Concurrent Learning Adaptive Control for Systems with Unknown Sign of Control Effectiveness

Benjamin Reish and Girish Chowdhary

***Abstract*— Most Model Reference Adaptive Control methods assume that the sign of control effectiveness is known. These methods cannot be used in situations that require adaptation in presence of unknown sign of control effectiveness, such as when controls reverse on an flexible aircraft due to wing twist, or when actuator mappings are unknown. To handle such situations, a Concurrent Learning Model Reference Adaptive Control method is developed for linear uncertain dynamical systems where the sign of the control effectiveness, and parameters of the control allocation matrix, are unknown. The approach relies on simultaneous estimation of the control allocation matrix using online recorded and instantaneous data concurrently, while the system is being actively controlled using the online updated estimate. It is shown that the tracking error and weight error convergence depends on how accurate the estimates of the unknown parameters are. This is used to establish the necessity for purging the concurrent learning history stacks, and three algorithms for purging the history stack for eventual re-population are presented. It is shown that the system states will not grow unbounded even when the sign of the control effectiveness is unknown, and the control allocation matrix is being estimated online. Simulations validate the theoretical results.**

I. Introduction

Adaptive control of uncertain systems has been studied and applied to many areas [1]–[6]. A widely used approach is Model Reference Adaptive Control (MRAC) which drives the instantaneous tracking error to zero. In most MRAC approaches, the control allocation matrix of the plant (e.g. the B matrix in the standard state-space representation: $\dot{x} = Ax + Bu$) is assumed known, or the sign of the diagonal elements of the matrix is assumed known [1]–[8]. These methods cannot be used in situations that require adaptation in presence of unknown B matrix, or unknown sign of control effectiveness, such as when controls reverse on an flexible aircraft due to wing twist, or when an autopilot must control an Unmanned System whose actuator mappings are unknown. Authors have studied the problem of uncertain allocation matrices: Lavretsky et al. used a diagonal scaling matrix and adaptive laws to approximate the symmetric control effectiveness loss [9]; Somanath showed uncertainties in the allocation matrix could be handled if the allocation matrix was some multiplicative combination of an uncertain matrix and a known matrix [10]; and Tao et al. show how to control a multi-input system where some of the actuators became inoperative at a fixed or varying location of actuation at some unknown time which does not address the unknown allocation matrix problem, it just reduces the allocation matrix by the number of stuck actuators [11]. The works just discussed all use MRAC formulations with one model, but there are other formulations available. Multiple model adaptive control [12], [13] techniques can handle uncertain B matrix by choosing between candidate models for the B matrix, however, these candidate models need to be available before hand. The retrospective cost adaptive control method [14], [15] could also potentially handle uncertain B matrix, however in the retrospective cost adaptive method stability of the system may not be guaranteed while data is being collected for learning. It appears that very little work has been done on the general case of controlling a system when the control allocation matrix is unknown, and in particular, the sign of control effectiveness is unknown.

The main contribution of this paper is a hybrid MRAC method that directly uses an online updated estimate of the unknown allocation matrix in the adaptation laws while guaranteeing the convergence of the tracking error to zero. To drive the tracking error to zero the adaptive weights need to attain their ideal values [2]. Without Persistency of Excitation (PE) in the input as Boyd and Sastry showed in [16], the adaptive weights are not guaranteed to converge to the ideal values under traditional gradient based MRAC update laws. Concurrent Learning MRAC (CL-MRAC) is a method that guarantees the adaptive weights converge to the ideal with only finite excitation [17]–[19]. CL-MRAC achieves this by using specifically selected online data concurrently with instantaneous data for adaptation. However, existing results in CL-MRAC do not extend to the case when the sign of the control effectiveness matrix is unknown. In [8] we presented simulation results of CL-MRAC working with uncertain B matrix, however, that work did not provide rigorous justification of stability when the sign of the control effectiveness was unknown. In particular, unlike previously studied CL-MRAC approaches which assumed the allocation matrix was known [17], [18], the case for unknown control allocation matrix requires that the concurrent learning histories be purged once the estimate of the control allocation matrix converges close to its actual value. Toward that end, three methods for defining when to purge (shock) the history stacks are presented, and their effect on stability of the architecture analyzed.

This paper is laid out with an introduction, then an outline of Concurrent Learning Model Reference Adaptive Control and a discussion of how the CL-MRAC process was changed

B. Reish and Assist. Prof. G. Chowdhary are with the school of mechanical and aerospace engineering, Oklahoma State University, Stillwater, OK, ben.reish@okstate.edu, girish.chowdhary@okstate.edu

to be able to identify a control allocation matrix is in section II. Then section III discusses the need for shocking while IV covers the Lyapunov stability of the system and section V contains the results of a simulation of this method followed by the conclusion is in section VI.

II. CL-MRAC WITH UNCERTAIN ALLOCATION MATRIX

Concurrent Learning Model Reference Adaptive Control is built on the foundation of MRAC [20]. In order to describe the method for unknown B matrix CL-MRAC, the framework of CL-MRAC is outlined beginning with its MRAC roots.

