Concurrent Learning Adaptive Control in the Presence of Uncertain Control Allocation Matrix

Ben Reish [*], Girish Chowdhary, [†]
Distributed Autonomous Systems Laboratory, Oklahoma State University, Stillwater, OK, USA

Kemal Ure [‡], Jonathan P. How[§]
Aerospace Controls Laboratory, MIT, Cambridge, MA, USA

We consider the problem of solving adaptive control problems in the presence of an uncertain control allocation matrix. Concurrent learning adaptive control architectures are presented that simultaneously stabilize the system and estimate parameters of the system B matrix. Sufficient conditions for the boundedness of the architecture are presented. The validity of our approach is demonstrated through numerical simulations on linear multi-input multi-output models.

I. Introduction

Adaptive control is a well-established area within control theory. For adaptation to uncertain, nonlinear elements of a control system, the Model Reference Adaptive Control (MRAC) framework provides a means to achieve asymptotic tracking error convergence or ultimate boundedness of tracking error[1–9]. Particularity, adaptive flight control has been widely studied, both experimental and theoretical. For example, model reference adaptive controllers for both fixed wing and rotary wing unmanned aerial vehicles (UAVs) have been developed and tested[10–17]. Gavrilets et al., Chowdhary and Johnson and others have extended direct adaptive control methods to fault tolerant control[18,19]. However, most of these techniques focus on minimizing the instantaneous tracking error and are not guaranteed to learn a model of the underlying uncertainty of the system model unless system input is persistently exciting, a condition that might be difficulty or costly to enforce under typical operating conditions. Concurrent learning adaptive controllers use data, selected online and recorded, concurrently with current data for adaptation. Concurrent learning adaptive controllers can guarantee exponential closed loop stability of both the tracking and the modeling error without requiring persistency of excitation if the plant uncertainty can be linearly parameterized[20,21]. The convergence rate has also been shown to be directly proportional to the minimum singular value of the history stack and a singular value maximizing data recording algorithm was also presented[22]. Concurrent learning controllers have been shown to simultaneously guarantee stability while learning matched modeling uncertainty for multivariable uncertain linear dynamical systems[23]. However, in that work, changes to the parameters of the control effectiveness matrix (the B matrix in the usual linear dynamical system representation: $\dot{x} = Ax + Bu$) were not considered.

[*]M.S. candidate in the Distributed Autonomous Systems Lab at OSU, ben.reish@okstate.edu
[†]Assistant Professor, Oklahoma State University, girish.chowdhary@okstate.edu
[‡]PhD Candidate, Massachusetts Institute of Technology
[§]Richard C. Maclaurin Professor of Aeronautics and Astronautics, MIT, jhow@mit.edu

In fact, adaptation to actuator uncertainties that can be represented as variations in the B matrix is a challenging problem in MRAC. Many MRAC algorithms can handle matched uncertainties in A matrix, however need to assume that the structure of the B matrix, and its parameters are fully known[1–4,10]. This assumption can be restrictive when dealing with failures or actuator degradation. Lavretsky et al.[24] have developed adaptive laws that can handle symmetric scaled loss of control effectiveness representable as a diagonal scaling matrix. Somanath shows that most MRAC methods yield only local stability when B is unknown and provide algorithms that can handle certain classes of uncertainties in the B matrix[25]. Tao et al. developed adaptive controllers for the case of unknown B matrix and actuator failures under the condition that the sign of the ideal gain matrix satisfying appropriate matching conditions is known[26]. However, it may be difficult to know the sign of the ideal gain when B is unknown.

In this paper, we present a concurrent learning approach aimed at handling uncertainties in the B matrix. The insight behind our approach is that the difficulties in dealing with actuator uncertainties could be reduced if the controller attempted to learn parameters associated with the B matrix. In particular, we leverage the exponential parameter convergence properties of concurrent learning adaptive controllers to learn changes to the B matrix after a known switch in the parameters occurs. The adaptive controller switches to using the online estimated B matrix only after certain conditions on convergence are satisfied. Thus avoiding any undesirable effects the transient in learning may cause. We further show that our control architecture guarantees that the system states remain bounded while the parameters are being learned. We demonstrate the validity of the approach on a linear multi input multi output helicopter model.

