tracking error and weight error convergence in adaptive control. Concurrent learning formalizes the intuitive idea that if recorded data is sufficiently rich then exponential tracking error and weight error convergence can occur without the system states being persistently exciting.

Let $p$ denote the number of recorded data points, then letting for each recorded data point $j$, $e_j(t) = W^T(t)\Phi(z_j) - \Delta(z_j)$, a concurrent learning adaptive law that uses both recorded and current data concur-
rently for adaptation is chosen to have the following form

\[ \hat{W}(t) = -\Gamma_W \Phi(z(t))e^T(t)P - \sum_{j=1}^{p} \Gamma_W \Phi(z_j)e_j^T(t). \]  

(17)

The above update law augments traditional weight updates on current data \((-\Gamma_W \Phi(z(t))e^T(t)P, \text{equation 15})\) with weight updates on recorded data. The recorded data include the regressor vectors \(\Phi(z_j)\) which form a basis for the uncertainty \(\Delta(z_j)\) in equation 1, stored in a time-varying matrix referred to as the history-stack, and associated information (such as \(\hat{x}_j\)). Note that the concurrent learning update law of equation 17 exhibits switching in weight update dynamics as the elements of the history-stack change.

**Remark 1.** For evaluating the adaptive law of equation 17 the term \(\epsilon_j = W^T(t)\Phi(z_j) - \Delta(z_j)\) is required for the \(j^{th}\) data point where \(j \in [1, 2, \ldots, p]\). The model error \(\Delta(z_j)\) needs to be recorded along with \(\Phi(z_j)\) in the history-stack, and can be observed by using equation 5 noting that

\[ \Delta(z_j) = \hat{x}_j - \nu(z_j). \]  

(18)

Since \(\nu(z_j)\) is known, the problem of estimating system uncertainty can be reduced to that of estimation of \(\hat{x}\).

In cases where an explicit measurement for \(\hat{x}\) is not available, \(\hat{x}\) can be estimated using an implementation of a fixed point smoother.\(^{28}\) Fixed point smoothing uses a forward and a backward Kalman filter for accurately estimating \(\hat{x}_j\) in presence of noise and entails a selectable time delay before \(\epsilon_j\) can be calculated for that data point. This point stresses the benefit of using memory, as recorded states can undergo further processing to extract relevant information (modeling error in this case) in the background which can be used for adaptation. Furthermore, since \(\epsilon_j\) does not directly affect the tracking error at time \(t\), this delay does not adversely affect the instantaneous tracking performance of the controller. The details of this process are presented in the Appendix. Other methods, such as that suggested in\(^{29}\) and\(^{30}\) can also be used to estimate \(\hat{x}_j\).

Define the weight error as \(\hat{W} = W - W^*\), then the weight error dynamics for the case of can be written as

\[ \hat{W}(t) = -\Gamma_W \sum_{j=1}^{p} \Phi(z_j)\Phi^T(z_j)\hat{W}(t) - \Gamma_W \Phi(z(t))e^T(t)P. \]  

(19)

A. A Singular Value Maximizing Data Recording Algorithm

Equation 19 requires recorded data for concurrent adaptation. In this section, we describe an algorithm used to specifically select, record, and remove data points in the history stack. Let \(p \in \mathbb{N}\) denote the subscript of the last point stored. For ease of exposition, for a stored data point \(\hat{z}_j\), we let \(\Phi_j \in \mathbb{R}^n\) denote \(\Phi(z_j)\), which is the data point to be stored. We will let \(Z_t = [\Phi_1, \ldots, \Phi_p]\) denote the matrix containing the recorded state information in the history-stack at time \(t\). The \(j^{th}\) column of \(Z_t\) will be denoted by \(Z_t(:,j)\). It is assumed that the maximum allowable number of recorded data points is limited due to memory or processing power considerations. Therefore, we will require that \(Z_t\) has a maximum of \(\bar{p} \in \mathbb{N}\) columns, clearly, in order to be able to satisfy \(\text{rank}(Z_t) = m\), \(\bar{p} \geq m\). For the \(j^{th}\) data point, the associated model error \(\Delta(x_j)\) is assumed to be stored in the array \(\hat{\Delta}(\cdot,j) = \Delta(x_j)\).