Let $D \in \mathbb{R}^n$ be compact and let $x(t) \in D$ be the state vector of the system. Consider a linear time invariant dynamical system of the form

$$\begin{aligned}\dot{x}(t) =& Ax(t) + Bu(t) & (1)\\ \dot{x}_{rm}(t) =& A_{rm}x_{rm}(t) + B_{rm}r(t) & (2)\\ u(t) =& K^T(t)x(t) + K_r^T(t)r(t) & (3)\\ \dot{e}(t) =& A_{rm}e(t) + B\widetilde{K}^T(t)x(t) + B\widetilde{K_r}^T(t)r(t) & (4)\end{aligned}$$

where A is the known state matrix and B is an unknown input allocation matrix. The linear reference model is (2) where A_{rm} and B_{rm} are chosen to be Hurwitz and the reference model input is $r(t) = -B_{rm}^{\dagger}A_{rm}x_{des}(t)$ where the † means the psuedoinverse operation (Tikhonov regularization [21] may be used to guarantee the inverse exists). The desired state, x_{des} is a bounded signal. The control law applied to the plant is defined by [1], [4] as (3). Defining the error as $e(t) = x(t) - x_{rm}(t)$, then it can be shown ([1], [4], [20]) that the time derivative of the tracking error is (4) where $\widetilde{K}(t) = K(t) - K^*$ and $\widetilde{K_r}(t) = K_r(t) - K_r^*$ are the weight errors. The starred parameters come from the assumption of matched uncertainty which guarantees the existence of ideal constant gains K^*, K_r^* such that $A + BK^{*T} = A_{rm}$ and $BK_r^{*T} = B_{rm}$. It is interesting to note that the sign of the allocation matrix does not affect the matching assumption.

CL-MRAC differs from MRAC in that it operates on an estimate of the weight error as well as the tracking error. The weight error terms for the j^{th} recorded data point for adaptive gains K and K_r are:

$$\begin{aligned}\epsilon_{Kj} =& K^T x_j - B^{\dagger}(\dot{x}_j - A_{rm}x_j - B_{rm}r_j - B\epsilon_{Krj}) & (5)\\ \epsilon_{Krj} =& K_r^T r_j - B^{\dagger}B_{rm}r_j & (6)\end{aligned}$$

Note that the evaluation of these errors requires an estimate of $\dot{x}_j$ for a recorded data point. The estimate can be computed using fixed point smoothing. This method has been validated through several flight tests to yield acceptable results [18], [19], [22], [23], furthermore, [24] shows that the CL-MRAC framework is robust to noise in estimating $\dot{x}_j$.

The K and K_r update laws from CL-MRAC (see [8] [25]) use the error terms from (5) and (6) to define the following:

$$\dot{K}(t) = -\Gamma_x\left(e^T(t)x(t)PB + \sum_{j=1}^{p_{max}} x_j\epsilon_{Kj}^T\right) \quad (7)$$

$$\dot{K_r}(t) = -\Gamma_r\left(e^T(t)r(t)PB + \sum_{j=1}^{p_{max}} r_j\epsilon_{Krj}^T\right) \quad (8)$$

where p_{max} is the maximum number of data points to be stored. The error terms (the summations in (7) and (8)) are part of CL-MRAC and would not be present in MRAC update laws. Boyd and Sastry have shown that the adaptive gains K, K_r would not be guaranteed to converge to their ideal values K^*, K_r^* unless the system input is persistently exciting under MRAC adaptation laws ((7) and (8) without the summations) [16]. Exciting signals can be defined using Tao's definition in [1]: over some interval $[t, t+T]$ where $t > t_0$ and $T > 0$, the input signal is exciting if

$$\int_t^{t+T} u(t)u^T(t)\,\mathrm{d}t \geq \lambda I \quad (9)$$

for some $\lambda > 0$ and I is an identity matrix. But, with CL-MRAC the adaptive gains are shown to converge if the input signals are exciting over a finite time per the following theorem.

***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 $X_k = [x_1, x_2, \ldots, x_p]$ be the history stack matrix containing recorded states, and $R_k = [r_1, r_2, \ldots, r_p]$ 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 of [20]. Then, the concurrent learning weight update laws of (7) and (8) guarantee that the zero solution $(e(t), \widetilde{K}(t), \widetilde{K_r}(t)) \equiv 0$ is globally exponentially stable.

Proof: See [20] for the proof. ■

The classical implementation of MRAC would not operate consistently with an uncertain allocation matrix because the adaptive laws, grounded in the Lyapunov function for the system, assume the sign (and magnitude) of the B matrix. The wrong sign would drive the adaptation in the opposite direction which is called parameter burst. CL-MRAC will bound this growth ([8], [20]) but still the unknown B matrix is not tackled.

