

Concurrent Learning Adaptive Control of Linear Systems with Exponentially Convergent Bounds

Girish Chowdhary*, Tansel Yucelen, Maximillian Mühlegg, and Eric N. Johnson

Daniel Guggenheim School of Aerospace Engineering at the Georgia Institute of Technology, Atlanta GA, 30318

SUMMARY

Concurrent learning adaptive controllers, which use recorded and current data concurrently for adaptation, are developed for model reference adaptive control of uncertain linear dynamical systems. We show that a verifiable condition on the linear independence of the recorded data is sufficient to guarantee global exponential stability. We use this fact to develop exponentially decaying bounds on the tracking error and weight error, and estimate upper bounds on the control signal. These results allow the development of adaptive controllers that ensure good tracking without relying on high adaptation gains, and can be designed to avoid actuator saturation. Simulations and hardware experiments show improved performance. Copyright © 2011 John Wiley & Sons, Ltd.

Received ...

KEY WORDS: Model Reference Adaptive Control, Persistency of Excitation, Uncertain Multivariable Linear Dynamical Systems

1. INTRODUCTION

Various physical systems can be represented by using multivariable linear dynamical systems. For example, it is well known that aircraft and rotorcraft can be adequately modeled by linear systems around specific trim conditions (such as hovering flight, forward flight etc. [1], [2]). This is in fact a well established practice in the aerospace industry. However, the linear models can have significant uncertainties introduced in the modeling or the identification process. Model Reference Adaptive Control (MRAC) is a widely studied approach that can be used to design controllers in presence of such uncertainties [3, 4, 5]. In MRAC of uncertain multivariable linear dynamical systems, the design objective is to make the linear system behave like a chosen reference model. The underlying assumption is that a set of ideal linear weights exists that guarantees this design objective can be met. This assumption is often known as the *matching condition* (see for example [3], [5]). However, most MRAC methods concentrate on driving the tracking error to zero, and are not concerned with driving the weight error to zero. If, along with driving the tracking error to zero, the adaptive controller does drive the weights to their ideal values as prescribed by the matching conditions (that is, if the weights converge), then the closed loop system's behavior approaches exponentially fast to the reference model, and the desired transient response and stability properties of the chosen reference model are dynamically recovered. In fact, it is an established fact that exponential convergence of the weights to their true values results in exponential tracking error convergence [6, 7]. However, many classical and recent adaptive controllers that use only instantaneous data for adaptation (see

*Correspondence to: Daniel Guggenheim School of Aerospace Engineering at the Georgia Institute of Technology, Atlanta GA, Girish.Chowdhary@gatech.edu

†Girish Chowdhary, Tansel Yucelen, and Eric Johnson are with Georgia Institute of Technology, Maximillian Mühlegg is with the Technical University of Munich

[8, 4, 9, 3, 10, 7] and references therein), require that the system states be Persistently Exciting (PE) to guarantee exponential weight convergence, and hence to guarantee exponential stability of the entire closed loop system, which consists of both the tracking error and weight error dynamics. Boyd and Sastry have shown that the condition on PE states can be directly related to a condition on the spectral properties of the exogenous reference input [11]. However, enforcing PE through exogenous excitation is not always feasible, particularly in applications which require high precision or smooth operation. Furthermore, since it is hard to predict the future behavior of systems such as aircraft, it can often be difficult to monitor online whether a signal will remain PE. Hence exponential weight convergence, and therefore exponential tracking error convergence, cannot be guaranteed in many adaptive control applications.

In this paper we describe a novel approach to guarantee exponential stability of MRAC of uncertain linear multivariable dynamical systems by utilizing the idea of concurrent learning [12, 13]. Particularly, we show that a concurrent learning model reference adaptive controller, that uses both current and past data concurrently for adaptation, can guarantee global exponential stability of the zero solution of the closed loop dynamics subject to a verifiable condition on linear independence of the recorded data; *without requiring PE states*. That is, it can guarantee that the tracking error and the weight error dynamics simultaneously converge to zero exponentially fast as long as the system states are exciting over a *finite* period of time. Furthermore, we show that the guaranteed exponential stability results in guaranteed bound on how large the transient tracking error can be, and that this bound reduces exponentially fast in time. This bound can be used to design controllers that do not exceed pre-defined limits and do not saturate. Therefore, this result has significant importance in verification and validation of adaptive control systems. These results show that the inclusion of memory can significantly improve the performance and stability guarantees of adaptive controllers. Finally, these results compliment our previous work in concurrent learning (see for example [12], [13], [14]), and extend concurrent learning to adaptive control to the case where the system dynamics can be represented by switching multivariable linear systems. The performance of a concurrent learning adaptive controller is compared with a classical adaptive controller in an exemplary simulation study. Furthermore, hardware experiments are conducted on a 3 degree of freedom helicopter to validate the theory.

The organization of this paper is as follows, in Section 2 we pose the classical problem of MRAC of linear systems. In Section 3 we present the novel concurrent learning adaptive controller for linear systems, and establish its properties using Lyapunov analysis. In section 4 a connection between the presented method and Ioannou's classical σ -modification is established. In Section 5 the effect of estimation errors is analyzed. In Section 6 we present the results of an exemplary simulation study. The approach is validated through hardware based experiments in Section 7. The paper is concluded in Section 8.

In this paper, $f(t)$ represents a function of time t . Wherever appropriate, we will drop the argument t consistently over an entire for ease of exposition. The operator $|\cdot|$ denotes the absolute value of a scalar and the operator $\|\cdot\|$ denotes the Euclidian norm of a vector. Furthermore, for a vector $x(t) \in \mathbb{R}^n$, the infinity norm is defined as $\|x(t)\|_\infty = \max_{i=1,\dots,n} |x_i(t)|$, and the L_∞ norm of the vector signal x can be defined as $\|x\|_{L_\infty} = \max_{i=1,\dots,n} (\sup_{t \geq 0} |x_i(t)|)$. The operator $\text{vec}(\cdot)$ stacks the columns of a matrix into a vector, $\text{tr}(\cdot)$ denotes the trace operator, and operators $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ return the smallest and the largest eigenvalue of a matrix.

2. MODEL REFERENCE ADAPTIVE CONTROL OF LINEAR SYSTEMS

In this section we review results on classical MRAC of uncertain multivariable (inclusive of multi-input, multi-output) linear dynamical systems (see for example [8, 4, 9, 3, 5]).

2.1. The Classical Linear MRAC Problem

Let $x(t) \in \mathbb{R}^n$ be the state vector, let $u(t) \in \mathbb{R}^m$ denote the control input, and consider the following linear time-invariant system

$$\dot{x}(t) = Ax(t) + Bu(t), \quad (1)$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$. We assume that the pair (A, B) is controllable and that B has full column rank. For the case when $m \leq n$, this condition is normally satisfied by physical systems. The assumption can be restrictive when $m > n$, in this case, it can be relaxed by using matrix inverse and pseudoinverse approaches, constrained control allocation, pseudocontrols, or daisy chaining [15, 16, 17], to reduce the dimension of the control input vector. We assume that $u(t)$ is restricted to the class of admissible control inputs consisting of measurable functions, and $x(t)$ is available for full state feedback.

A chosen reference model that characterizes the desired closed-loop response of the system is given by

$$\dot{x}_{rm}(t) = A_{rm}x_{rm}(t) + B_{rm}r(t), \quad (2)$$

where $A_{rm} \in \mathbb{R}^{n \times n}$ is Hurwitz. Furthermore, $r(t)$ denotes a bounded, piecewise continuous, reference signal. An adaptive control law consisting of a linear feedback part $u_{pd}(t) = K^T(t)x(t)$, and a linear feedforward part $u_{rm}(t) = K_r^T(t)r(t)$ with time varying weights $K(t) \in \mathbb{R}^{n \times m}$ and $K_r(t) \in \mathbb{R}^{1 \times m}$ is proposed to have the following form

$$u(t) = u_{rm}(t) + u_{pd}(t). \quad (3)$$

Substituting (3) in (1) we have

$$\dot{x}(t) = (A + BK^T(t))x(t) + BK_r^T(t)r(t). \quad (4)$$

The design objective is to have (4) behave as the chosen reference model of (2). To that effect, we introduce the following matching conditions:

Assumption 1

There exists $K^* \in \mathbb{R}^{n \times m}$ and $K_r^* \in \mathbb{R}^{1 \times m}$ such that

$$A + BK^{*T} = A_{rm} \quad (5)$$

$$BK_r^{*T} = B_{rm}. \quad (6)$$

Adding and subtracting $BK^{*T}x(t)$ and $BK_r^{*T}r(t)$ in (4) and letting $\tilde{K} = K - K^*$ and $\tilde{K}_r = K_r - K_r^*$ we have

$$\dot{x}(t) = A_{rm}x(t) + B_{rm}r(t) + B\tilde{K}^T(t)x(t) + B\tilde{K}_r^T(t)r(t). \quad (7)$$

Note that if $\tilde{K} = 0$ and $\tilde{K}_r = 0$, then $\dot{x} = A_{rm}x(t) + B_{rm}r(t)$ and the design objective of having the system behave as the reference model is satisfied. Furthermore, defining the tracking error as $e(t) \triangleq x(t) - x_{rm}(t)$ yields

$$\dot{e}(t) = A_{rm}e(t) + B\tilde{K}^T(t)x(t) + B\tilde{K}_r^T(t)r(t). \quad (8)$$

As noted, the tracking error converges to zero exponentially fast if $\tilde{K} = 0$ and $\tilde{K}_r = 0$, since A_{rm} is Hurwitz. It follows from converse Lyapunov theory that there exists a unique positive definite $P \in \mathbb{R}^{n \times n}$ satisfying the Lyapunov equation

$$A_{rm}^T P + P A_{rm} + Q = 0 \quad (9)$$

for any positive definite matrix $Q \in \mathbb{R}^{n \times n}$.

Let $\Gamma_x > 0$ and $\Gamma_r > 0$ denote positive definite learning rates, and consider a well known adaptive law for MRAC of linear systems which uses only instantaneous data for adaptation:

$$\dot{K}(t) = -\Gamma_x x(t)e^T(t)PB, \quad (10)$$

$$\dot{K}_r(t) = -\Gamma_r r(t)e^T(t)PB. \quad (11)$$

It is a classical result that the adaptive law of (10)-(11) will guarantee the weight error $\tilde{K}(t) \rightarrow 0$ and $\tilde{K}_r(t) \rightarrow 0$ exponentially, and the tracking error $e(t) \rightarrow 0$ exponentially if and only if the system

states are persistently exciting [3], [9], [18], [19], [4]. Furthermore, it is also well known that the classical adaptive law of (10) does not guarantee that the weights remain bounded in a predefined domain unless further modifications are made. Commonly used modifications to the adaptive law include σ modification of Ioannou [18], e modification of Narendra [8], or artificial restriction of the weights using parameter projection [3], [20].