The history stack is populated using an algorithm that aims to maximize the minimum singular value of the symmetric matrix \(\Omega = \sum_{j=1}^{p} \Phi(x_j)\Phi^T(x_j)\). The algorithm also ensures that any data point linearly independent of the data stored in the history stack is included in the history stack. At the initial time \(t_0\) the algorithm begins by setting \(Z_t(:,1) = \Phi(z(t_0))\). The algorithm then selects sufficiently different points for storage, a data point is considered to be sufficiently different if it is linearly independent of the points in the history stack or it is sufficiently different in the sense of the Euclidean norm of the last point stored. If the number of stored points increases the maximum allowable number, the algorithm seeks to incorporate new data points in such a way that the minimum singular value of \(Z_t\) is increased. To achieve this, the algorithm sequentially replaces every recorded data point in the history-stack with the current data point and stores the resulting minimum singular value in a variable. The algorithm then finds the maximum over these values, and accepts the new data point for storage into the history-stack (by replacing the corresponding existing point) if the resulting configuration results in an increase in the instantaneous minimum singular value of \(\Omega\). This results in an algorithm that ensures the minimum singular value of the history-stack matrix increases monotonically.
Algorithm 1: Singular Value Maximizing Algorithm for Recording Data Points

**Require:** \( p \geq 1 \)

if \( \frac{\|\Phi(x(\cdot)) - \Phi(x(t))\|}{\|\Phi(x(t))\|} \geq \epsilon \) or rank([\( Z_t \), \( \Phi(x(t)) \)]) > rank([\( Z_t \)]) then

\( p = p + 1 \)

\( Z_t(:, p) = \Phi(x(t)) \); store \( \Delta(:, p) = \dot{x}(t) - \nu(t) \)

end if

if \( p \geq \bar{p} \) then

\( T = Z_t \)

\( S_{old} = \min \text{SVD}(Z_t^T) \)

for \( j = 1 \) to \( p \) do

\( Z_t(:, j) = \Phi(x(t)) \)

\( S(j) = \min \text{SVD}(Z_t^T) \)

\( Z_t = T \)

end for

find max \( S \) and let \( k \) denote the corresponding column index

if max \( S > S_{old} \) then

\( Z_t(:, k) = \Phi(x(t)) \); store \( \Delta(:, k) = \dot{x}(t) - \nu(t) \)

\( p = p - 1 \)

else

\( p = p - 1 \)

\( Z_t = T \)

end if

end if

IV. Exponential Convergence Guarantees for the Case of Structured Uncertainty

In this section we present a concurrent learning adaptive controller which guarantees global exponential tracking error and parameter error convergence if a verifiable condition on the recorded data is met and the basis of the uncertainty is known. The key contribution is to include the effect of the singular value maximizing data recording algorithm in the proof.

**Theorem 1.** Consider the system in equation 1, the reference model in equation 6, the inverting controller of equation 2, the control law of equation 7, the case of structured uncertainty with the uncertainty given by equation 10, the switching weight update law of equation 17, assume that algorithm 1 is used to specifically select and record data points \( \Phi(z_j) \). Furthermore, assume that at \( t = t_0 \), the history stack is populated with pre-recorded elements such that rank(\( Z_0 \)) = \( m \), then the zero solution (e(t), \( \hat{W}(t) \)) \( \equiv (0, 0) \) of the closed loop system given by equations 8 and 19 is globally uniformly exponentially stable.