Since the B matrix is not known, the controller uses an internal estimate of B, denoted as $\widehat{B}(t)$. The error of the $\widehat{B}(t)$ is $\Delta_e(t) = \dot{x}(t) - Ax(t)$. To take $\widehat{B}$ into account, (4), (5), and (6) are rewritten using $\widehat{B}$ as follows:

$$\begin{aligned}\dot{e}(t) =& A_{rm}e(t) + \widehat{B}(t)\widetilde{K}^T(t)x(t) + \widehat{B}(t)\widetilde{K_r}^T(t)r(t) & (10)\\ \widehat{\epsilon}_K =& K^T x(t) - \widehat{B}^{\dagger}(t)(\dot{x}(t) - A_{rm}x(t) - B_{rm}r(t) & \\ & - \widehat{B}(t)\epsilon_{Kr}) & (11)\\ \widehat{\epsilon}_{Kr} =& K_r^T r(t) - \widehat{B}^{\dagger}(t)B_{rm}r(t) & (12)\\ \epsilon_B =& \widehat{B}(t)u(t) - \Delta_e = \widehat{B}(t)u(t) - \dot{x}(t) + Ax(t) & (13)\end{aligned}$$

and (13) is added to account for the concurrent learning of the $\widehat{B}(t)$ term. The update law for $\widehat{B}(t)$ is chosen as follows:

$$\dot{\widehat{B}}(t) = -\Gamma_B \left(u(t)u^T(t)\widetilde{B}^T(t) + \sum_{i=1}^{p_{max}} u_i u_i^T \widetilde{B}^T(t) \right) \quad (14)$$

where $\widetilde{B}(t) = \widehat{B}(t) - B$. And, (7) and (8) are rewritten as

$$\dot{K}(t) = -\Gamma_x \left(e^T(t)x(t)P\widehat{B}(t) + \sum_{j=1}^{p_{max}} x_j {\epsilon_K}_j^T \right) \quad (15)$$

$$\dot{K}_r(t) = -\Gamma_r \left(e^T(t)r(t)P\widehat{B}(t) + \sum_{j=1}^{p_{max}} r_j {\epsilon_{Kr}}_j^T \right) \quad (16)$$

due to $\dot{\widetilde{K}}(t) = \dot{K}(t) - \dot{K}^* = \dot{K}(t)$ and $\dot{\widetilde{K_r}}(t) = \dot{K}_r(t) - \dot{K}_r^* = \dot{K}_r(t)$ because the starred terms are constants. As in [25], [8], and [18], the history stacks of the concurrent learning mechanism are populated while the signal is exiting per the definition given in (9) up to a given limit, p_{max}.

Assumption 1: The state matrix, A, is assumed to be known.

The knowledge of A is a restrictive assumption, however, there is a lack of results on adaptive control with unknown sign of control effectiveness even with this assumption. Future work will try to relax the assumption about the knowledge of A along the empirical evidence presented in [8].

III. Shocking

The concurrent learning history stacks are empty at $t = t_0$ and are filled with data per the algorithm explored in [20]. The following lemma shows that $\widetilde{B}$ goes to zero.

***Lemma** 1:* Consider the system of (1), the control law of (3), and the weight update law of (14), then by *Theorem 1*, $\widetilde{B}(t) \to 0$ as $t \to \infty$.

Hence, by the Lemma 1, there exists a time, $t_s > 0$, such that $\|\widetilde{B}(t_s)\| \leq \epsilon_0$, where ϵ_0 is a small positive constant. Before t_s, the data stored in the history stacks has estimates of K^* and K_r^* formed by using $K^{*T} = \widehat{B}^{\dagger}(A_{rm} - A)$ and $K_r^{*T} = \widehat{B}^{\dagger}B_{rm}$. However, when these estimates were recorded, the sign of $\widehat{B}$ could have been opposite of B, and the values of $\widehat{B}$ could have been very different since the estimate had not converged yet. As the theoretical results show later, this incorrect data in the history stack helps in ensuring that the system response stays bounded, but K and K_r will not converge to their ideal values as long as the incorrect data remains in the stacks. Therefore, the stacks for K and K_r must be *shocked* or purged to remove this incorrect data and allow collection of new data where the estimate, $\widehat{B}$, is closer to the actual allocation matrix. Hence, the condition for shocking the stacks becomes $\|\widetilde{B}(t)\| \leq \epsilon_0$. Since $\widetilde{B}(t)$ is assumed to be not directly measurable, three methods to estimate it are presented: a heuristic (Algorithm 1), by hypothesis testing (Algorithm 2), and by investigating the variance of the expectation of Δ_e (Algorithm 3).

Algorithm 1 estimates $\|\widetilde{B}(t)\|$ by looking at the number of iterations that $|\dot{\widehat{B}}(t)| < \epsilon_0$. When the number of iterations accounts for a second of time that $\dot{\widehat{B}}(t)$ is small, the algorithm shocks the history stacks. While Algorithm 1 is simple to implement, it needs to be told from an outside source that the allocation matrix has changed. The algorithm is not robust to multiple changes of the allocation matrix without outside information.