II. Concurrent Learning Adaptive Control with Uncertain B Matrix

II.A. Vehicle Level Linear Switched Dynamical System Formulation

Let a switching linear dynamical system be defined as follows,

$$\dot{x}(t) = A_{\sigma(t)}x^i(t) + B_{\sigma(t)}u^i(t), i = 1, \ldots, n_{veh} \tag{1}$$

The subscript σ will be dropped from the formulation and the analysis and design for the control laws will focus on a single switch of the dynamical system, (A_{old}, B_{old}) will represent the dynamics of the system before the failure and (A_{new}, B_{new}) will represent the dynamics after the switch.

II.B. Simultaneous Model Estimation and Stabilization using Concurrent Learning Adaptive Control

Concurrent learning adaptive control is used as the main tool for this objective, due to its exponential convergence guarantees[23] and its ability to estimate unknown parameters without persistency of excitation[21]. However, like other existing MRAC architectures, the concurrent learning adaptive controller relies on the knowledge of B_{new} matrix, which is unavailable after the failure. It is possible to update the estimate of B_{new} while controlling the system, however depending on the magnitude of perturbation in the B_{old} matrix, this approach may render the system unstable.

To overcome this problem, a separate concurrent learning model estimation loop is implemented, which estimates the A_{new} and B_{new} matrices after the failure in an online fashion and updates the B_{old} matrix in the control law only after the model estimation converges. While the model estimation loop is running, the concurrent learning adaptive controller keeps the system bounded by using the B_{old} matrix. Once the model estimation converges, the controller uses this new information to improve the tracking performance. Details of the concurrent model estimation algorithm are given in subsection II.B.1 while the concurrent learning adaptive controller with mismatched B matrices is presented at subsection II.B.2. Finally the details on how the model estimation and control loops work together is provide in subsection II.B.3.

II.B.1. Concurrent Learning Model Estimation

Assume that the state of the system $x(t)$ and the input signal $u(t)$ are available or can be constructed from the measurements. Let $(\hat{A}, \hat{B})$ represent the estimate of (A, B) and let $\dot{\hat{x}} = \hat{A}x + \hat{B}u$. Define error dynamics as

$$\epsilon(t) = ([\hat{A}, \hat{B}] - [A, B])[x^T(t), u^T(t)]^T = \dot{\hat{x}} - \dot{x}.$$

The objective of the model estimation algorithm is to drive $\epsilon(t) \to 0$ asymptotically.

Let x_i, u_i for $i \in 1, \ldots, p$ be the data points recorded online at times t_i. Concurrent learning model estimation updates are given as follows,

$$\dot{\hat{A}}(t) = -\Gamma_A[x(t)\epsilon(t) - \sum_{j=1}^{p} x_j \epsilon_j] \tag{2}$$

$$\dot{\hat{B}}(t) = -\Gamma_B[u(t)\epsilon(t) - \sum_{j=1}^{p} u_j \epsilon_j], \tag{3}$$

where $\Gamma_A, \Gamma_B > 0$. The following theorem can be proven using arguments in[21,22]

Theorem 1. *Assume that the control signal $u(t)$ is exciting over a finite interval and that the data points for concurrent learning are selected using the singular value maximizing algorithm (Algorithm 1 from[22]), then, $\hat{A} \to A$ and $\hat{B} \to B$ exponentially fast.*

Remark 1. *Note that model estimation law Eqs. 2 and 3 requires the knowledge of $\dot{x}$. Mühlegg et. al showed that if a noisy estimate of $\dot{x}$ is available, then the adaptation law is guaranteed to be uniformly bounded under some additional assumptions[27]. Also, note that the $\epsilon(t)$ feedback term on the model update law can be dropped, and estimates of ϵ_j can be improved using a fixed point smoother[23].*

II.B.2. Concurrent Learning Adaptive Control with Unknown B Matrix

Let $\dot{x}_{rm} = A_{rm}x_{rm} + B_{rm}r$ represent the dynamics of the reference model to be used in MRAC architecture, where $r(t)$ is the reference signal. Assume A_{rm} is Hurwitz and P is the solution to the Lyapunov equation $A_{rm}^T P + PA_{rm} + Q = 0$, where $Q \in \mathbb{R}^{n\times n}$ is a positive definite matrix.