**Proof.** Let \( tr(.) \) denote the trace operator and consider the following quadratic functional

\[
V(e, \hat{W}) = \frac{1}{2} e^T P e + tr(\frac{1}{2} \hat{W}^T \Gamma \hat{W}^{-1} \hat{W}).
\]  
(20)

Note that \( V(0, 0) = 0 \) and \( V(e, \hat{W}) > 0 \) \( \forall (e, \hat{W}) \neq 0 \), therefore, \( V(e, \hat{W}) \) is a Lyapunov candidate. Let \( \xi = [e, vec(\hat{W})] \) where \( vec(.) \) is the operator that stacks the columns of a matrix into a vector, and let \( \lambda_{\min}(.) \) and \( \lambda_{\max}(.) \) denote operators that return the smallest and the largest eigenvalue of a matrix, then we have

\[
\frac{1}{2} \min(\lambda_{\min}(P), \lambda_{\min}(\Gamma W^{-1})) \|\xi\|^2 \leq V(e, \hat{W}) \leq \frac{1}{2} \max(\lambda_{\max}(P), \lambda_{\max}(\Gamma W^{-1})) \|\xi\|^2.
\]  
(21)

Differentiating 20 along the trajectory of 8 and the weight error dynamics of equation 19, and using the
Lyapunov equation (equation 9), we have

\[
\dot{V}(e, \bar{W}) = -\frac{1}{2} e^T Q e + e^T P (\nu_{ad} - \Delta) \\
+ tr(\bar{W}^T (\sum_{j=1}^{P} \Phi(z_j) \Phi^T(z_j) \bar{W} - \Phi(z) e^T P)).
\]  \hspace{1cm} (22)

Using equations 10 and 11 to note that \(\nu_{ad}(z(t)) - \Delta(z(t)) = \bar{W}^T(t) \Phi(z(t))\), canceling like terms and simplifying we have

\[
\dot{V}(e, \bar{W}) = -\frac{1}{2} e^T Q e - tr(\bar{W}^T (\sum_{j=1}^{P} \Phi(z_j) \Phi^T(z_j)) \bar{W}).
\]  \hspace{1cm} (23)

Let for all \(t\), \(\Omega(t) = \sum_{j=1}^{P} \Phi(z_j) \Phi^T(z_j)\), where \(\Phi(z_j)\) are the elements of the history-stack \(Z_t\) at time \(t\). Since rank\(Z_0) = m\), we have that \(\lambda_{\text{min}}(\Omega(0)) > 0\), hence we have

\[
\dot{V}(e, \bar{W}) \leq -\frac{1}{2} \lambda_{\text{min}}(Q) e^T e - \lambda_{\text{min}}(\Omega) tr(\bar{W}^T \bar{W}).
\]  \hspace{1cm} (24)

Note that algorithm 1 guarantees that \(\lambda_{\text{min}}(\Omega(t))\) is monotonically increasing, hence, letting \(\bar{\lambda} = \tilde{\lambda}_{\text{min}}(\Omega(0))\) we have

\[
\dot{V}(e, \bar{W}) \leq - \frac{\min(\lambda_{\text{min}}(Q), 2\bar{\lambda})}{\max(\lambda_{\text{max}}(P), \lambda_{\text{max}}(\Gamma_W^{-1}))} V(e, \bar{W}).
\]  \hspace{1cm} (25)

Hence, \(V(e, \bar{W})\) is a common Lyapunov function, therefore the uniform exponential stability of the solution \((e(t), \bar{W}(t)) \equiv (0, 0)\) of the closed loop system given by equations 8 and 19 is established. Since \(V(e, \bar{W})\) is radially unbounded, the result is global.

\[ \square \]