Algorithm 1: Heuristic, Time Based Method

```
Require: $x(t)$, $x_{rm}(t)$, $\widehat{B}(t)$, $u(t)$, $\mathrm{d}t$
$\dot{x}(t) \Leftarrow$ plant (1)
$\dot{x}_{rm}(t) \Leftarrow$ model (2)
$\dot{\widehat{B}}(t) \Leftarrow$ control law (14)
if $|\dot{\widehat{B}}(t)| < \epsilon_0$ then          ▷ Choose $\epsilon_0$ to be small
    Step counter, $cnt$
end if
if $cnt \cdot \mathrm{d}t == 1\mathrm{sec}$ then          ▷ Heuristic
    Purge history stacks for $K(t)$ and $K_r(t)$
end if
```

To address the limitation of Algorithm 1, a hypothesis test is used to detect changes in the B matrix in Algorithm 2 based on a rolling set of 2-norms of the expectations of $\dot{\hat{x}}(t) - \dot{x}(t)$ values where $\dot{\hat{x}}(t) = Ax(t) + \widehat{B}(t)u(t)$. A rolling set has a fixed number of elements which are replaced, oldest first, by new elements. Using only p_{max} elements in the rolling set, the set is updated whenever $\|u(t)\|$ is greater than zero. The check to purge the history stacks is executed every iteration because due to Lemma 1 concurrent learning will be driving $\|\widetilde{B}(t)\|$ to zero continuously.

Algorithm 2: Hypothesis Test on Expectation of $(\dot{\hat{x}} - \dot{x})$

```
Require: $x(t)$, $x_{rm}(t)$, $\widehat{B}(t)$, $u(t)$
$\dot{x}(t) \Leftarrow$ plant (1)
$\dot{x}_{rm}(t) \Leftarrow$ model (2)
$\dot{\widehat{B}}(t) \Leftarrow$ control law (14)
$\dot{\hat{x}}(t) \Leftarrow Ax(t) + \widehat{B}(t)u(t)$
if $\|u(t)\|_2 > 0$ then
    $x_{rollmean} \Leftarrow \frac{1}{p_{max}} \sum_i^{p_{max}} \left\{ (\dot{\hat{x}}_i - \dot{x}_i)(\dot{\hat{x}}_i - \dot{x}_i)^T \right\}$
    Add $\|(\dot{\hat{x}} - \dot{x})(\dot{\hat{x}} - \dot{x})^T - x_{rollmean}\|_2$ to rolling list, $x_{roll}$
    $x_{rollave} \Leftarrow \mathrm{mean}(x_{roll})$
    $x_{rollS} \Leftarrow \sqrt{\mathrm{var}(x_{roll})}$
end if
Expected-$\dot{x} \Leftarrow (\dot{\hat{x}} - \dot{x})(\dot{\hat{x}} - \dot{x})^T - x_{rollmean}$
UCL $\Leftarrow 4.781 x_{rollS}/\sqrt{p_{max}}$          ▷ 99.99% Upper Control Limit
Tstat$\Leftarrow \|$Expected-$\dot{x}\|_2 \sqrt{p_{max}}/x_{rollS}$          ▷ Statistic
if Tstat<UCL then
    Purge history stacks for $K$ and $K_r$
end if
```

Algorithm 2 estimates $\|\widetilde{B}(t)\|$ with $\|(\dot{\hat{x}}(t) - \dot{x}(t))(\dot{\hat{x}}(t) - \dot{x}(t))^T\|_2$ since $\dot{\hat{x}}(t) - \dot{x}(t) = (\widehat{B}(t) - B)u(t) = \widetilde{B}(t)u(t)$. The upper control limit of the hypothesis test changes based on the mean and standard deviation of the rolling set. The statistic depends on the current measurement as well as the standard deviation. The statistic is the difference between the plant and the controller's estimate of the plant. The outside product of this difference is normed to return a positive number so the statistic is positive semi-definite. The variance approaches zero faster due to being squared. Unlike Algorithm 1, if the allocation matrix changes again, then the difference $\dot{\hat{x}}(t) - \dot{x}(t)$ will be non-zero and the control limit will increase as will the statistic automatically allowing for repeated changes in the B matrix to be detected.

Algorithm 3: Expectation of Δ_e Standard Deviation

```
Require: $x(t)$, $x_{rm}(t)$, $\widehat{B}(t)$, $u(t)$
$\dot{x}(t) \Leftarrow$ plant (1)
$\dot{x}_{rm}(t) \Leftarrow$ model (2)
$\dot{\widehat{B}}(t) \Leftarrow$ control law (14)
$\dot{\hat{x}}(t) \Leftarrow Ax(t) + \widehat{B}(t)u(t)$
if $\|u(t)\|_2 > 0$ then
    $x_{rollmean} \Leftarrow \frac{1}{p_{max}} \sum_i^{p_{max}} \left\{ (\dot{\hat{x}}_i - \dot{x}_i)(\dot{\hat{x}}_i - \dot{x}_i)^T \right\}$
    Add $\|(\dot{\hat{x}} - \dot{x})(\dot{\hat{x}} - \dot{x})^T - x_{rollmean}\|_2$ to rolling list, $x_{Snroll}$
    $x_{Snrollstd} \Leftarrow \sqrt{\text{var}(x_{roll})}$
end if
if $x_{Snrollstd} < tol$ then                         ▷ small tolerance
    Purge history stacks for $K$ and $K_r$
    Set flag to indicate that stacks have been purged
end if
if flag set then
    if $x_{Snrollstd} > tol2$ then
        Set flag to indicate that stacks can be purged
    end if
end if
```