Various equivalent definitions of excitation and the persistence of excitation of a bounded vector signal exist in the literature [9], [4], we will use the definitions proposed by Tao in [3]:

Definition 1

A bounded vector signal $x(t)$ is exciting over an interval $[t, t + T]$, $T > 0$ and $t \geq t_0$ if there exists $\gamma > 0$ such that

$$\int_t^{t+T} x(\tau)x^T(\tau)d\tau \geq \gamma I. \quad (12)$$

Definition 2

A bounded vector signal $x(t)$ is persistently exciting if for all $t > t_0$ there exists $T > 0$ and $\gamma > 0$ such that

$$\int_t^{t+T} x(\tau)x^T(\tau)d\tau \geq \gamma I. \quad (13)$$

As an example, consider that in the two dimensional case, vector signals containing a step in every component are exciting, but not persistently exciting; whereas the vector signal $x(t) = [\sin(t), \cos(t)]$ is persistently exciting. In the MRAC framework discussed above, Boyd and Sastry have shown that if the exogenous input $r(t)$ contains at least n spectral lines, then the system states $x(t)$ are persistently exciting [11].

3. CONCURRENT LEARNING FOR EXPONENTIAL CONVERGENCE IN MRAC OF LINEAR SYSTEMS

The key idea in concurrent learning is to use recorded data concurrently with current data for adaptation. Intuitively, one can argue that if the recorded data is sufficiently rich, then convergence should occur without requiring persistency of excitation. Here, we formalize this argument, and show that a verifiable condition on the linear independence of the recorded data is sufficient to guarantee exponential stability of the tracking error and weight error dynamics.

3.1. A Concurrent Learning Adaptive Controller

Let (x_j, r_j) denote the j^{th} recorded data pair of the state and reference signal. Recall that since we have assumed B is full column ranked, $(B^T B)^{-1}$ exists, and define for each recorded data point the error variables $\epsilon_{K_j}(t) \in \mathbb{R}^m$, and $\epsilon_{K_{r_j}}(t) \in \mathbb{R}^m$ as follows:

$$\epsilon_{K_j}(t) = (B^T B)^{-1} B^T (\dot{x}_j - A_{rm} x_j - B_{rm} r_j - B \epsilon_{K_{r_j}}(t)), \quad (14)$$

and

$$\epsilon_{K_{r_j}}(t) = K_r^T r_j - (B^T B)^{-1} B^T B_{rm} r_j. \quad (15)$$

The concurrent learning weight update law is then given by:

$$\dot{K}(t) = -\Gamma_x(x(t)e^T(t)PB + \sum_{j=1}^p x_j \epsilon_{K_j}^T(t)), \quad (16)$$

$$\dot{K}_r(t) = -\Gamma_r(r(t)e^T(t)PB + \sum_{j=1}^p r_j \epsilon_{K_{r_j}}^T(t)). \quad (17)$$

The above concurrent learning law can be evaluated if the first derivative of the state (\dot{x}_j) for the j^{th} recorded data point is known. Note that one does not need to know the state derivative

$\dot{x}(t)$ at the current time t , rather, only the state derivative of a data point recorded in the past. If measurements are not available, \dot{x}_j can be estimated using a fixed point optimal smoother which uses a forward and a backward Kalman filter to accurately estimate \dot{x}_j in presence of noise. This point stresses the benefit of using memory, as recorded states can undergo further processing to extract relevant information (projected weight error in this case) in the background which can be used for adaptation. Furthermore, since ϵ_j does not directly affect the tracking error at time t , this delay does not adversely affect the instantaneous tracking performance of the controller. Details of this process can be found in [13].

Furthermore, note that by rearranging (7), we see that $\epsilon_{K_j}(t) = \tilde{K}^T(t)x_j$, and by rearranging (2) we see that $\epsilon_{K_r j}(t) = \tilde{K}_r^T(t)r_j$. Hence, in both cases, the m dimensional terms in equations (14) and (15) contain the information of the m by n dimensional weight error \tilde{K} and the m dimensional weight error \tilde{K}_r respectively. The key point to note here is that although the concurrent learning law in (16) will be affected by the weight errors, explicit knowledge of the weight errors (and hence the ideal weights K^* and K_r^*) is not required for evaluating (16). Noting this fact, and recalling $\tilde{K}(t) = K(t) - K^*$, and $\tilde{K}_r = K_r - K_r^*$, the weight error dynamics can be written as a function of the tracking error and the weight error as follows

$$\dot{\tilde{K}}(t) = -\Gamma_x(x(t)e^T(t)PB + \sum_{j=1}^p x_j x_j^T \tilde{K}(t)), \quad (18)$$

$$\dot{\tilde{K}}_r(t) = -\Gamma_r(r(t)e^T(t)PB + \sum_{j=1}^p r_j r_j^T \tilde{K}_r(t)). \quad (19)$$

3.2. Data Recording

Let (x_j, r_j) denote the j^{th} recorded data pair and assume that the data are contained in history stack matrices $X_k = [x_1, x_2, \dots, x_p]$ and $R_k = [r_1, r_2, \dots, r_p]$, where the subscript $k \in \mathbb{N}$ is incremented every time a history stack is updated. Due to memory considerations, it is assumed that a maximum of m data points can be stored. The idea is to update the history stack by adding data points to empty slots or by replacing an existing point if no empty slot is available to maximize the minimum singular value of X_k , $\sigma_{\min}(X_k)$. This is motivated by the fact that the convergence rate of the adaptive controller is directly proportional to $\sigma_{\min}(X_k)$, this relationship is further explored in proof of Theorem 1. This can be achieved using the singular value maximizing data recording algorithm described below.

3.3. Stability Analysis

The following theorem contains the main contribution of this paper.

Theorem 1

Consider the system in (1), the control law of (3), and let $p \geq n$ be the number of recorded data points. Let $X_k = [x_1, x_2, \dots, x_p]$ be the history stack matrix containing recorded states, and $R_k = [r_1, r_2, \dots, r_p]$ be the history stack matrix containing recorded reference signals. Assume that over a finite interval $[0, T]$ the exogenous reference input $r(t)$ is exciting, the history stack matrices are empty at $t = 0$, and are consequently updated using Algorithm 1. Then, the concurrent learning weight update laws of (16) and (17) guarantee that the zero solution $(e(t), \tilde{K}(t), \tilde{K}_r(t)) \equiv 0$ is globally exponentially stable.

Proof

Consider the following positive definite and radially unbounded Lyapunov candidate

$$V(e, \tilde{K}, \tilde{K}_r) = \frac{1}{2}e^T P e + \frac{1}{2}\text{tr}(\tilde{K}^T \Gamma_x^{-1} \tilde{K}) + \frac{1}{2}\text{tr}(\tilde{K}_r^T \Gamma_r^{-1} \tilde{K}_r). \quad (20)$$

Algorithm 1 Singular Value Maximizing Algorithm for Recording Data Points

```

if  $\frac{\|x(t) - x_l\|^2}{\|x(t)\|^2} \geq \epsilon$  or  $\text{rank}([X_k, x(t)]) > \text{rank}([X_k])$  then
  if  $p < \bar{p}$  then
     $p = p + 1$ , set  $l = p$ 
     $X_k(:, p) = x(t)$ ; {store  $r(t)$ ,  $\dot{x}(t)$ , initiate smoother for estimating  $\dot{x}(t)$  if measurement not available}
  else
     $T = X_k$ 
     $S_{old} = \min \text{SVD}(X_k^T)$ 
    for  $j = 1$  to  $p$  do
       $X_k(:, j) = x(t)$ 
       $S(j) = \min \text{SVD}(X_k^T)$ 
       $X_k = T$ 
    end for
    find max  $S$  and let  $l$  denote the corresponding column index
    if max  $S > S_{old}$  then
       $X_k(:, l) = x(t)$ ; {store  $r_j$ ,  $\dot{x}(t)$ , initiate smoother for estimating  $\dot{x}(t)$  if measurement not available}
    end if
  end if
end if

```

Let $\xi = [e^T, \text{vec}(\tilde{K}), \tilde{K}_r^T]^T$ then we can bound the Lyapunov candidate above and below with positive definite functions as follows

$$\begin{aligned} \frac{1}{2} \min(\lambda_{\min}(P), \lambda_{\min}(\Gamma_x^{-1}), \Gamma_r^{-1}) \|\xi\|^2 &\leq V(e, \tilde{K}, \tilde{K}_r) \\ &\leq \frac{1}{2} \max(\lambda_{\max}(P), \lambda_{\max}(\Gamma_x^{-1}), \Gamma_r^{-1}) \|\xi\|^2. \end{aligned} \quad (21)$$

Let $[t_1, \dots, t_k, \dots]$ be a sequence where each t_k denotes a time when the history stack was updated. Due to the discrete nature of Algorithm 1 it follows that $t_{k+1} > t_k$. Taking the time derivative of the Lyapunov candidate along the trajectories of system (8), (41), and (19) over each interval $[t_k, t_{k+1}]$, and using the Lyapunov equation (9) we have

$$\begin{aligned} \dot{V}(e(t), \tilde{K}(t), \tilde{K}_r(t)) &= -\frac{1}{2} e^T(t) Q e(t) + e^T(t) P B (\tilde{K}^T(t) x(t) + \tilde{K}_r^T(t) r(t)) \\ &\quad - \text{tr}(\tilde{K}^T(t) (x(t) e^T(t) P B + \sum_{j=1}^p x_j x_j^T \tilde{K}(t))) \\ &\quad - \text{tr}(\tilde{K}_r^T(t) (r(t) e^T(t) P B + \sum_{j=1}^p r_j r_j^T \tilde{K}_r(t))). \end{aligned} \quad (22)$$

Simplifying further and canceling like elements the time derivative of the Lyapunov candidate reduces to

$$\begin{aligned} \dot{V}(e(t), \tilde{K}(t), \tilde{K}_r(t)) &= -\frac{1}{2} e^T(t) Q e(t) - \text{tr}(\tilde{K}^T(t) \sum_{j=1}^p x_j x_j^T \tilde{K}(t)) \\ &\quad - \text{tr}(\tilde{K}_r^T(t) \sum_{j=1}^p r_j r_j^T \tilde{K}_r(t)). \end{aligned} \quad (23)$$