**Remark 2.** In the above theorem it is shown that a verifiable condition (rank\(Z_{t_0} = m\)) on the linear independence of the recorded data at \(t_0\) is sufficient to guarantee that the zero solutions of the tracking error and the parameter error are globally exponentially stable. It is important to note that the imposed rank-condition on the recorded data (rank\(Z) = m\)) is significantly different than a condition of persistency of excitation in the states. Firstly, this condition applies only to the recorded data, which is a small subset of all past states, whereas, the persistency of excitation condition applies to all past and future states. Secondly, since the rank of a matrix can be easily determined online, it is possible to verify whether this condition is met online, whereas it is impossible to determine whether a signal will be persistently exciting without knowing its future behavior. Hence, the rank-condition required to guarantee convergence when recorded data is concurrently used for adaptation with instantaneous data is less restrictive.

\[ \square \]

V. Concurrent Learning for Systems with Unstructured Uncertainty

When the structure of the uncertainty is unknown (case II in section II), a RBF NN can be used as the adaptive element. The universal approximation theorem for RBF NN guarantees that for a given number of neurons an ideal set of weights \(W^*\) exists such that the difference \(\tilde{\epsilon}(\hat{x}) = \Delta(\hat{x}) - W^* \sigma(\hat{x})\) is bounded by \(\tilde{\epsilon}\) (see equation 14). Similar to the structured uncertainty case, the baseline adaptive law \(\hat{W} = -\Gamma_W \sigma(\hat{x}) e^T P\) only guarantees that the weights approach the ideal weights if the plant states are persistently exciting. In this section we present a concurrent learning neuro-adaptive law that uses recorded data concurrently with instantaneous data to guarantee that the tracking error remain bounded within a compact neighborhood of zero and the adaptive weights approach and remain bounded in a compact neighborhood of the ideal weights.

In the following, we let \(\kappa\) denote a small positive constant.

**Theorem 2.** Consider the system in equation 1, the reference model in equation 6, the inverting controller of equation 2, the control law of equation 7. Assume that the structure of the plant uncertainty is unknown and the uncertainty is approximated over a compact domain \(D\) using a Radial Basis Function NN as in equation 14 with \(\tilde{\epsilon} = \sup_{x \in D} \|\tilde{\epsilon}(z)\|\). Furthermore, assume that algorithm 1 is used to specifically select and
record data points \( \Phi(z_j) \), and consider the following switching concurrent learning weight update law:

\[
\dot{\hat{W}} = \begin{cases} 
-\Gamma_W \sigma(z) e^T P - \sum_{j=1}^{p} \Gamma_W \sigma(z_j) e_j^T - \kappa \hat{W} & \text{if } \text{rank}(Z_t) < q \\
-\Gamma_W \sigma(z) e^T P - \sum_{j=1}^{p} \Gamma_W \sigma(z_j) e_j^T & \text{if } \text{rank}(Z_t) = q
\end{cases}
\] (26)

Let \( T \) denote the time at which \( \text{rank}(Z_t) = q \), let \( B_\alpha \) be the largest compact ball in \( D \), and assume \((x(0)) \in B_\alpha\). Define \( \delta = \max(\beta, \frac{2||e||^2}{\lambda_{\min}(Q)^{\kappa}} + \frac{2||e^*||^2}{\lambda_{\min}(Q)^{\kappa}} + \frac{1}{\kappa} ||e^*||^2 + \frac{1}{\kappa} ||e^*||^2 + \frac{1}{\kappa} ||P^*e^*||^2)\), and assume that \( D \) is sufficiently large such that \( m = \alpha - \delta \) is a positive scalar. If the exponential input \( r(t) \) is such that the state \( x_{rm}(t) \) of the bounded input bounded output reference model of equation 6 remains bounded in the compact ball \( B_m = \{ x_{rm} : ||x_{rm}|| \leq m \} \) for all \( t \geq 0 \) then the solution \((e(t), \hat{W}(t)) \) of the closed loop system of equations 8 and 26 is uniformly ultimately bounded.