Algorithm 3 is defined in order to shock the stacks less often as compared to Algorithm 2. It focuses on the standard deviation of the expectation of Δ_e, reasoning that when $\|\widetilde{B}(t)\| < \epsilon_0$, then $\sqrt{\text{var}((\dot{\hat{x}}(t) - \dot{x}(t))(\dot{\hat{x}}(t) - \dot{x}(t))^T)}$ will be small. Therefore, two tolerances are given. The first tol is small, around 10^{-8}, and the second tolerance, $tol2$, is much larger ($\approx 10^{-3}$). In this way, the algorithm allows for multiple changes to the allocation matrix. While the selection of tolerances is arbitrary, using the variance of the expectation of the difference between $\dot{\hat{x}}(t)$ and $\dot{x}(t)$ gives a measure of how consistently the expectation is near zero.

IV. Stability of CL-MRAC in presence of Unknown Sign of Control Effectiveness and an Uncertain Allocation Matrix

Lyapunov stability is used to demonstrate the ultimate boundedness of the zero solution by the following theorem.

***Theorem** 2:* Consider the system of (1), the control law of (3), the weight update laws of (14), (15), and (16), and *Theorem 1*, then the zero solution of $[e(t), \widetilde{B}(t), \widetilde{K}(t), \widetilde{K_r}(t)]$ of the closed loop system is uniformly ultimately bounded.

Proof: See [26] for the proof. ■

The proof is too long for this publication and shows that the system is bounded before $\|\widetilde{B}\| < \epsilon_0$ and once the history stacks have been shocked, the $\|\widetilde{B}\|$ will be less than ϵ_0. By Lemma 1, $\widetilde{B}$ will continue to decrease so as $t \to \infty$, the Lyapunov candidate derivative will simplify to (17) which is eventually bounded by a non-positive value by (18), (19), and (20). Each time the history stacks are shocked the value of ϵ_0 decreases. To collect more data, the algorithm in [20] needs exciting input again. The $\hat{\epsilon}$ parameters are the data collected by the concurrent learning algorithm using (11) and (12). The $\Delta\overline{\epsilon}_K, \Delta\overline{\epsilon}_{Kr}$ are the differences between the ϵ parameters defined in (5) and (6) and those actually collected. Therefore, $\widehat{\epsilon}_K = \Delta\overline{\epsilon}_K + \epsilon_K$ and $\widehat{\epsilon}_{Kr} = \Delta\overline{\epsilon}_{Kr} + \epsilon_{Kr}$. Defining two new scalars, $\overline{c}_{xerr}, \overline{c}_{rerr} > 0$, such that $\overline{c}_{xerr} > \| \sum_{i=1}^{p} x_i \Delta\overline{\epsilon}_{K\,i}^T \|$ and $\overline{c}_{rerr} > \| \sum_{j=1}^{p} r_j \Delta\overline{\epsilon}_{Kr\,j}^T \|$ allows the Lyanpunov derivative to be bounded as follows:

$$\begin{aligned} \dot{V} \leq & - c_Q\|e\|^2 - c_{\Omega_x}\|\widetilde{K}\|^2 - c_{\Omega_r}\|\widetilde{K_r}\|^2 + \overline{c}_{xerr}\|\widetilde{K}\| \\ & + \overline{c}_{rerr}\|\widetilde{K_r}\| \end{aligned} \tag{17}$$

$$\|e\| \geq \sqrt{\frac{c_{\Omega_x}\|\widetilde{K}\|^2 + c_{\Omega_r}\|\widetilde{K_r}\|^2 - \overline{c}_{xerr}\|\widetilde{K}\| - \overline{c}_{rerr}\|\widetilde{K_r}\|}{c_Q}} \tag{18}$$

$$\begin{aligned} a_x =& - c_{\Omega_x} \qquad b_x = \overline{c}_{xerr} \\ c_x =& - c_Q\|e\|^2 - c_{\Omega_r}\|\widetilde{K_r}\|^2 + \overline{c}_{rerr}\|\widetilde{K_r}\| \\ \|\widetilde{K}\| \geq & \frac{-b_x - \sqrt{b_x^2 - 4a_x c_x}}{2a_x}. \end{aligned} \tag{19}$$

$$\begin{aligned} a_r =& - c_{\Omega_r} \qquad b_r = \overline{c}_{rerr} \\ c_r =& - c_Q\|e\|^2 - c_{\Omega_x}\|\widetilde{K}\|^2 + \overline{c}_{xerr}\|\widetilde{K}\| \\ \|\widetilde{K_r}\| \geq & \frac{-b_r - \sqrt{b_r^2 - 4a_r c_r}}{2a_r}. \end{aligned} \tag{20}$$