Let $\Omega_K = \sum_{j=1}^p x_j x_j^T$, then it follows that Ω_K is non-negative definite. Hence over every interval $[t_k, t_{k+1}]$ $\dot{V}(e(t), \tilde{K}(t), \tilde{K}_r(t)) \leq 0$, hence $e(t)$, $\tilde{K}(t)$, $\tilde{K}_r(t)$ are bounded. Now consider that over a finite interval $r(t)$ is exciting. Then it follows that over a finite interval $[0, T]$ $x(t)$ is exciting (see [11] for proof of this fact). Then Algorithm 1 guarantees that the history stack X_T contains at least n linearly independent elements for all $t > T$. Hence we have $\Omega_K > 0$ for all $k > T$ (that is Ω_K is positive definite). Similarly since $r(t)$ is exciting, there exists at least one non-zero r_j , hence we have $\sum_{j=1}^p r_j r_j > 0$. Therefore

$$\begin{aligned} \dot{V}(e, \tilde{K}, \tilde{K}_r) &\leq -\frac{1}{2} \lambda_{\min}(Q) \|e\|^2 - \lambda_{\min}(\Omega_K) \|\tilde{K}\|^2 \\ &\quad - \sum_{j=1}^p r_j^2 \|\tilde{K}_r\|^2. \end{aligned} \quad (24)$$

This can be reduced to

$$\begin{aligned} \dot{V}(e, \tilde{K}, \tilde{K}_r) &\leq \\ &\quad \min \left(\lambda_{\min}(Q), 2\lambda_{\min}(\Omega_K), 2 \sum_{j=1}^p r_j^2 \right) \\ &\quad - \frac{V(e, \tilde{K}, \tilde{K}_r)}{\max(\lambda_{\max}(P), \lambda_{\max}(\Gamma_x^{-1}), \Gamma_r^{-1})} V(e, \tilde{K}, \tilde{K}_r). \end{aligned} \quad (25)$$

Note that Algorithm 1 guarantees that $\lambda_{\min}(\Omega_K)$ is monotonically increasing. Hence, (25) guarantees that the equation (20) is a common Lyapunov function, and establishes uniform exponential stability of the zero solution $e \equiv 0$ and $\tilde{K}(t) \equiv 0$, $\tilde{K}_r(t) \equiv 0$ [21]. Since $V(e, \tilde{K}, \tilde{K}_r)$ is radially unbounded, the result is global and x tracks x_{ref} exponentially and $K(t) \rightarrow K^*$, and $K_r(t) \rightarrow K_r^*$ exponentially fast as $t \rightarrow \infty$. \square

Remark 1

For the concurrent learning adaptive controller of Theorem 1, a sufficient condition for guaranteeing exponential convergence of tracking error $e(t)$ and weight estimation error $\tilde{K}(t)$ and $\tilde{K}_r(t)$ to zero is that the matrix $X = [x_1, x_2, \dots, x_p]$ contain n linearly independent data points, and for at least one of the recorded r_j , $r_j \neq 0$. This condition, referred to as the rank condition, requires only that the states be exciting (see Definition 1) over the finite interval when the data was recorded. The condition however, is not equivalent to a condition on PE states (see Definition 2) which requires the states to be exciting over *all* finite intervals. To further elaborate, note that the rank condition is only concerned with the behavior of the data in the past, whereas, PE is concerned with how the states behave over all past and future intervals. Finally, note that while the rank of a matrix can be calculated online, Definition 2 does not yield easily to online verification of whether a signal is PE.

Remark 2

The minimum singular value of the history stack $\sigma_{\min}(X_k)$ is directly proportional to the rate of convergence, and can be used as an online performance metric that relates the richness of the recorded data to guaranteed exponential convergence. Furthermore, it can be shown that $\sigma_{\min}(X_k) = \sqrt{\lambda_{\min}(\Omega_K)}$ [22].

Remark 3

If the rank condition is met, $\tilde{K}(t)$ and $\tilde{K}_r(t)$ converge to zero, and hence no additional modifications to the adaptive law, such as σ -modification or e -modification, or a projection operator that restricts the weights in a compact set are required to guarantee the boundedness of weights for this class of adaptive controllers. While we did not assume a pre-recorded history stack, note that the history stack can be pre-populated with selected system states from a set of recorded data from either open-loop or closed-loop experiments, or even from simulation testing. This allows for a nice way of incorporating test data into the control design process. If however, a pre-recorded data is not

available, the previously mentioned modifications can be used to guarantee the boundedness of weights until Algorithm 1 records data online such that $\text{rank}(X) = n$. The stability proof for this case follows by using arguments similar to that of Theorem 1 and is omitted due to space limitations.

Remark 4

In the adaptive control law of Theorem 1 we assumed that B was known. Alternatively, the adaptive law can be reformulated to only require the knowledge of $\text{sign}(B)$ (see [5]) or $\text{sign}(K^*)$ (see [3]).

Theorem 1 guarantees that the tracking error and weight error dynamics are globally exponentially stable subject to a rank condition. This property of concurrent learning adaptive controllers allows us to further characterize their response through transient performance bounds. The quantities $\|x - x_{rm}\|_{L_\infty}$ and $\|\tilde{K}\|_{L_\infty}$ have previously been used to characterize transient performance bounds of adaptive controllers [7]. We will follow a similar approach. For the ease of exposition, we assume without loss of generality that $e(0) = 0$, $K(0) = 0$, $K_r(0) = 0$, $\Gamma_x = \gamma_x I$, and $\Gamma_r = \gamma_r I$, where γ_x and γ_r are positive scalars, while I is the identity matrix of appropriate dimensions.

Corollary 2

If Theorem 1 holds, then the quantities $\|e(t)\|_{L_\infty}$, $\|\tilde{K}(t)\|_{L_\infty}$, and $\|\tilde{K}_r(t)\|_{L_\infty}$ are bounded from above by exponentially decreasing functions.

Proof

Assume for sake of exposition, $K(0) = 0$ and $K_r(0) = 0$, it follows from (20) that

$$\begin{aligned} V(e(0), \tilde{K}(0), \tilde{K}_r(0)) &= \frac{1}{2}e^T(0)Pe(0) + \frac{1}{2}\text{tr}(K^{*T}\Gamma_x^{-1}K^*) \\ &\quad + \text{tr}\left(\frac{1}{2}K_r^{*T}\Gamma_r^{-1}K_r^*\right) \\ &\leq \frac{1}{2}\lambda_{\max}(P)\|e(0)\|_2^2 + \frac{1}{2}\lambda_{\max}(\Gamma_x^{-1})\|K^*\|_2^2 \\ &\quad + \frac{1}{2}\lambda_{\max}(\Gamma_r^{-1})\|K_r^*\|_2^2 \end{aligned} \quad (26)$$

Let $\bar{\epsilon} = \frac{\min\left(\lambda_{\min}(Q), 2\lambda_{\min}(\Omega_K), 2\sum_{j=1}^p r_j^2\right)}{\max(\lambda_{\max}(P), \lambda_{\max}(\Gamma_x^{-1}), \Gamma_r^{-1})}$, then from (25) we have $\dot{V}(e, \tilde{K}, \tilde{K}_r) \leq -\bar{\epsilon}V(e, \tilde{K}, \tilde{K}_r)$. Therefore from Theorem 1 it follows that

$$V(e(t), \tilde{K}(t), \tilde{K}_r(t)) \leq V(e(0), \tilde{K}(0), \tilde{K}_r(0))e^{-\bar{\epsilon}t}. \quad (27)$$

Let $\Psi(t) = V(e(0), \tilde{K}(0), \tilde{K}_r(0))e^{-\bar{\epsilon}t}$, since $\frac{1}{2}\lambda_{\min}(P)\|e\|_\infty^2 \leq \frac{1}{2}\lambda_{\min}(P)\|e\|_2^2 \leq \frac{1}{2}e^TPe \leq \Psi$ it follows that

$$\begin{aligned} \|e\|_{L_\infty} &\leq \left(\frac{\lambda_{\max}(P)}{\lambda_{\min}(P)}\|e(0)\|_2^2 + \frac{\lambda_{\max}(\Gamma_x^{-1})}{\lambda_{\min}(P)}\|K^*\|_2^2 + \right. \\ &\quad \left. \frac{\lambda_{\max}(\Gamma_r^{-1})}{\lambda_{\min}(P)}\|K_r^*\|_2^2\right)^{\frac{1}{2}} e^{-\frac{\bar{\epsilon}t}{2}} \end{aligned} \quad (28)$$

If $e(0) = 0$, $\Gamma_x = \gamma_x I$, and $\Gamma_r = \gamma_r I$, the above equation reduces to

$$\|e\|_{L_\infty} \leq \sqrt{\frac{\|K^*\|_2^2}{\gamma_x \lambda_{\min}(P)} + \frac{\|K_r^*\|_2^2}{\gamma_r \lambda_{\min}(P)}} e^{-\frac{\bar{\epsilon}t}{2}} \quad (29)$$

Similarly, $\frac{1}{2}\lambda_{\min}(\Gamma_x^{-1})\|\tilde{K}\|_\infty^2 \leq \Psi$ and $\frac{1}{2}\lambda_{\min}(\Gamma_r^{-1})\|\tilde{K}_r\|_\infty^2 \leq \Psi$. Therefore, it follows that

$$\|\tilde{K}\|_{L_\infty} \leq \sqrt{\|K^*\|_2^2 + \frac{\gamma_x}{\gamma_r}\|K_r^*\|_2^2} e^{-\frac{\bar{\epsilon}t}{2}}, \quad (30)$$

and

$$\|\tilde{K}\|_{L_\infty} \leq \sqrt{\frac{\gamma_r}{\gamma_x} \|K^*\|_2^2 + \|K_r^*\|_2^2} e^{-\frac{\epsilon t}{2}}. \quad (31)$$

□

Remark 5

Equation (29) brings an interesting property of the presented controller to light. It shows that the tracking error is bounded above by an exponentially decaying function, hence, even a high adaptation gain (γ_x and γ_r) is not required for guaranteeing good tracking performance when concurrent learning is used. It is interesting to note here that recently proposed high performance adaptive controllers that rely only on instantaneous data need high adaptation gains to guarantee good tracking (see [7], [23]). The results here suggest that memory can be used in a similar manner as described here to relieve the high gain requirement in adaptive control for the class of systems discussed here and to guarantee simultaneous exponential weight and tracking error convergence.