Let the control law be of the form

$$u(t) = K^T(t)x(t) + K_r^T(t)r(t), \tag{4}$$

where $K \in \mathbb{R}^{n\times m}$ and $K_r \in \mathbb{R}^{1\times m}$. Assume that matching conditions hold, i.e. there exits as K^* and K_r^* such that,

$$A + BK^{*T} = A_{rm} \tag{5}$$

$$BK_r^{*T} = B_{rm} \tag{6}$$

After substituting the control law to the system equation and performing algebraic manipulations, the following form of the error dynamics $e(t) = x - x_{rm}$ is obtained,

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

The objective of the controller is to update K, K_r such that error dynamics $e(t) = x - x_{rm}$ and weight error dynamics $\tilde{K} = K - K^*, \tilde{K_r} = K - K_r^*$ are asymptotically stable. When the control assignment matrix B is available to the designer, it has been shown in[23], that concurrent learning satisfies this objective exponentially fast without needing persistency of excitation. In this section we will extend this work to deal with the case where the control allocation matrix, B, is unknown.

Let $\hat{B}$ be a fixed estimate of the B matrix which is available to the controller. Assume that the A matrix is known. Let x_i, r_i be the i^{th} data point recorded online, define the error variables $\hat{\epsilon}_{K_j}$ and $\hat{\epsilon}_{K_r}$ as,

$$\hat{\epsilon}_{K_j} = \hat{B}^{-1}(\dot{x_j} - A_{rm}x_j - B_{rm}r_j) \tag{8}$$

$$\hat{\epsilon}_{K_{r_j}} = K_r^T r_j - \hat{B}^{-1}B_{rm}r_j. \tag{9}$$

The concurrent learning weight update laws are given as,

$$\dot{K} = -\Gamma_x[xe^T P\hat{B} + \sum_{j=1}^{p} x_j \hat{\epsilon}_{K_j}^T] \tag{10}$$

$$\dot{K_r} = -\Gamma_r[re^T P\hat{B} + \sum_{j=1}^{p} r_j \hat{\epsilon}_{K_{r_j}}^T] \tag{11}$$

The following theorem states that as long as the estimate $\hat{B}$ is close enough to B, the system $[e, \tilde{K}, \tilde{K_r}]$ is bounded.

Theorem 2. *Consider the system in Eq. 1, the control law in Eq. 4, and weight update laws in Eqs. 10 and 11. Assume that the reference signal $r(t)$ is exciting over a finite interval and that the data points for concurrent learning are selected using the singular value maximizing algorithm (Algorithm 1 from[23]). In addition assume that $\|B - \hat{B}\|$ is bounded, $\mathtt{sgn}(B) = \mathtt{sgn}(\hat{B})$, and pairs (A, B) and $(A, \hat{B})$ are controllable. Then the zero solution $(e, \tilde{K}, \tilde{K_r})$ of the closed loop system is uniformly ultimately bounded.*

The proof is presented in the appendix.

Given the above theorem, the following corollary is immediate if one considers the case when the control allocation matrix B is completely known.

Corollary 1. *In addition to the assumptions of Theorem 1, if $\tilde{B} = 0$, i.e. the control allocation matrix B is known to the controller, then the zero solution $(e, \tilde{K}, \tilde{K_r})$ of the closed loop system is exponentially stable, that is the tracking error and weight error dynamics converge to zero exponentially fast.*

Proof. The proof of this theorem follows from that of Theorem 2 with $\tilde{B} = 0$. This result has been proven in Theorem 1 of[23] under the assumption that $\tilde{B} = 0$. □

II.B.3. Safe Model Estimation Using Switched Control

The model estimation method described in section II.B.1 and the control law described in section II.B.2 can be combined to build a switched control model estimation algorithm that concurrently learns the model and stabilizes the UAV after failures. We assume that there is a separate health monitoring system that signals to controller that a failure is occurred. The basic idea of the switched control algorithm is to use the control law developed in section II.B.2 to keep the system bounded without the knowledge of (A_{new}, B_{new}), while the parameter estimation law developed in section II.B.1 estimates the new system model in the background. Once the new system model is estimated, controller switches to newly learned model to stabilize the system. This process is described in Algorithm 1.