Now (17) is quadratic in $\|e\|$, $\|\widetilde{K}\|$, and $\|\widetilde{K_r}\|$. The compact set outside of which the solutions converge can be estimated by beginning with $\|e\|$ and holding $\|\widetilde{K}\| > 0$ and $\|\widetilde{K_r}\| > 0$, then $\dot{V} \leq 0$ if (18) holds. Moving to $\|\widetilde{K}\|$ and assuming $\|e\| \geq 0$ and $\|\widetilde{K_r}\| \geq 0$, then in this situation $\dot{V} \leq 0$ if (19) holds. Now considering $\|\widetilde{K_r}\|$ assuming $\|e\| \geq 0$ and $\|\widetilde{K}\| \geq 0$, in this situation $\dot{V} \leq 0$ if (20) holds. The bounds in (18), (19), and (20) will continue to decrease as long as there is excitation because the values of $\overline{c}_{xerr}$ and $\overline{c}_{rerr}$ will reduce and the values of c_{Ω_x} and c_{Ω_r} will continue to increase due to concurrent learning. Then as $t \to \infty$, the original concurrent learning results of [20] and *Theorem 1* are recovered.

V. Simulation Study

In this section, an example is presented where the controller is placed in a system where the state matrix is known, but the input allocation matrix is not (a general case). The system state and the input allocation matrices are defined as follows:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0.5 \\ -1 & 2 & -3 \end{bmatrix} \quad B = \begin{bmatrix} -1 & 0.1 & 0.1 \\ 0.1 & 0.9 & 0 \\ 0.5 & 0 & -0.5 \end{bmatrix}$$

The reference model is chosen to be Hurwitz with an identity input allocation matrix and the estimate of B is $\widehat{B}(0) = \mathbb{I}_3$. Note that $\widehat{B}(0)$ is significantly different from B, not only in magnitude, but also the signs of parameters on the main diagonal differ. The reference input for state X_1 is set to 2 for the first 5 seconds. Then the reference for state X_2 is set to 2 from 15 to 25 seconds. Then the reference for state X_3 is set to 2 from 35 to 45 seconds. Then state X_1 is set to 1, state X_2 is set to -1, and state X_3 is set to 0.5 from 60 seconds to 80 seconds. To show that the system will track reference inputs where state X_2 is the derivative of state X_1 and state X_3 is the derivative of state X_2, from 85 to 95 seconds, the reference input, R_1, is a sine wave, its derivative for state R_2, and its second derivative for state R_3.

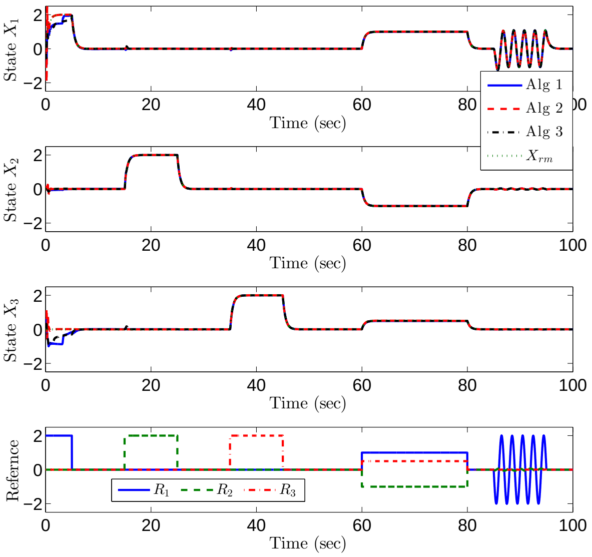

Fig. 1. Time History of Reference Model Tracking Using Algorithms 1, 2, and 3. The Reference $r(t)$ is shown on its own. Note the tracking is quite good after each algorithm shocks the stacks.

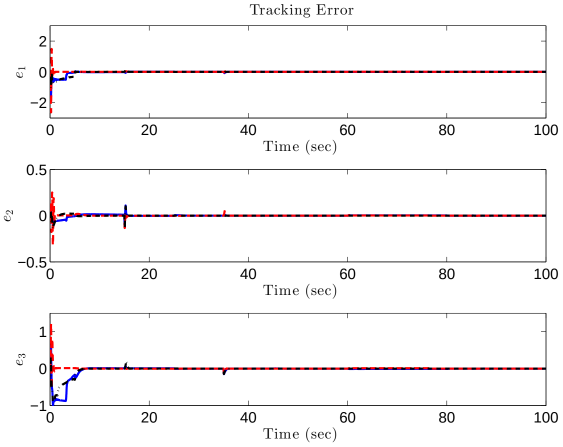

Fig. 2. System Tracking Errors Shown Using Algorithms 1 (solid), 2 (dashed), and 3 (dash dot). Note that the error reduces quickly once the stacks are shocked.

In Fig. 1, the system states are shown tracking the reference model states. The tracking is quite good after an initial disparity. This is to be expected, since the controller has the wrong sign of B when the system is initialized. Note that even with the initial disparity, the system states remain bounded. The disparity starts to reduce immediately as each of the algorithms shocks the history stacks. It should be noted that baseline MRAC laws diverged for the presented case of unknown sign of the B matrix, and hence their response is not shown. The reference input is also shown in Fig. 1. In Fig. 2, the error between the system states and the reference model states is shown with the three algorithms. The error reduces as the B matrix is identified and the K matrices converge.