The analysis can be further extended to guarantee the boundedness of the control signal, particularly, let \bar{x}_{rm} be a known upper bound for the bounded input bounded output reference model, then it follows that

$$\|x\|_{L_\infty} \leq \|e\|_{L_\infty} + \bar{x}_{rm}, \quad (32)$$

similarly,

$$\|K\|_{L_\infty} \leq \|\tilde{K}\|_{L_\infty} + \|K^*\|_\infty, \quad (33)$$

$$\|K_r\|_{L_\infty} \leq \|\tilde{K}_r\|_{L_\infty} + \|K_r^*\|_\infty. \quad (34)$$

Since $u = K^T x + K_r^T r$, we have

$$\|u\|_{L_\infty} \leq \|K\|_{L_\infty} \|x\|_{L_\infty} + \|K_r\|_\infty \|r\|_{L_\infty}. \quad (35)$$

Due to (29), (30), and (31), as $t \rightarrow \infty$ we have,

$$\|u\|_{L_\infty} \leq \|K^*\|_{L_\infty} \bar{x}_{rm} + \|K_r^*\|_\infty \|r\|_{L_\infty}. \quad (36)$$

Letting $\bar{x}_{rm} = \alpha \|r\|_{L_\infty}$, where α is a positive constant, we obtain an upper bound on the control input.

$$\|u\|_{L_\infty} \leq \alpha (\|K^*\|_{L_\infty} + \|K_r^*\|_\infty) \|r\|_{L_\infty}. \quad (37)$$

This result is of particular significance, since, by estimating conservative bounds on K^* and K_r^* , it allows the design of a reference model to ensure that the control signal does not exceed a predefined value. This fact can be useful, since physical actuators saturate.

4. A CONNECTION TO σ MODIFICATION

A key shortcoming of the traditional MRAC adaptive law of equation (10) is that it does not guarantee boundedness of the adaptive weights. It is well known that if the weights are not guaranteed to be bounded, they can drift, and for sufficiently large Γ_x and Γ_r , the bursting phenomena may be encountered [9][4]. Bursting is a result of unbounded weight error growth and is characterized by high frequency oscillations. Various modifications have been proposed to counter weight drift, these include the classical σ modification of Ioannou [24], e modification of Narendra and Annaswamy [25], and projection based algorithms (see e.g. [3]). These modification guarantee that the weights stay bounded in a compact neighborhood of a preselected value. In this section we make a connection between the concurrent learning adaptive law presented before and the classical σ modification, which can be viewed as a way to add damping to the weight error dynamics. Consider the traditional adaptive law with σ modification for K (the case for K_r follows similarly):

$$\dot{K}(t) = -\Gamma_x(x(t)e^T(t)PB + \sigma(K(t) - \bar{K})), \quad (38)$$

where $\sigma > 0$ denotes the σ modification gain and \bar{K} is an *a-priori* selected desired value around which the weights are required to be bounded. Often, \bar{K} is set to zero. Let $\Delta K = \bar{K} - K^*$, then the weight error dynamics for the σ modification adaptive law in (38) can be written as

$$\dot{\tilde{K}}(t) = -\Gamma_x \sigma \tilde{K}(t) - \Gamma_x (x(t)e^T(t)PB - \sigma \Delta K). \quad (39)$$

Since Γ_x and σ are assumed to be positive definite, and since the equation contains a term linear in \tilde{K} , one can intuitively argue that the weight error dynamics will be bounded-input-bounded-state stable if the tracking error e stays bounded. In fact, through Lyapunov arguments it can be shown that the update law in (38) will guarantee that the tracking error and the weight error stay bounded. This result is classical, and we do not intend to get into its details here (they can be found for example in [18]). The point that we intend to make here follows from noting that if $\bar{K} = K^*$, that is, if the ideal weights are known *a-priori*, then σ modification based adaptive law of (38) will also guarantee exponential parameter error and tracking error convergence. Now if we compare the weight error dynamics in (39) with those obtained in (41) with concurrent learning it can be seen that once X_k is full ranked, the weight error dynamics with concurrent learning also contain a linear term in \tilde{K} which is multiplied by a Hurwitz matrix. Furthermore, the weight error dynamics with concurrent learning do not contain the ΔK term. This suggests that the concurrent learning update law in (16) is comparable in its behavior to a σ modification based adaptive law when one knows *a-priori* the ideal weights K^* , and hence contains inherently the desirable robustness properties of σ modification. This analysis goes to show how exponential weight convergence results in robustness of MRAC schemes and is in agreement with the finding of the authors in [11], [19], and [6].

The analysis presented here also suggests another adaptive law that concurrently uses recorded data with current data. Consider the single input case ($m = 1$), let $Y = [\epsilon_{K_1}, \dots, \epsilon_{K_p}]$, and assume that the history stack X_k is full ranked, then using least-squares regression based arguments we see that $K^* = K(t) - (X_k X_k^T)^{-1} X_k Y(t)$. This allows us to estimate K^* and use it in the σ modification adaptive law in place of \bar{K} . It can be shown that this results in an exponentially stable adaptive controller (see for example [26]). However, this adaptive law requires an inversion, does not scale well to multiple input case, and its robustness may suffer due to numerical errors in inverting ill-conditioned matrices. On the other hand the concurrent learning law presented in the previous sections guarantees that the weights approach smoothly and exponentially fast to their ideal values and provides us with a metric ($\sigma_{\min}(X_k)$) to quantify the tracking error and weight error convergence rates.

5. EFFECT OF ERRORS IN ESTIMATION OF \dot{x}_j

In the analysis in Section 3 it was assumed that \dot{x}_j was available for a recorded data point. If a direct measurement for \dot{x} is not available, a fixed point smoother can be used to obtain an estimate $\hat{\dot{x}}$. In this section we present a brief analysis of the effect of errors in estimating \dot{x}_j . The analysis indicates that in the presence of estimation errors in \dot{x}_j , the tracking error and weight error is ultimately bounded, and the size of the bound depends on the magnitude of the estimation error. The analysis is performed for effects on $K(t)$, the analysis for $K_r(t)$ follows similarly. Let $\hat{\dot{x}} = \dot{x}_j + \delta_{\dot{x}_j}$, where $\delta_{\dot{x}_j}$ is the estimation error for the j^{th} recorded data point. It follows from 14

$$\hat{\epsilon}_{K_j}(t) = (B^T B)^{-1} B^T (\dot{x}_j - A_{rm} x_j - B_{rm} r_j - B \epsilon_{K_r j}(t) + \delta_{\dot{x}_j}), \quad (40)$$

where $\hat{\epsilon}_{K_j}$ denotes the estimate of ϵ_{K_j} . It follows that $\hat{\epsilon}_{K_j} = \epsilon_{K_j} + \hat{\delta}_{\dot{x}_j}$, where $\hat{\delta}_{\dot{x}_j} = (B^T B)^{-1} B^T \delta_{\dot{x}_j}$. The weight error dynamics now become:

$$\dot{\tilde{K}}(t) = -\Gamma_x \left(x(t)e^T(t)PB + \sum_{j=1}^p x_j x_j^T \tilde{K}(t) + x_j \hat{\delta}_{\dot{x}_j} \right). \quad (41)$$

Consequently, the Lie derivative of the Lyapunov function changes from (25) to

$$\begin{aligned} \dot{V}(e(t), \tilde{K}(t), \tilde{K}_r(t)) = & -\frac{1}{2}e^T(t)Qe(t) - \text{tr}(\tilde{K}^T(t) \sum_{j=1}^p x_j x_j^T \tilde{K}(t)) \\ & - \text{tr}(\tilde{K}_r^T(t) \sum_{j=1}^p r_j r_j \tilde{K}_r(t)) + \|\tilde{K}(t)\| \bar{c}, \end{aligned} \quad (42)$$

where $\bar{c} = \|\sum_{j=1}^p x_j \hat{\delta}_{\dot{x}_j}\|$. It follows therefore that the tracking error is uniformly ultimately bounded (given that only bounded data points are recorded), furthermore, the adaptive weights $K(t)$ approach and stay bounded in a compact ball around the ideal weights K^* . This is a stronger result than that obtained with σ -modification[24], because the weights are guaranteed to be bounded around their ideal values, and the size of the ball is dependent on the estimation error $\delta_{\dot{x}_j}$. It can be shown that the tracking error and weight errors approach the compact ball around the origin exponentially fast, furthermore, exponential stability results are recovered if $\delta_{\dot{x}_j} = 0$. This intuitive result is confirmed by the experimental studies in Section 7.

6. ILLUSTRATIVE EXAMPLE

In this section we present an illustrative simulation example of control of a simple linear system with concurrent learning adaptive law. The focus is on understanding the concurrent learning approach, hence we use an exemplary second order system:

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 4 & -2 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u. \quad (43)$$

It is assumed that the state matrix (matrix A in (1)) is unknown for the purpose of control design. Note that the system is unstable as the eigenvalues of the A matrix are 1.9025, and -2.1025 . The control objective is to make the system behave like the following second order reference model with natural frequency of 3.9 rad/sec and damping ratio of 0.7,

$$\dot{x}_{rm} = \begin{bmatrix} 0 & 1 \\ -15.21 & -5.46 \end{bmatrix} x_{rm}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(t). \quad (44)$$

The simulation runs for 9 seconds with a time-step of 0.001 seconds. The reference signal $r(t)$ is: $r(t) = 10$ between 0 to 2 seconds, $r(t) = -10$ between 2 to 3 seconds, and $r(t) = 10$ thereafter. Control law of (3) is used along with the adaptation law of Theorem 1, with $\Gamma_x = 10$ and $\Gamma_r = 1$, these are held constant through all presented simulations. The initial values of K and K_r were set to zero.