**Proof.** Consider the following positive definite and radially unbounded function

\[
V(e, \hat{W}) = \frac{1}{2} e^T P e + tr\left(\frac{1}{2} \hat{W}^T \Gamma_W^{-1} \hat{W}\right).
\] (27)

Note that \( V(0, 0) = 0 \) and \( V(e, \hat{W}) \geq 0 \ \forall (e, \hat{W}) \neq 0 \) hence 27 is a Lyapunov like candidate.\(^{32}\) Consider first the case when \( \text{rank}(Z_t) \leq 1 \), then noting that \( \nu_{ad}(x_j) + \Delta(x_j) = \hat{W}^T \sigma(x_j) + e(x_j) \) the weight error dynamics take the following form

\[
\dot{\hat{W}}(t) = -\Gamma(\sum_{j=1}^{p} \sigma(x_j)e^T(z_j) + \kappa I) \hat{W}(t) - \sum_{j=1}^{p} \sigma(x_j)e^T(z_j) - \Gamma_W \sigma(z(t))e^T(t)P - \kappa \hat{W}^* 
\] (28)

Differentiating 27 along the trajectory of 8, 28, using the Lyapunov equation (equation 9), canceling like terms, noting that \( \nu_{ad}(x) - \Delta(x) = \hat{W}^T \sigma(x) + e(x) \), and simplifying we have

\[
\dot{V}(e, \hat{W}) = -\frac{1}{2} e^T Q e - tr(\hat{W}^T(\sum_{j=1}^{p} \sigma(x_j)e^T(z_j) + \kappa I) \hat{W}) + \sum_{j=1}^{p} \sigma(x_j)e^T(z_j) - \kappa \hat{W}^* + e^T \bar{P} \bar{e}(z).
\] (29)

Simplifying further, noting that \( \|\sigma(z(t))\| \leq \sqrt{q} \) due to the definition of RBF (equation 13)

\[
\dot{V}(e, \hat{W}) \leq -\frac{1}{2} \lambda_{\min}(Q)^{\kappa} e^T e - \kappa \|\hat{W}\|^2 + \kappa \|\hat{W}^*\| \|\hat{W}\| + p \|\hat{W}\| \bar{e} \sqrt{q} + \|e^T P\| \bar{e}.
\] (30)

Let \( c_1 = \|P\| \bar{e}, c_2 = \kappa \|\hat{W}^*\|, c_3 = p \bar{e} \sqrt{q} \), then

\[
\dot{V}(e, \hat{W}) \leq -\|e\|\left(\frac{1}{2} \lambda_{\min}(Q) \|e\| - c_1 \right) - \|\hat{W}\| \left(\kappa \|\hat{W}^*\| - c_2 - c_3 \right) 
\] (31)

Hence \( \dot{V}(e, \hat{W}) \leq 0 \) when \( \|e\| \geq \frac{2c_1}{\lambda_{\min}(Q)} \) and \( \|\hat{W}\| \geq \frac{\kappa c_3 - \kappa c_2 - c_1}{\kappa} \), hence \((e(t), \hat{W}(t))\) are bounded. Therefore the set \( \Theta_d = \{(e, \hat{W}) : \|e\| + \|\hat{W}\| \leq \frac{2c_1}{\lambda_{\min}(Q)} + \frac{\kappa c_3 - \kappa c_2 - c_1}{\kappa} \} \) is positively invariant for all \( t \leq T \).

Consider now that at time \( T \) the history-stack satisfies \( \text{rank}(Z_t) = q \), and note that \((e(T), \hat{W}(T))\) are bounded due to the previous result. In this case, the weight dynamics become

\[
\dot{\hat{W}}(t) = -\Gamma(\sum_{j=1}^{p} \sigma(x_j)\sigma(z_j)\hat{W}(t) - \sum_{j=1}^{p} \sigma(x_j)e^T(z_j) - \Gamma_W \sigma(z(t))e^T(t)P.
\] (32)