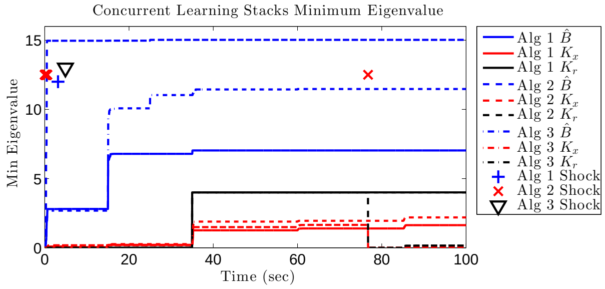

Fig. 3. Minimum Eigenvalues of the $\widehat{B}$, K, and K_r History Stacks Using Algorithms 1, 2, and 3 (The markers denote when one of the algorithms purged the history stacks)

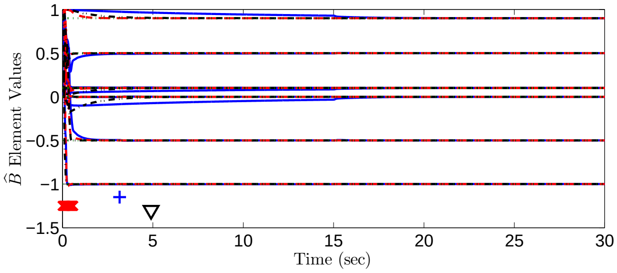

Fig. 4. $\widehat{B}$ Convergence Using Algorithms 1 (solid), 2 (dashed), and 3 (dash dot), Ideal Values (dotted). Note that the dashed lines of Alg. 2 arrive faster than the others at the ideal values. (markers same as Fig. 3)

The convergence of $\widehat{B}$, K, and K_r depends on the minimum eigenvalue of the respective history stack matrix. A time history of the minimum eigenvalues of the three history stacks are displayed in Fig. 3 with the different algorithms noted. The markers in Fig. 3 and 4 indicate when the stacks were purged under the different algorithms: a + for Algorithm 1, x's for Algorithm 2, and a $\bigtriangledown$ for Algorithm 3. Algorithm 2 purges the stacks many times in the first second and then does so once again at about 78 seconds which is about the same time that all three state errors converge back to zero in Fig. 2. Notice that the minimum eigenvalues for Algorithm 2 K and K_r drop to zero, but begin increasing again as the input becomes exciting (95 sec). Algorithms 1 and 3 purge the stacks only once.

Fig. 4 indicates the convergence of the $\widehat{B}$ matrix to the B matrix under the three algorithms. Note that only the first 30 seconds of the simulation are shown in Fig. 4. The balance is quite similar to the last 5 seconds shown in the figure.

The time history of the element values of K and K_r converge to their ideal values (the dotted lines) in Fig. 5. The elements have arrived in the first 45 seconds which concludes the state by state excitation of the input. It is interesting to note that the bounding of the growth of the K values aligns with each algorithm's time for shocking the history stacks, but the steps toward the ideal values align with steps in the input.

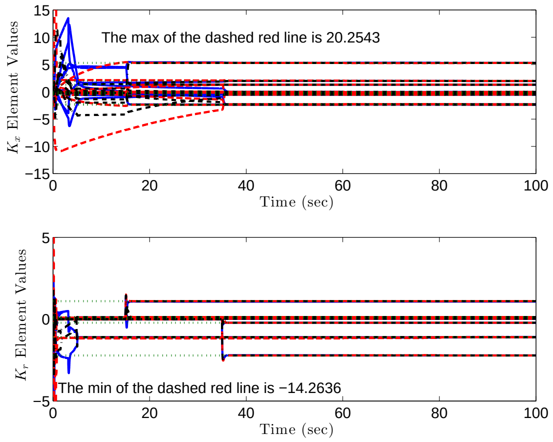

Fig. 5. K and K_r Convergence Using Algorithms 1 (solid), 2 (dashed), and 3 (dash dot), Ideal Values (dotted). Note that the steps in convergence align with steps in the input, but the bounding of growth aligns with shocking the stacks.

VI. Conclusion

This paper describes how concurrent learning can be used to handle the general case of an uncertain allocation matrix in a dynamical system. The approach relies on simultaneous estimation of the unknown control allocation matrix while the system is actively controlled using that estimate. Lyapunov argument were used to show that the approach will result in system states being bounded. The concurrent learning history stack purging or shocking is required to remove data retained while the B matrix estimates were far from their true values. Three algorithms for shocking the history stack were presented. Algorithms 2 and 3 have the advantage to be able to handle multiple changes in the control allocation matrix. Algorithm 1 would not be able to do that in its present form. The simulation results show that the method works on a 3 state linear model, with the controller initialized with the wrong sign of the B matrix. These results establish the feasibility of using a learning based adaptive controller to handle uncertainties in control allocation matrices.

Acknowledgment

This work was supported in part by University of Oklahoma NASA Cooperative Agreement No. NNX13AB21A.

References

[1] G. Tao, *Adaptive Control Design and Analysis*. New York: Wiley, 2003.

[2] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems*. Englewood Cliffs: Prentice-Hall, 1989.