6.1. Performance with an online recorded history stack

The results presented in this section do not assume that a pre-recorded history stack is available, and Algorithm 1 is used to populate and manage the history stack when full. Figure 1 compares the tracking performance of the adaptive controllers with and without concurrent learning. It can be seen that system states are almost indistinguishable from the reference model when using concurrent learning adaptive controller after about 2 seconds into the simulation, particularly when tracking the reference command between 2 and 3 seconds. Whereas, for the classical adaptive laws which uses only instantaneous data, the tracking is not as precise. The reason for this becomes clear when we examine the evolution of weights in Figure 2. With concurrent learning, the weights approach their ideal values, and have converged shortly after 2 second, particularly note that K_r converges within a single time step. Whereas for the classical adaptive laws, the weights do not converge to their ideal values. Figure 3 compares the tracking error with and without concurrent learning, we observe that

the tracking error approaches and remains at zero with concurrent learning within 3 seconds of the simulation, in contrast with the tracking error for the classical adaptive law, which only approaches zero in steady-state. Particularly, in contrast with the classical adaptive law, for the step at 2 seconds, no visible transient can be seen with concurrent learning, this is in agreement with Corollary 2, which guarantees exponentially decaying bounds on the tracking error. For the presented results, $\bar{\epsilon} = 0.0801$. Figure 4 compares the control effort required by the two adaptive controllers. It is observed that while the peak magnitudes of the required control effort remain comparable, the control effort with concurrent learning does not oscillate as much as that without concurrent learning. This is a result of the weights arriving their ideal values, and indicates that the control effort with concurrent learning requires less energy and is in that sense optimal. Note that increasing the learning rates Γ_x , and Γ_r only makes the control effort without concurrent learning more oscillatory and could lead to bursting [9]. Furthermore, note that the reference signal is not persistently exciting, it is however, exciting over a finite interval (0 to 5 seconds). Hence, the simulation results show that concurrent learning adaptive controllers guarantee exponential stability of the zero solution of tracking error and weight error dynamics without requiring persistency of excitation. This is achieved through concurrent adaptation on specifically selected and online recorded data stored in the history stack using Algorithm 1. Figure 11 shows the singular value of the history stack matrix, it is seen that Algorithm 1 ensures that $\sigma_{\min}(X_k)$ increases monotonically. Finally, Figure 6 compares the evolution of the Lyapunov function with and without concurrent learning. With concurrent learning, the Lyapunov function converges exponentially to zero, supporting the claim of Theorem 1, whereas, without concurrent learning, as expected, the Lyapunov function remains bounded. In that figure, the asterisk symbols depict where a new data point was recorded.

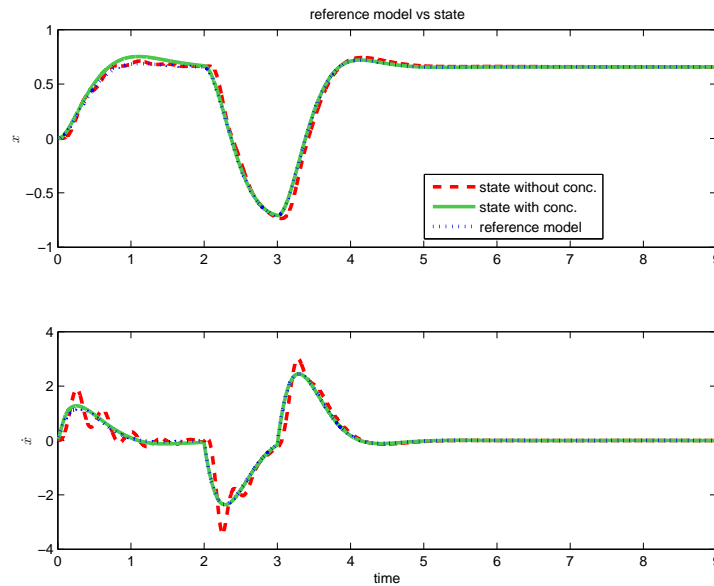


Figure 1. Comparison of tracking performance of adaptive controllers with and without concurrent learning. Note that concurrent learning results in significantly improved reference model tracking, in fact, the system states and the reference model states are almost indistinguishable with concurrent learning after about 2 seconds, whereas without concurrent learning this is not the case.

6.2. Performance with a pre-recorded and online recorded history stack

A pre-recorded history-stack of 10 data points was recorded during a previous simulation run with only the baseline adaptive law of (10) and with $r(t) = 5$ for the entire duration of that simulation run. The history-stack is then updated online using Algorithm 1. For the history-stack used, $\sigma_{\min}(\Omega_0) \approx 1.1$, at $t = 0$.

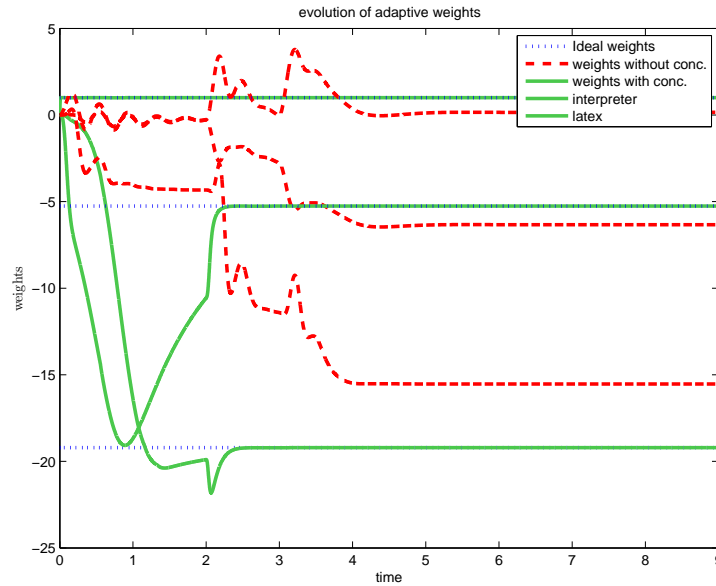


Figure 2. Comparison of evolution of adaptive weights with and without concurrent learning. Note that the weights converge to their true values within 3 seconds when using concurrent learning, whereas, for the classical adaptive law the weights do not converge to their ideal values.

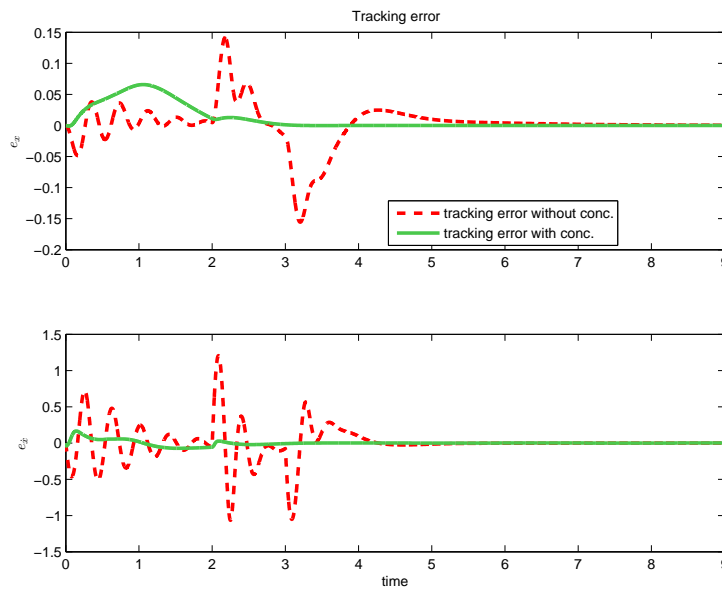


Figure 3. Comparison of tracking error with and without concurrent learning. Note that transient performance is significantly improved with concurrent learning after the step at 2 seconds, this corresponds to the time when the weights have converged. In contrast, there is a visible transient when concurrent learning is not used.

Figure 7 compares the tracking performance of the adaptive controllers with and without concurrent learning. It can be seen that system states are almost indistinguishable from the reference model when using concurrent learning adaptive controller, particularly when tracking the reference command between 2 and 3 seconds. Whereas, for the classical adaptive law which uses only instantaneous data, the tracking is not as precise. The reason for this becomes clear

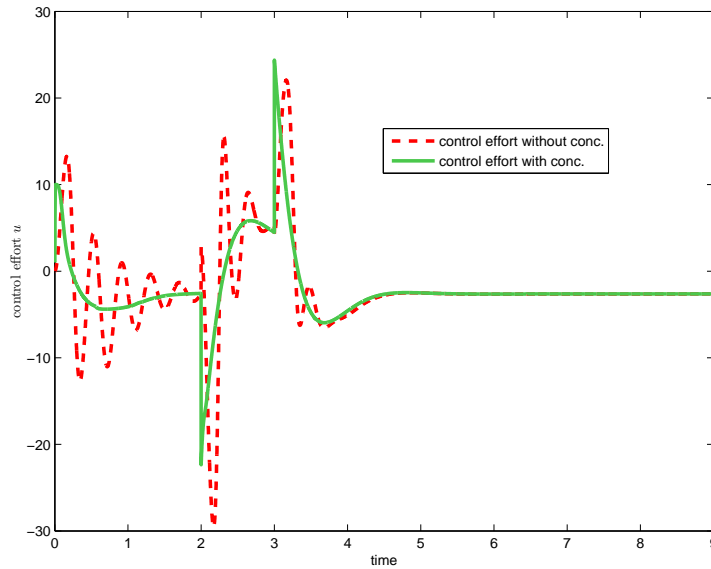


Figure 4. Comparison of control effort with and without concurrent learning. Note that while the peak magnitude of the control effort remains comparable, the control effort with concurrent learning does not oscillate as much as the control effort without concurrent learning. This indicates that the control effort with concurrent learning requires less energy, an effect arising due to the convergence of adaptive weights to their ideal values.

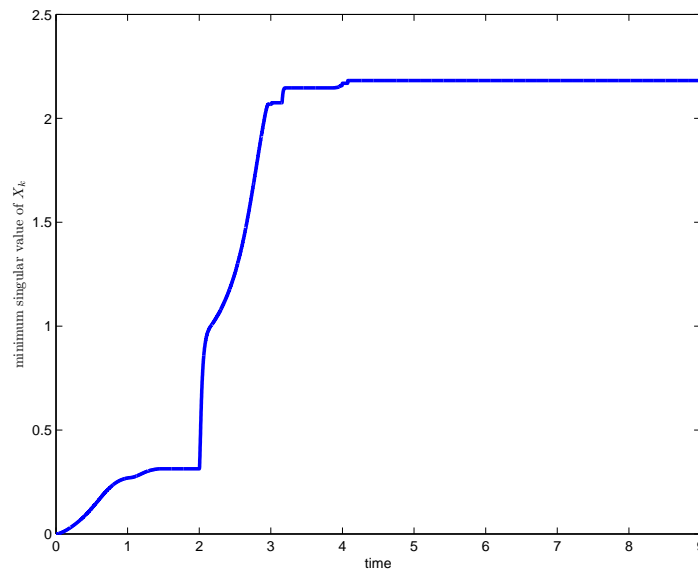


Figure 5. Evolution of the minimum singular value of the history stack matrix X_k , note that Algorithm 1 guarantees that $\sigma_{\min}(X_k)$ increases monotonically

when we examine the evolution of weights in Figure 8. With concurrent learning, the weights rapidly approach their ideal values, and have converged within 1 second, particularly note that K_r converges within a single time step. Whereas for the classical adaptive law, the weights do not converge to their ideal values. Figure 9 compares the tracking error with and without concurrent learning, we observe that the tracking error rapidly approaches and remains at zero with concurrent