Algorithm 1: Safe Model Estimation and Control Algorithm

Input: Initial state $x(0)$, Model before the failure (A_{old}, B_{old}). Reference Model (A_{rm}, B_{rm}),Time Tolerance t_{end}, Model Tolerance m_{tol}

$(A_{new}, B_{new}) \leftarrow (A_{old}, B_{old})$
while $(t \leftarrow \texttt{CurrentTime}()) \leq t_{end}$ **do**
  $x(t) \leftarrow \texttt{ObserveState}(t)$
  $(A_{new}, B_{new}) \leftarrow$ Eqs. 2 and 3
  $u(t) \leftarrow$ Eqs. 4, 10,11 using $(\hat{B} = B_{old})$
  if $\|A_{new} - A_{old}\| < m_{tol}$ **and** $\|B_{new} - B_{old}\| < m_{tol}$ **then**
    $(A_{old}, B_{old}) \leftarrow (A_{new}, B_{new})$

Corollary 2. *Assume* $t_{tol} \leq \max(\Delta t_{switch}, \Delta t_{est})$*, where* Δt_{est} *is the time required by model estimation algorithm to drive* $[\tilde{A}, \tilde{B}] \rightarrow 0$*. Then the Algorithm 1 ensures that the system* $[e, \tilde{K}, \tilde{K_r}]^T$ *is stable.*

Proof. Boundedness of the system during the model estimation process is guaranteed by Theorem 2 and since the model estimation law is proven to be exponentially fast by Theorem 1, as long as t_{tol} is big enough, Algorithm 1 is guaranteed to converge to the new model after the failure in finite time. After Algorithm 1 terminates, Corollary 1 implies that the control law, Eq. 4, will stabilize the system. □

III. Simulation Results

III.A. Performance of the Safe Model Estimation Algorithm

In this subsection, stand alone performance of the safe model estimation and control algorithm (Algorithm 1) is verified on a helicopter model taken from[28] and a fixed wing unmanned aerial vehicle (UAV) from[29]. The helicopter model is a fairly large scale linear system, consisting of 11 states and 4 inputs. The fixed wing UAV has 8 states and 4 inputs.

In the simulation, the system was initialized with complete knowledge of A and B and a failure was induced at the 20 second mark. Then the controller uses Algorithm 1 to continue operation.

The velocities track the references with some error which decays in about 20 seconds. The angular velocities track the reference command with some error that is also decaying after the failure in Fig. 1. After the failure, total model estimation error rapidly converges towards zero near the 60 seconds mark in Fig. 2. Between $20 - 60$ seconds mark, controller uses the B_{old} matrix to keep the system bounded. After the 60 seconds mark controller switches to B_{new} matrix provided by the model estimator to control the system, and tracking error rapidly decreases to zero.

Figures 3 and 4 are from the fixed wing unmanned aerial vehicle from[29]. Fig. 3 shows sideslip angle tracking, roll rate, yaw rate, and Euler Roll angle rate of the UAV. Fig. 4 shows the evolution of certain elements of the A and B matrices. The estimated A and B matrices are approaching the values of the old A and B matrices after the failure which happened at the 15 second mark in these two figures.

III.B. Performance of a Combined Concurrent Learning Controller and Estimator

In this section we analyze the performance of the adaptive controller that is implemented through the following three equations:

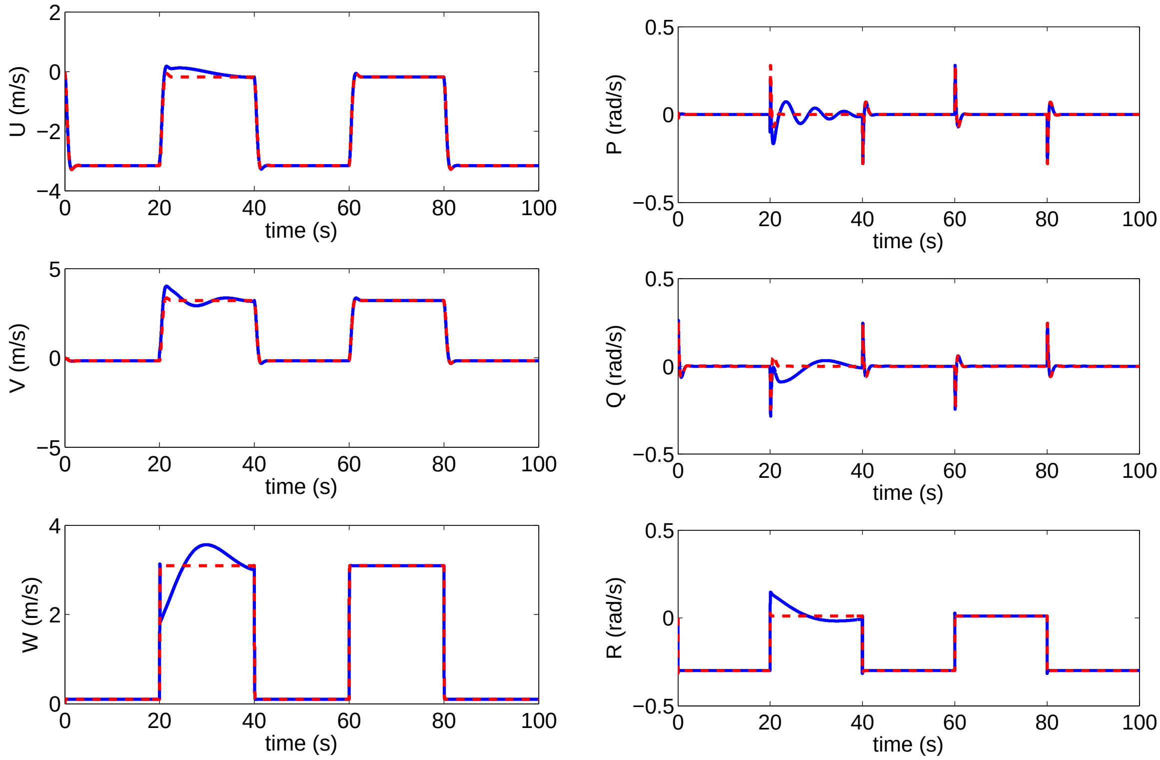

Figure 1. Reference and actual velocity states over time on the left. U, V, W denote the velocity in X, Y, Z in body axes. Reference and actual angular velocity states over time on the right. P, Q, R denote the rolling, pitching and yawing velocities.

$$\dot{K} = -\Gamma_x[xe^T P\hat{B} + \sum_{j=1}^{p} x_j \hat{\epsilon}_{K_j}^T] \tag{12}$$

$$\dot{K}_r = -\Gamma_r[re^T P\hat{B} + \sum_{j=1}^{p} r_j \hat{\epsilon}_{K_{r_j}}^T] \tag{13}$$

$$\dot{\hat{B}}(t) = -\Gamma_B[u(t)\epsilon(t) - \sum_{j=1}^{p} u_j \epsilon_j], \tag{14}$$

That is, the concurrent learning controller attempts to figure out the ideal control gains as well as the B matrix at the same time. The A matrix is not estimated. The following known and unstable A matrix is considered:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 2 & -3 \end{bmatrix} \tag{15}$$

III.B.1. Results with known sign of B

Here we examine a system with known sign of the control allocation matrix. We will apply adaptive control, then concurrent learning adaptive control and then concurrent learning with model

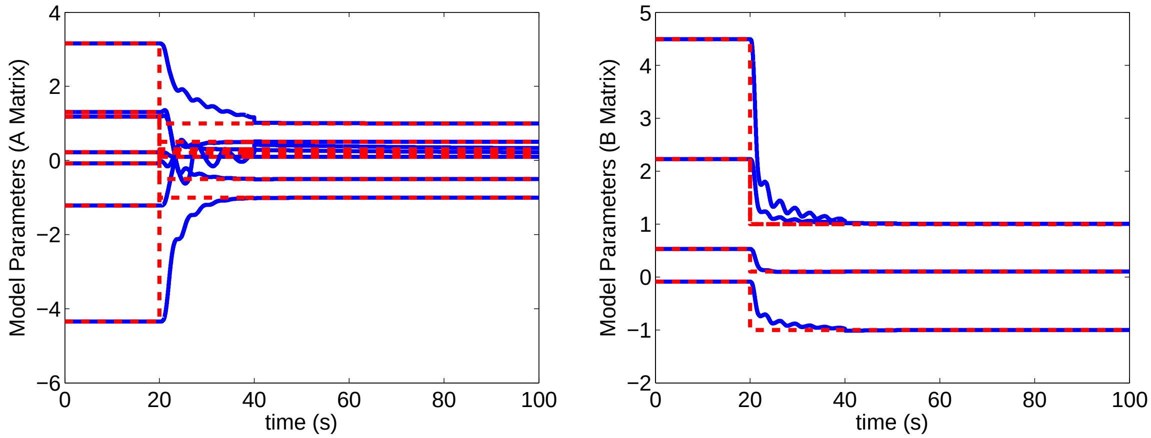

Figure 2. Model parameters (entries of A matrix on the left, entries of the B matrix on the right) over time. The failure occurs around the 20 second mark.