Let \( \Omega(t) = Z_t Z_t^T \), and note that \( \Omega(t) \) is positive definite for all \( t \) once \( \text{rank}(Z_t) = q \) due to algorithm 1. Algorithm 1 guarantees that \( \sigma_{\min}(Z_t) \) is monotonically increasing for all \( t \geq T \). Let \( \lambda_{\min}(\Omega(t)) \) denote the minimum eigenvalue of \( \Omega \) at time \( t \) then \( \lambda_{\min}(\Omega(t)) \geq \lambda_{\min}(\Omega(T)) \). Hence,

\[
\dot{V}(e, \hat{W}) \leq \|e\|\left(-\frac{1}{2} \lambda_{\min}(Q) \|e\| + c_1 \right) + \|\hat{W}\|\left(-\lambda_{\min}(\Omega(T)) \|\hat{W}\| + c_2 \right).
\] (33)
Hence, if \( \|e\| > \frac{2c}{\lambda_{\min}(Q)} \) and \( \|\hat{W}\| > \frac{c_s}{\lambda_{\min}(Q(T))} \), we have that \( \hat{V}(e, \hat{W}) < 0 \). Therefore the set \( \Omega_\delta = \{ (e, \hat{W}) : \|e\| + \|\hat{W}\| \leq \frac{2c}{\lambda_{\min}(Q)} + \frac{c_s}{\lambda_{\min}(Q(T))}\} \) is positively invariant.

Let \( \delta = \max(\beta, \frac{2c_p}{\lambda_{\min}(Q)}, \frac{2c_p}{\lambda_{\min}(Q(T))}, \frac{c_s}{\lambda_{\min}(Q(T))}) \), and \( m = a - \delta \). Hence, if the exogenous input \( r(t) \) is such that the state \( x_{nm}(t) \) of the bounded input bounded output reference model of equation 6 remains bounded in the compact ball \( B_m = \{ x_{nm} : \|x_{nm}\| \leq m \} \) for all \( t \geq 0 \), then \( x(t) \in D \forall t \) hence the NN approximation of equation 14 holds and the solution of the closed loop system of equations 8 and 26 \((e(t), \hat{W}(t))\) is ultimately bounded.

\[ \square \]

**Remark 3.** The term \( \kappa W \) is Ioannou’s \( \sigma \) modification, and it is required to guarantee the boundedness of weights before the rank-condition \((\text{rank}(Z_t) = q)\) is satisfied. Once the rank condition is satisfied, the concurrent learning adaptive law no longer requires the \( \sigma \) modification, or any other modification, to guarantee the boundedness of weights. Furthermore, if only a \( \sigma \) modification based adaptive law is used, that is, if the adaptive law is given by \(-\Gamma_W \sigma(z)e^T P - \kappa W\), then it can be shown that the set \( \Omega_\sigma = \{ (e, \hat{W}) : \|e\| + \|\hat{W}\| \leq \frac{2c_p}{\lambda_{\min}(Q)} + ||W^*||\} \) is positively invariant. Hence, if the history-stack is such that \( \frac{p_c^{\sqrt{q}}}{\lambda_{\min}(Q(T))} \leq ||W^*|| \) then the ultimate bounds achieved by using concurrent learning will be smaller than those achieved by using only \( \sigma \) modification, thereby improving the tracking performance of the adaptive controller.

**Remark 4.** If \( r(t) \) is such that the system states are exciting over a finite time interval \([0, T]\) then algorithm 1 guarantees that \( \text{rank}(Z_t) = q \) for all \( t \geq T \). Hence, the concurrent learning adaptive law of Theorem 2 requires only that the states be exciting over a finite interval to guarantee the ultimate boundedness of the weights around their ideal values. This can be contrasted with the ultimate bounds achieved when using other modifications that guarantee the boundedness of weights, such as projection based adaptation, \( \sigma \), or \( e \) modification which only guarantee that the weights stay bounded around a preselected value, usually set to 0 in the case of \( \sigma \) or \( e \) modification.