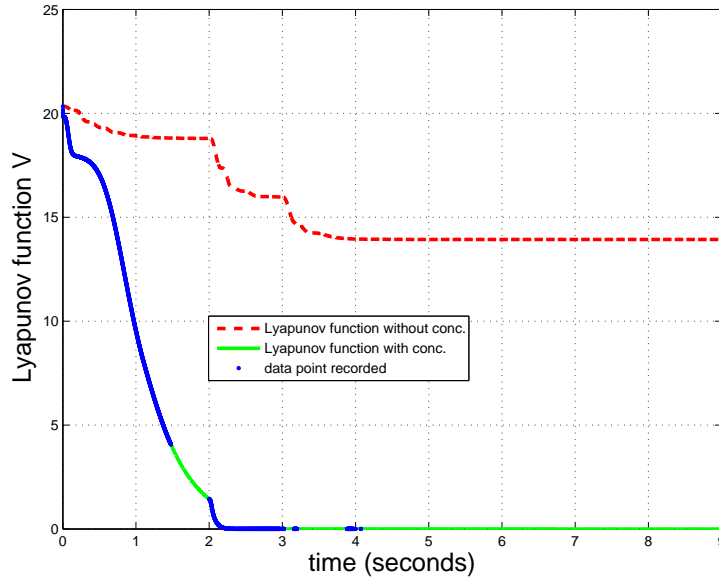


Figure 6. Comparison of the Lyapunov function with and without concurrent learning. Note that the Lyapunov function converges to zero and can be bounded above by an exponentially convergent function when concurrent learning is used, supporting the developed theory. Without concurrent learning, the Lyapunov function does not converge to zero, rather, only remains bounded. The asterisk symbols mark the time when the history stack was updated.

learning, in contrast with the tracking error for the classical adaptive law. Particularly, in contrast with the classical adaptive law, for the step at 2 seconds, no visible transient can be seen with concurrent learning, this is in agreement with Corollary 2, which guarantees exponentially decaying bounds on the tracking error. For the presented results, $\bar{\epsilon} = 0.0455$ and $\sqrt{\gamma} = 20.166$. Hence, $\|\xi\|_{L_\infty} \leq 20.166$, and $\|\xi(t)\|^2 \leq 20.166^2 e^{-0.046t}$ which turns out to be a conservative bound. Figure 10 compares the control effort required by the two adaptive controllers. We observe that while the peak magnitudes of the required control effort remain comparable, the control effort with concurrent learning does not oscillate as much as that without concurrent learning. This is a result of the weights arriving at their ideal values, and indicates that the control effort with concurrent learning requires less energy and is in that sense optimal. Note that increasing the learning rates Γ_x , and Γ_r only make the control effort without concurrent learning more oscillatory and could lead to bursting [9]. Furthermore, note that the reference signal is not persistently exciting. Hence, the simulation results show that if the rank condition is satisfied, concurrent learning adaptive controllers guarantee exponential stability of the zero solution of tracking error and weight error dynamics without requiring persistency of excitation. Figure 11 shows that Algorithm 1 ensures that $\sigma_{\min}(X_k)$ increases monotonically. Finally, Figure 12 compares the evolution of the Lyapunov function with and without concurrent learning. With concurrent learning, the Lyapunov function converges exponentially to zero, supporting the claim of Theorem 1, whereas, without concurrent learning, as expected, the Lyapunov function remains bounded. In that figure, the asterisk symbols depict where a new data point was recorded.

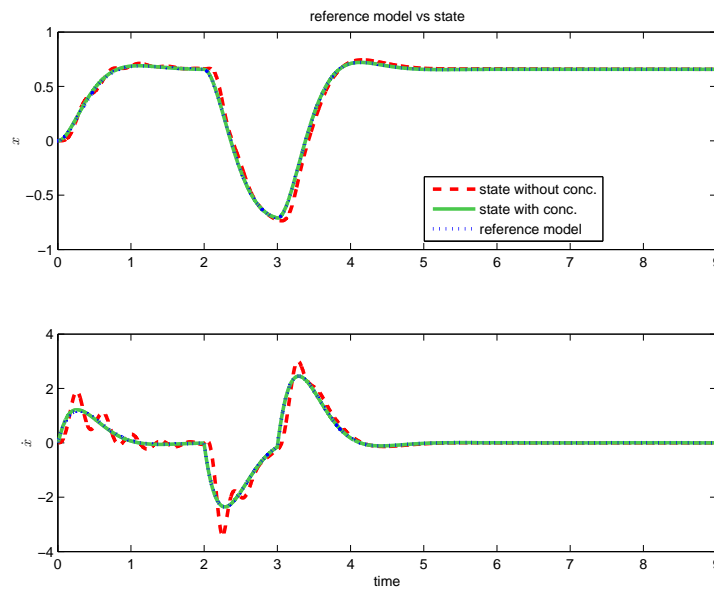


Figure 7. Comparison of tracking performance of adaptive controllers with and without concurrent learning. Note that the system states and the reference model states are almost indistinguishable with concurrent learning from $t = 0$ due to the use of a pre-recorded history stack, whereas without concurrent learning this is not the case.

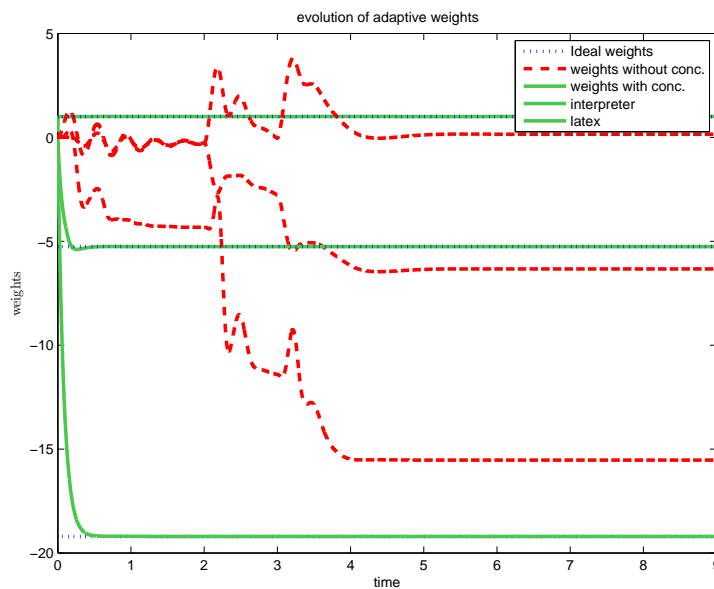


Figure 8. Comparison of evolution of adaptive weights with and without concurrent learning. Note that the weights converge rapidly (in less than 0.5 seconds) to their true values when using concurrent learning with a pre-recorded history stack, whereas, for the classical adaptive law the weights do not converge to their ideal values.

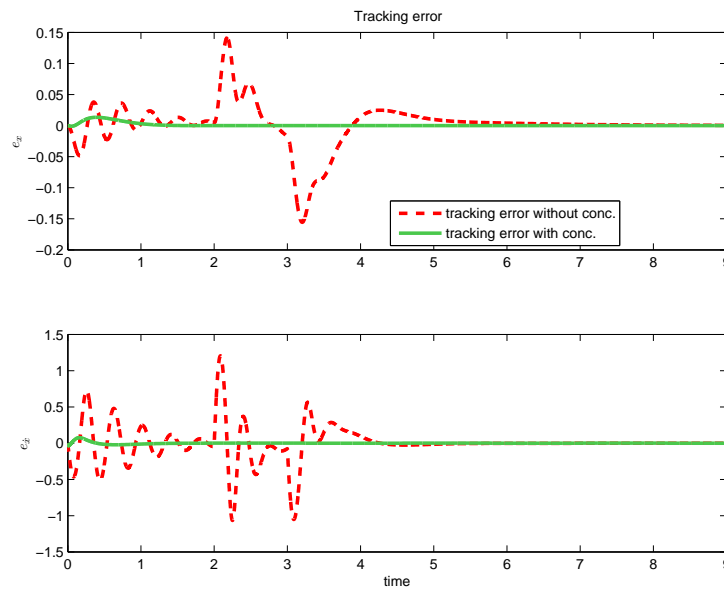


Figure 9. Comparison of tracking error with and without concurrent learning. Note that no visible transient is observed for the step at 2 seconds with concurrent learning, in contrast with the transient observed without concurrent learning.

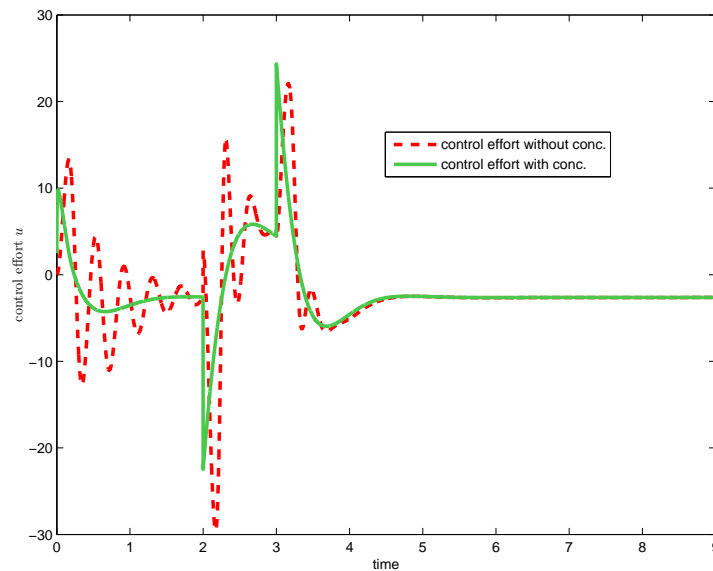


Figure 10. Comparison of control effort with and without concurrent learning. Note that while the peak magnitude of the control effort remains comparable, the control effort with concurrent learning does not oscillate as much as the control effort without concurrent learning. This indicates that the control effort with concurrent learning requires less energy, an effect arising due to the convergence of adaptive weights to their ideal values. It is interesting to note that the control effort with concurrent learning using a pre-recorded history stack is comparable when only online recorded data is used.