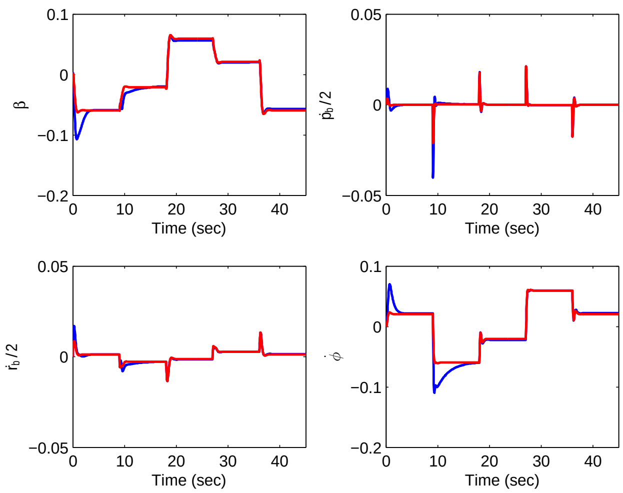

Figure 3. From the UAV model, β is the sideslip angle (top left), $\dot{p_b}/2$ is the roll rate (top right), $\dot{r_b}/2$ is the yaw rate (bottom left), and $\dot{\phi}$ is the Euler Roll angle rate (bottom right).

estimation adaptive control to the system. The B matrix for this simulation is defined to be:

$$B = \begin{bmatrix} 1 & .1 & .1 \\ 0.1 & 0.9 & 0 \\ 0.5 & 0 & 0.5 \end{bmatrix}. \tag{16}$$

The controller starts with the following estimate of the B matrix ($\hat{B}$):

$$\hat{B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{17}$$

which makes B and $\hat{B}$ have the same sign.

As shown in Fig. 5-8, the system works with only a normal adaptation law because the sign of the B matrix is the same as the sign of the $\hat{B}$ matrix. The noise in the signal does have some

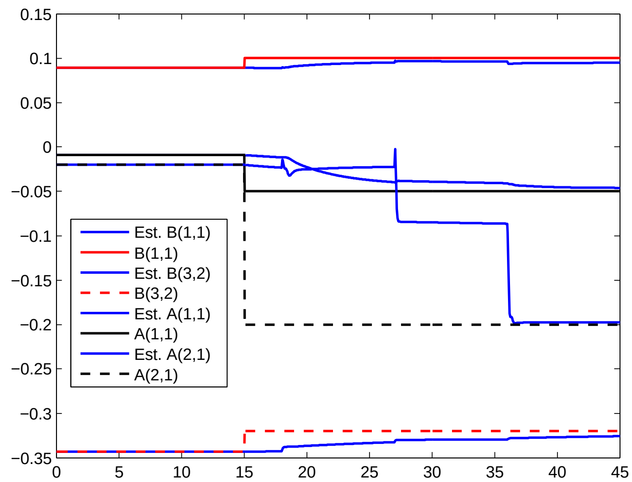

Figure 4. UAV Model parameter estimation (certain entries of A and B) over time with failure at 15 second mark.

effect and note that the weights never arrive to the ideal weights in Fig. 8 which indicates that no learning of the underlying system has taken place. In Fig. 9-12, the system is affected less by the noise in the signal when using concurrent learning to update the adaptive weights. The tracking error is smaller in Fig. 10 than in Fig. 6 with no concurrent learning. The weights never arrive to the ideal weights in Fig. 12 either. So concurrent learning is working to bound the system, but because of the differences between B and $\hat{B}$, the adaptive weights are not arriving at the ideal weights. Then in Fig. 13-16, the system is using concurrent learning and is also estimating the B matrix. The tracking error is on the same order as Fig. 10 and less than when the system did not use concurrent learning. Note that the weights arrive at the ideal weights in Figs. 16. This shows that the controller has learned the underlying model to be able to drive the weights to their ideal values.