**Remark 5.** It can be shown that the adaptive law of theorem 2 also guarantees the ultimate boundedness \((e, \hat{W}(t))\) in presence of parameter variations.

VI. Simulation Results: Trajectory Tracking in Presence of Wing Rock Dynamics

Modern highly swept-back or delta wing fighter aircraft are susceptible to slingly damped oscillations in roll angle known as “Wing Rock”. Wing rock often occurs at low speeds and at high angle of attack, conditions commonly encountered when the aircraft is landing (see reference 33 for a detailed discussion of the wing rock phenomena). Hence, precision control of the aircraft in the presence of wing rock dynamics is critical to ensure safe landing. In this section we use concurrent learning control to track a sequence of roll commands in the presence of wing rock dynamics. Let \( \phi \) denote the roll angle of an aircraft, \( p \) denote the roll rate, \( \delta_a \) denote the aileron control input, then a model for wing rock dynamics is:

\[
\begin{align*}
\dot{\phi} &= p \\
\dot{p} &= \delta_a + \Delta(x).
\end{align*}
\]

where \( \Delta(x) = W_0 + W_1 \phi + W_2 p + W_3 |\phi||p| + W_4 |p|^2 + W_5 \phi^3 \). The parameters for wing rock motion are adapted from references 35–37, they are \( W_1 = 0.2314, W_2 = 0.6918, W_3 = -0.6245, W_4 = 0.0095, W_5 = 0.0214. \) In addition to these parameters, a trim error is introduced by setting \( W_0 = 0.8 \). The linear part of the control law is given by \( u_{\text{rad}} = -1.5 \dot{\phi} - 1.3p \), a second order reference model with natural frequency of 1 rad/sec and damping ratio of 0.5 is chosen, and the learning rate is set to \( \Gamma_W = 2 \). A simple inversion model has the form \( \nu = \delta_a \). The adaptive controller uses the control law of equation 7. The simulation uses a time-step of 0.05 seconds. The concurrent learning controller uses the the weight update law of theorem 1, while the adaptive controller without concurrent learning uses the weight update law of equation 15.

A. Concurrent Learning Control of Wing Rock Dynamics

It is assumed that no previously recorded data points are available, and the concurrent learning adaptive controller actively populates the history stack through the simulation. An attitude tracking command of 1 deg
and $-1$ deg is commanded between 15 to 17 seconds and 25 to 27 seconds respectively. The history stack is restricted to contain 30 recorded data points. New data points are selected and old data points replaced using the minimum singular value maximizing algorithm 1. The initial conditions are set to $\phi(0) = 1.2$ deg and $p(0) = 1$ deg/sec. Figure 2 compares the trajectory tracking performance of the adaptive controller of equation 7 with and without concurrent learning. The tracking performance is seen to improve significantly with concurrent learning, particularly when tracking the two steps in altitude. Figure 3 explicitly compares the tracking errors and the control commands for both cases. It can be seen that with concurrent learning, the tracking error rapidly approaches zero. Furthermore, the commanded control inputs remain comparable in magnitude. This is expected, as the improved performance of concurrent learning is through better estimation of the uncertainty. Figure 5 compares the evolution of the weights with and without concurrent learning. The weights rapidly converge to their true values with concurrent learning.

![Figure 2. Comparison of tracking performance of adaptive controller with and without concurrent learning. Note the significantly improved tracking error performance with concurrent learning.](image)

### VII. Conclusion

We showed through Lyapunov based argument that a singular value maximizing algorithm combined with a concurrent learning adaptive controller results in exponential tracking error and weight error convergence to zero. The integration of the data recording algorithm with the concurrent learning adaptive law makes the closed loop dynamics switching in nature. The key point to note here is that the exponential convergence is subject to a verifiable condition on linear independence of the recorded data, and exponential convergence is guaranteed without requiring persistency of excitation. It can be expected that data recording/removal algorithm plays a pivotal role in determining the rate of convergence. We formalized this by showing that the rate of convergence is directly proportional to the minimum singular value of the matrix containing the recorded data.