[3] P. A. Ioannou and P. V. Kokotovic, *Adaptive Systems with Reduced Models*. Secaucus, NJ: Springer Verlag, 1983.

[4] K. J. Astrm and B. Wittenmark, *Adaptive Control*, 2nd ed. Readings: Addison-Weseley, 1995.

[5] G. Chowdhary, M. Mühlegg, J. P. How, and F. Holzapfel, "Concurrent learning adaptive model predictive control," in *Advances in Aerospace Guidance, Navigation and Control*, Q. Chu, B. Mulder, D. Choukroun, E.-J. Kampen, C. Visser, and G. Looye, Eds. Springer Berlin Heidelberg, 2013, pp. 29–47. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-38253-6_3

[6] T. E. Gibson, A. M. Annaswamy, and E. Lavretsky, "Adaptive systems with closed-loop reference models: Stability, robustness and transient performance," *arXiv preprint arXiv:1201.4897*, 2012.

[7] N. Kim, "Improved methods in neural network based adaptive output feedback control, with applications to flight control," Ph.D. dissertation, Georgia Institute of Technology, Atlanta Ga, 2003.

[8] B. Reish, G. Chowdhary, K. Ure, and J. P. How, "Concurrent learning adaptive control in the presence of uncertain control allocation matrix," in *Proc. of the Conference on Guidance, Navigation, and Control*. AIAA, 2013, pp. 1–19.

[9] E. Lavretsky, "Combined/composite model reference adaptive control," *Automatic Control, IEEE Transactions on*, vol. 54, no. 11, p. 26922697, nov. 2009.

[10] A. Somanath, "Adaptive control of hypersonic vehicles in presence of actuation uncertainties," SM, Massachusetts Institute of Technology, Cambridge, MA, June 2010.

[11] G. Tao, S. M. Joshi, and X. Ma, "Adaptive state feedback and tracking control of systems with actuator failures," *Automatic Control, IEEE Transactions on*, vol. 46, no. 1, p. 7895, 2001.

[12] K. Narendra and J. Balakrishnan, "Adaptive control using multiple models," *Automatic Control, IEEE Transactions on*, vol. 42, no. 2, pp. 171 –187, feb 1997.

[13] Z. Han and K. Narendra, "Multiple adaptive model for control," in *Conference on Decision and Control*. Atlanta: IEEE, December 2010, p. 6065.

[14] M.-J. Yu, Y. Rahman, E. M. Atkins, I. V. Kolmanovsky, and D. S. Bernstein, "Minimal modeling adaptive control of the nasa generic transport model with unknown control-surface faults," in *AIAA Guidance, Navigation, and Control (GNC) Conference, Proc. of*, Boston, MA, August 2013, p. 121.

[15] M. A. Santillo and D. S. Bernstein, "Adaptive control based on retrospective cost optimization," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 2, p. 289304, March-April 2010.

[16] S. Boyd and S. Sastry, "Necessary and sufficient conditions for parameter convergence in adaptive control," *Automatica*, vol. 22, no. 6, p. 629639, 1986.

[17] G. Chowdhary and E. N. Johnson, "Concurrent learning for convergence in adaptive control without persistency of excitation," in *49th IEEE Conference on Decision and Control*, 2010, p. 36743679.

[18] G. Chowdhary, M. Mhlegg, and E. Johnson, "Exponential parameter and tracking error convergence guarantees for adaptive controllers without persistency of excitation," *International Journal of Control*, 2013, accepted.

[19] G. Chowdhary, T. Yucelen, M. Mühlegg, and E. N. Johnson, "Concurrent learning adaptive control of linear systems with exponentially convergent bounds," *International Journal of Adaptive Control and Signal Processing*, 2012.

[20] G. Chowdhary, M. Mühlegg, T. Yucelen, and E. Johnson, "Concurrent learning adaptive control of linear systems with exponentially convergent bounds," *International Journal of Adaptive Control and Signal Processing*, 2012. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1002/acs.2297/abstract

[21] A. N. Tikhonov, "On the stability of inverse problems," *Dokl. Akad. Nauk SSSR*, vol. 39, no. 5, p. 195198, 1943.

[22] G. Chowdhary, T. Wu, M. Cutler, and J. P. How, "Rapid transfer of controllers between UAVs using learning based adaptive control," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2013.

[23] G. Chowdhary and E. N. Johnson, "Theory and flight test validation of a concurrent learning adaptive controller," *Journal of Guidance Control and Dynamics*, vol. 34, no. 2, p. 592607, March 2011.

[24] M. Mühlegg, G. Chowdhary, and E. Johnson, "Concurrent learning adaptive control of linear systems with noisy measurements," in *AIAA Guidance, Navigation, and Control Conference, Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics*, 2012.

[25] G. Chowdhary, "Concurrent learning for convergence in adaptive control without persistency of excitation," Ph.D. dissertation, Georgia Institute of Technology, Atlanta, GA, 2010.

[26] B. Reish and G. Chowdhary, "Concurrent Learning Adaptive Control for Systems with Unknown Sign of Control Effectiveness," IEEE, 2014, contains stability proof. [Online]. Available: http://daslab.okstate.edu/paperpubs