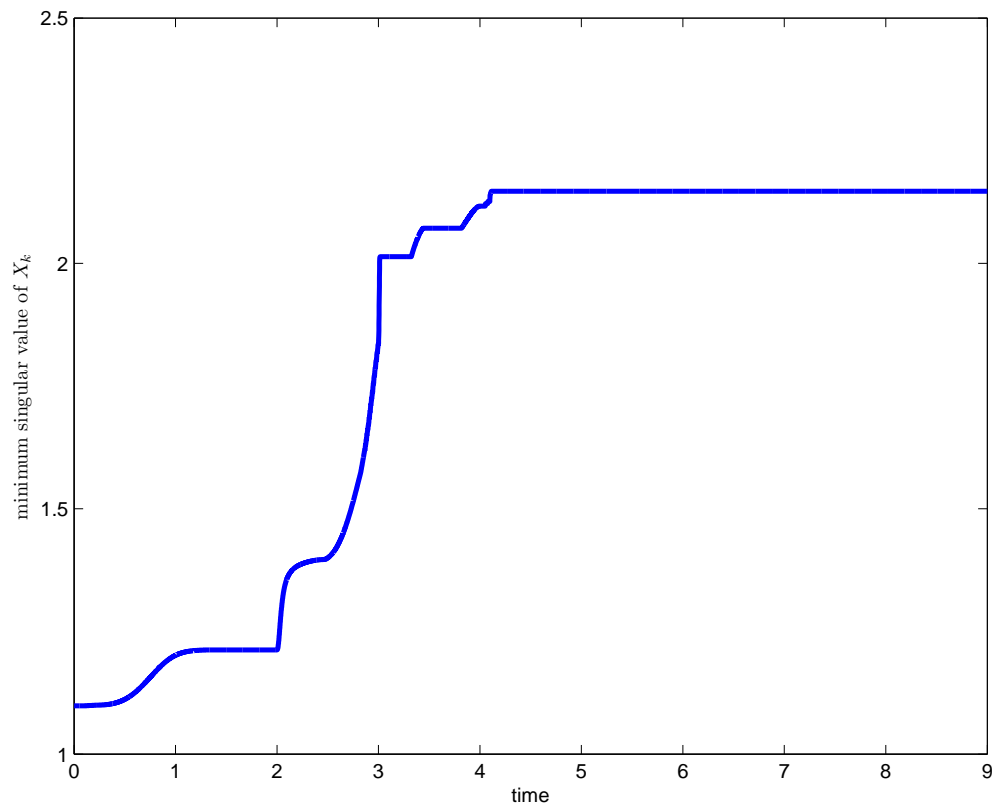


Figure 11. Evolution of the minimum singular value of the history stack matrix X_k , note that Algorithm 1 guarantees that $\sigma_{\min}(X_k)$ increases monotonically, note also the nonzero initial minimum singular value due to the use of a pre-recorded history stack.

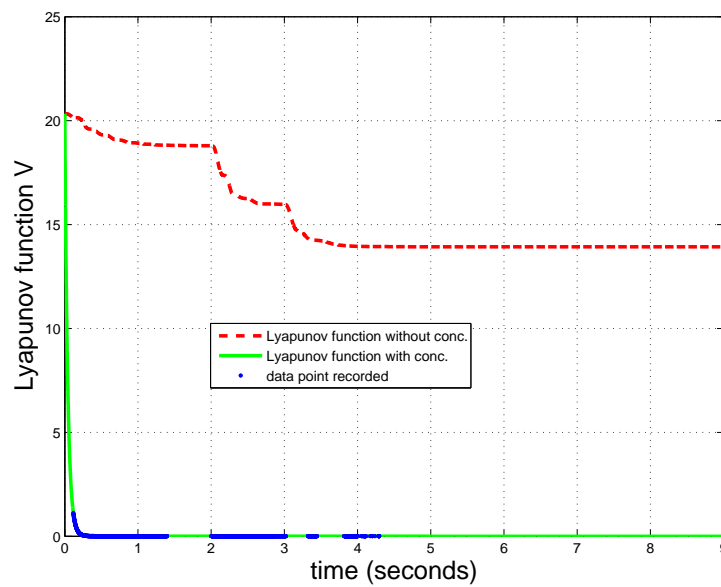


Figure 12. Comparison of the Lyapunov function with and without concurrent learning. Note that the Lyapunov function converges exponentially and rapidly to zero when concurrent learning is on, supporting the developed theory. Without concurrent learning, the Lyapunov function does not converge to zero, rather, only remains bounded. The asterisk symbols mark the time when the history stack was updated.

7. EXPERIMENTAL VALIDATION ON A 3 DEGREE OF FREEDOM HELICOPTER MODEL

This section presents experimental results of a concurrent learning adaptive controller on a real world system, namely the Quanser 3-DOF helicopter. A picture of the system is presented in Figure 13. The system consists of an arm mounted on a base with two rotors and counterweight at each end. The system can pitch and yaw about the mount of the arm by varying the voltage on the two rotors attached to the arm. The system can also roll about the axis of the arm. Yawing motion is achieved by applying a rolling moment to the arm through the application of different voltages to each of the rotors. The focus here is on the control of the pitch axis the helicopter. The relevant state vector is defined as $x = [\theta, \dot{\theta}]$, where θ is the pitch angle, and $\dot{\theta}$ is the pitch rate. The actuation u is achieved by varying the voltage to both the motors simultaneously to avoid roll moments. It is denoted by $v_1 = v_2 = u/2$, where v_1 and v_2 denote the voltages to the left and the right rotor respectively.

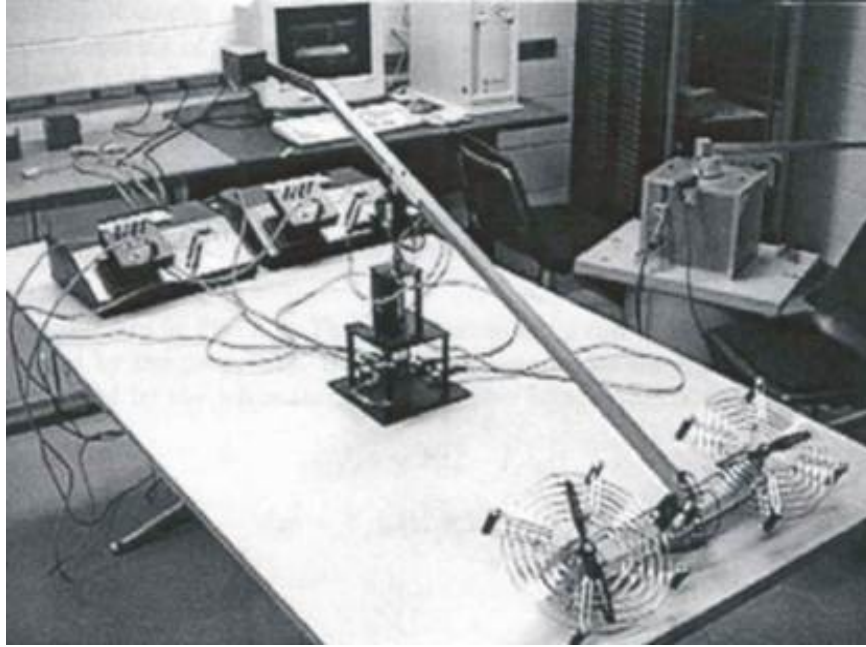


Figure 13. The Quanser 3-DOF helicopter. The system can roll about the axis of the arm as well as pitch and yaw about the mount of the arm by varying the voltages to the two rotors attached to the arm.

A linear model for the system can be assumed to have the following form:

$$\dot{x} = Ax + \begin{bmatrix} 0 \\ M_\delta \end{bmatrix} u \quad (45)$$

The state matrix A of the system is unknown, the control effectiveness derivative relating the pitch actuation to pitch rate was estimated to be $M_\delta = 0.25 \frac{rad}{sec^2 v}$. The pitch angle θ is available for measurement through an encoder, the pitch rate is estimated using a Kalman filter based fixed point smoother using a simplified process model for the system given below:

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x. \quad (46)$$

Further details on how such a smoother can be implemented can be found in [13].

The control objective is to make the system behave like a second order reference model of the form (2) with natural frequency of 2.8 rad/sec and damping ratio of 0.7. For these values, an estimate of the ideal gains was found to be $K^* = [-16 \ -8]$ and $K_r^* = 16$. Note that the estimates of K^*, K_r^* are not required for the implementation of the controller, they were estimated for the

purpose of comparison that follows. The experiment runs for 90 seconds with a time step of 0.0025 seconds. The reference signal is a cosine function with frequency 0.3 and amplitude 5 for 47 seconds, followed by two steps of magnitude of 5 at 54 and 74 seconds each lasting 10. Control law of (3) is used along with the adaption law of Theorem 1, with $\Gamma_x = 6$ and $\Gamma_r = 10$, which are held constant through the experiment. A pre-recorded history stack is not available. Hence, until the history stack is verified to meet the rank condition, a σ -modification term with $\lambda = 0.1$ is added to the adaptive update law in order to guarantee boundedness of the weights. The σ -modification term is removed after the history stack reaches full rank. The adaptive weights K were initialized at zero and K_r was initialized at 5.

Figure 14 shows the tracking performance of the adaptive controller with concurrent learning. It can be seen that the system performance improves significantly once the rank condition on the history stack is met at about 15 seconds. After which, the system tracks the magnitude of the cosine function and the steps accurately. However, the system lags slightly behind the reference model. This is possibly due to unknown prevalent hardware time delays in relaying the command to the system and unmodeled actuator dynamics. The reason for the improved performance becomes clear if the evolution of the weights in Figure 15 is examined. Once the rank condition is met at about 15 seconds the weights begin to converge to a compact set around their estimated ideal values rapidly. The feedforward gain K_r needs only 10 seconds to converge to a compact set while the feedback gain K takes about 30 seconds longer. This difference in convergence rate is comparable to that obtained in the simulation results presented in Section 6. The adaptive gains converge to a close neighborhood of the estimated ideal values. Particularly, the feedforward gain differs by about 5% from its estimated ideal value of $K_r^* = 16$, and the steady-state values of the feedback gains k_1 differ by about 13% and k_2 differs by about 5% from the estimated ideal values of $K^* = [-16 \ -8]$. This can be attributed to errors in estimating the ideal values and also to the presence of noise, disturbances, and unmodeled dynamics in the system. Figure 16 shows the tracking error as well as the control effort of the adaptive controller with concurrent learning. It can be seen that the tracking error reduces significantly after the weights converge at about 30 seconds. This indicates that the concurrent learning adaptive controller delivers better performance over classical MRAC with σ -modification. The transient response is bounded, however, the time delays as well as the unmodeled dynamics prevent the system from tracking the reference model exactly in transient phases. It is interesting to note that almost zero tracking error is observed during steady state conditions, which indicates that the adaptive controller can trim the system effectively in presence of disturbances and time delays. In combination, these results validate that the concurrent learning adaptive controller can deliver good tracking performance despite noisy estimates of the pitch angle from the encoder, unmodeled time delays, and presence of external disturbances. Figure 16 also shows the control effort required by the adaptive controller. As was observed in the simulation above, the control effort decreases when concurrent learning adaptive controller is used. Particularly, the control effort is seen to reduce once the weights start to converge (after about 15 seconds).