### Acknowledgments

This work was supported in part by NSF ECS-0238993 and NASA Cooperative Agreement NNX08AD06A.
Figure 3. Comparison of tracking errors of adaptive controller with and without concurrent learning. Note that the tracking error with concurrent learning is effectively zero when tracking the second command.

References


Figure 4. Comparison of evolution of adaptive weights with and without concurrent learning. Note that the weights of the concurrent learning adaptive controller converge to the true weights, while the weights without concurrent learning do not converge.


Figure 5. Comparison of evolution of the Lyapunov function \( V \) with and without concurrent learning. Note that with concurrent learning, \( V \) decays to zero. The figure also shows that the singular value maximizing algorithm 1 guarantees that points are only recorded when the minimum singular value increases.

Appendix A: Optimal Fixed Point Smoothing

Numerical differentiation for estimation of state derivatives suffers from high sensitivity to noise. An alternate method is to use a Kalman filter based approach. Let \( x \), be the state of the system and \( \dot{x} \) be its first derivative, and consider the following system:

\[
\begin{pmatrix}
\dot{x} \\
\dot{\dot{x}}
\end{pmatrix}
= \begin{bmatrix}
0 & 1 \\
0 & 0
\end{bmatrix}
\begin{pmatrix}
x \\
\dot{x}
\end{pmatrix}
\tag{36}
\]

Suppose \( x \) is available as sensor measurement, then an observer in the framework of a Kalman filter can be designed for estimating \( \dot{x} \) from available noisy measurements using the above system. Optimal Fixed Point Smoothing is a non real time method for arriving at a state estimate at some time \( t \), where \( 0 \leq t \leq T \), by using all available data up to time \( T \). Optimal smoothing combines a forward filter which operates on all data before time \( t \) and a backward filter which operates on all data after time \( t \) to arrive at an estimate of the state that uses all the available information. For ease of implementation on modern avionics, we present the relevant equations in the discrete form. Let \( \hat{x}_{(k|k)} \) denote the estimate of the state \( x = [ x \ \dot{x} ]^T \), let \( Z_k \) denote the measurements, (−) denote predicted values, and (+) denote corrected values, \( dt \) denote the discrete time step, \( Q \) and \( R \) denote the process and measurement noise covariance matrices respectively, while \( P \) denotes the error covariance matrix. Then the forward Kalman filter equations can be given as follow:

\[
\Phi_k = e^{\begin{bmatrix}
0 & 1 \\
0 & 0
\end{bmatrix} dt},
\tag{37}
\]

\[
Z_k = \begin{bmatrix}
1 & 0
\end{bmatrix}
\begin{pmatrix}
x \\
\dot{x}
\end{pmatrix},
\tag{38}
\]

\[
\hat{x}_k(−) = \Phi_k \hat{x}_{k−1},
\tag{39}
\]

\[
P_k(−) = \Phi_k P_{k−1} \Phi_k^T + Q_k,
\tag{40}
\]

\[
K_k = P_k(−)H_k^T [H_k P_k(−) H_k^T + R_k]^{-1},
\tag{41}
\]

\[
\hat{x}_k(+) = \hat{x}_k(−) + K_k [Z_k - H_k \hat{x}_k(−)],
\tag{42}
\]

\[
P_k(+) = [I - K_k H_k] P_k(−).
\tag{43}
\]

American Institute of Aeronautics and Astronautics
The smoothed state estimate can be given as:

$$\hat{x}_{k|N} = \hat{x}_{k|N-1} + B_N[\hat{x}_N(+) - \hat{x}_N(-)],$$

(44)

where $\hat{x}_{k|k} = \hat{x}_k$. 