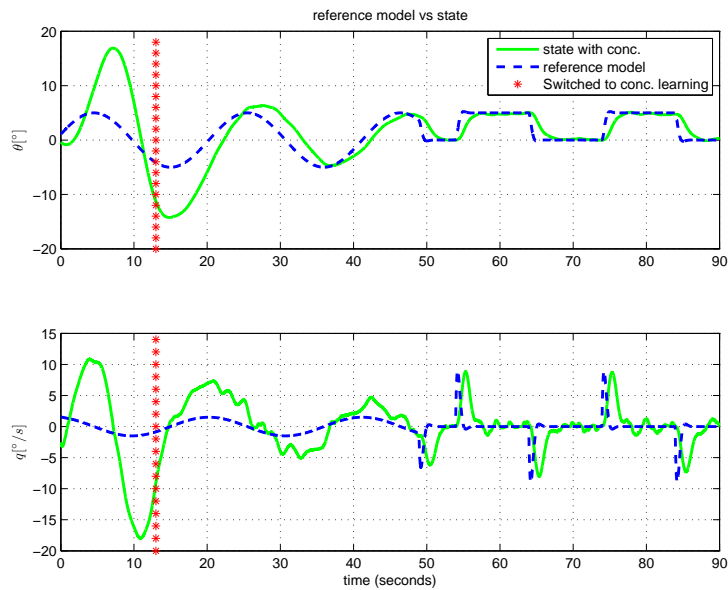


Figure 14. Tracking performance of the adaptive controller with concurrent learning. Note that the performance increases drastically after 30 seconds, this corresponds to the time when the weights have converged. Uncertainties and time delays prevent the system to track the reference model perfectly.

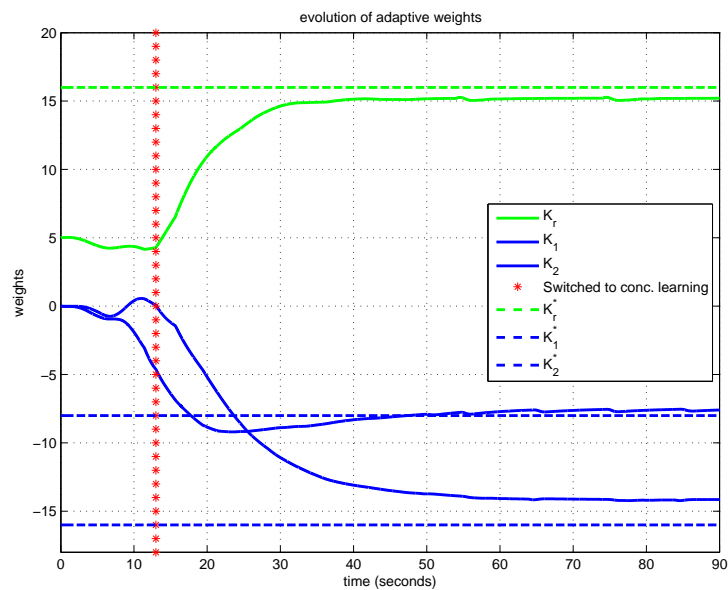


Figure 15. Evolution of the adaptive weights with concurrent learning. Note that after the history stack meets the rank condition at about 15 seconds the weights converge to a set around their estimated ideal values. Note that the estimate of the ideal weights is not required for the implementation of the controller.

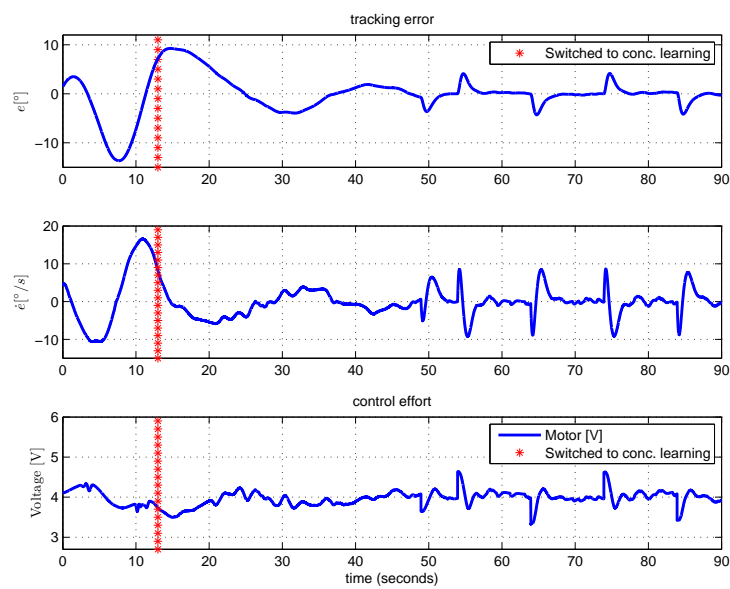


Figure 16. Tracking error and control effort with concurrent learning. Note that the tracking performance is dramatically improved once the history stack meets the rank condition. Errors during changes of the reference signal occur mainly due to time delays in the system. Also the control effort is clearly reduced once the rank condition is met.

8. CONCLUSION

A concurrent learning adaptive controller for uncertain linear multivariable dynamical systems was presented. The concurrent learning adaptive controller uses instantaneous data concurrently with recorded data for adaptation and can guarantee global exponential stability of the zero solution of the tracking error and weight error dynamics subject to a sufficient verifiable condition on linear independence of the recorded data. This indicates that the tracking error and weight errors exponentially approach zero and their infinity norms are exponentially bounded. This fact was used to formulate exponentially decaying upper bounds on tracking and weight error without relying on high adaptation gain. Furthermore, the exponential convergence also indicates that system dynamics will exponentially approach the reference model dynamics, thereby recovering the desired transient response and stability characteristics of the reference model. These results have significant implications in verification and validation of adaptive controllers and in designing adaptive controllers that avoid saturation. We demonstrated the effectiveness of the presented concurrent learning approach through an exemplary simulation study. Furthermore, the presented controller architecture was also validated through experimentation on a 3 degree of freedom experimental model.

ACKNOWLEDGEMENT

This work was supported in part by NSF ECS-0238993 and NASA Cooperative Agreement NNX08AD06A. The authors thank Maximilian Mühlegg for his valuable help in revising this paper.

REFERENCES

1. Mettler B. *Modeling Identification and Characteristics of Miniature Rotorcrafts*. Kluwer Academic Publishers: USA, 2003.
2. Jategaonkar RV. *Flight Vehicle System Identification A Time Domain Approach, Progress in Astronautics and Aeronautics*, vol. 216. American Institute of Aeronautics and Astronautics: Reston, 2006.
3. Tao G. *Adaptive Control Design and Analysis*. Wiley: New York, 2003.
4. Narendra KS, Annaswamy AM. *Stable Adaptive Systems*. Prentice-Hall: Englewood Cliffs, 1989.
5. Hovakimyan N, Cao C. *L_1 Adaptive Control Theory*. SIAM, 2010.
6. Sastry S, Bodson M. *Adaptive Control: Stability, Convergence, and Robustness*. Prentice-Hall: Upper Saddle River, 1989.
7. Cao C, Hovakimyan N. Design and analysis of a novel adaptive control architecture with guaranteed transient performance. *Automatic Control, IEEE Transactions on* march 2008; **53**(2):586–591, doi: 10.1109/TAC.2007.914282.
8. Narendra K, Annaswamy A. A new adaptive law for robust adaptation without persistent excitation. *IEEE Transactions on Automatic Control* February 1987; **32**(2):134–145.
9. Åström KJ, Wittenmark B. *Adaptive Control*. 2nd edn., Addison-Wesley: Readings, 1995.
10. Volyanskyy K, Haddad W, Calise A. A new neuroadaptive control architecture for non-linear uncertain dynamical systems: Beyond σ and e -modifications. *IEEE Transactions on Neural Networks* Nov 2009; **20**(11):1707–1723, doi:10.1109/TNN.2009.2030748. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5272372>.
11. Boyd S, Sastry S. Necessary and sufficient conditions for parameter convergence in adaptive control. *Automatica* 1986; **22**(6):629–639.
12. Chowdhary G, Johnson EN. Concurrent learning for convergence in adaptive control without persistency of excitation. *IEEE Conference on Decision and Control*, 2010; 3674–3679.
13. Chowdhary G, Johnson EN. Theory and flight test validation of a concurrent learning adaptive controller. *Journal of Guidance Control and Dynamics* March 2011; **34**(2):592–607.
14. Chowdhary G. Concurrent learning for convergence in adaptive control without persistency of excitation. PhD Thesis, Georgia Institute of Technology, Atlanta, GA 2010.
15. Durham W. Constrained control allocation. *AIAA Journal of Guidance, Control, and Dynamics* 1993; **16**:717–772.
16. Wise K. Applied controls research topics in the aerospace industry. *Decision and Control, 1995., Proceedings of the 34th IEEE Conference on*, vol. 1, 1995; 751–756 vol.1, doi:10.1109/CDC.1995.479070.
17. Yucelen K, CNN Kim AJ. Derivative-free output feedback adaptive control of an aeroelastic generic transport model. *Guidance Navigation and Control Conference*, AIAA: Portland, OR, 2011.
18. Ioannou PA, Sun J. *Robust Adaptive Control*. Prentice-Hall: Upper Saddle River, 1996.
19. Anderson B. Exponential stability of linear equations arising in adaptive identification. *IEEE Transactions on Automatic Control* Feb 1977; **22**(1):83–88, doi:10.1109/TAC.1977.1101406. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1101406>.
20. Kim N. Improved methods in neural network based adaptive output feedback control, with applications to flight control. PhD Thesis, Georgia Institute of Technology, Atlanta Ga 2003.

21. Liberzon D. *Handbook of Networked and Embedded Control Systems*, chap. Switched Systems. Birkhauser: Boston, 2005; 559–574.
22. Chowdhary G, Johnson EN. A singular value maximizing data recording algorithm for concurrent learning. *American Control Conference*, 2011. Submitted.
23. Nguyen N. Asymptotic linearity of optimal control modification adaptive law with analytical stability margins. *Infotech@AIAA conference*, Atlanta, GA, 2010.
24. Ioannou PA, Kokotovic PV. *Adaptive Systems with Reduced Models*. Springer Verlag: Secaucus, NJ, 1983.
25. Narendra KS, Annaswamy AM. Robust adaptive control in the presence of bounded disturbances. *IEEE Transactions on Automatic Control* 1986; **AC-31**(4):306–315.
26. Chowdhary G, Johnson EN. Least squares based modification term for adaptive control. *IEEE Conference on Decision and Control*, Atlanta, 2010.