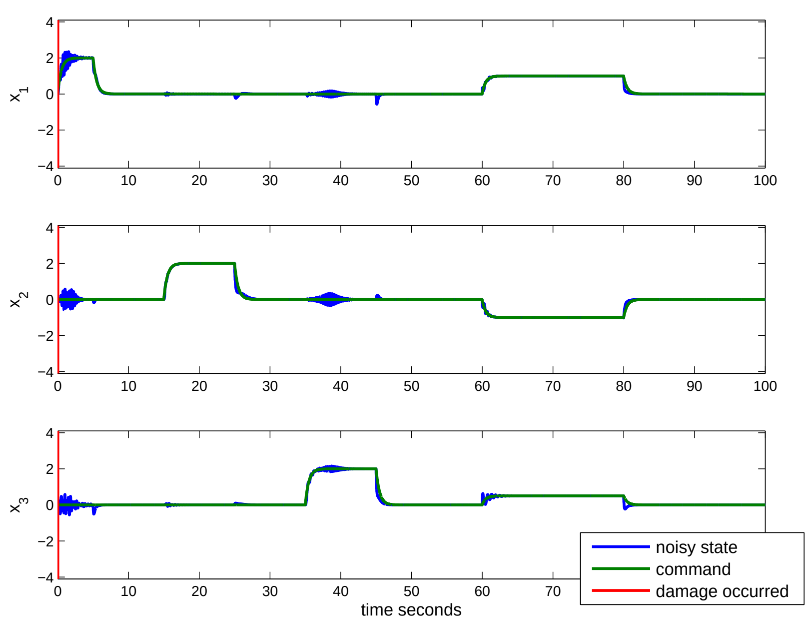

Figure 5. Tracking Performance over Time (No Concurrent Learning nor Estimation of B Matrix)

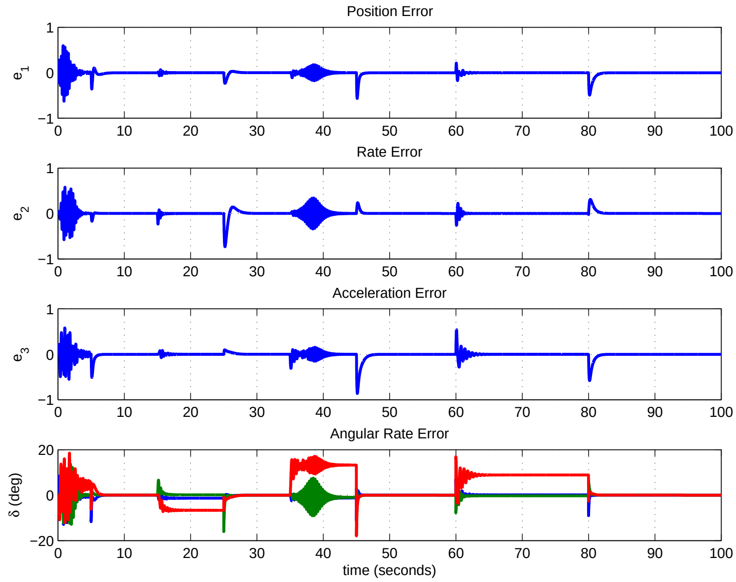

Figure 6. Tracking Error over Time (No Concurrent Learning nor Estimation of B Matrix)

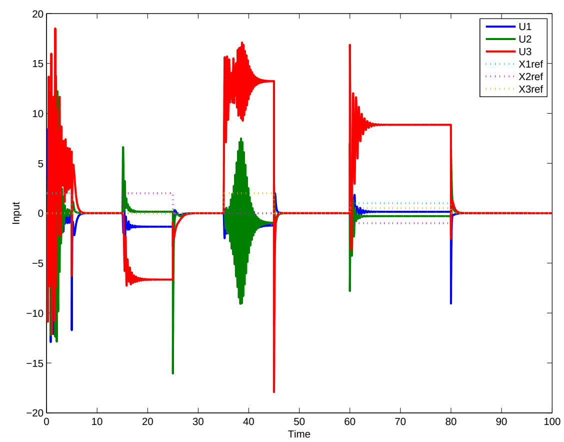

Figure 7. Control Input over Time (No Concurrent Learning nor Estimation of B Matrix)

III.B.2. Results with Unknown sign of B

In this section we present results when the sign of the B matrix is not known, and the controller starts by assuming the wrong sign of B. The true B matrix is:

$$B = \begin{bmatrix} -1 & .1 & .1 \\ 0.1 & 0.9 & 0 \\ 0.5 & 0 & -0.5 \end{bmatrix}. \tag{18}$$

The $\hat{B}$ matrix is still an identity matrix so the sign is off on two elements. Using the same update law as previous example, the system is controlled while the B matrix is identified. Running this

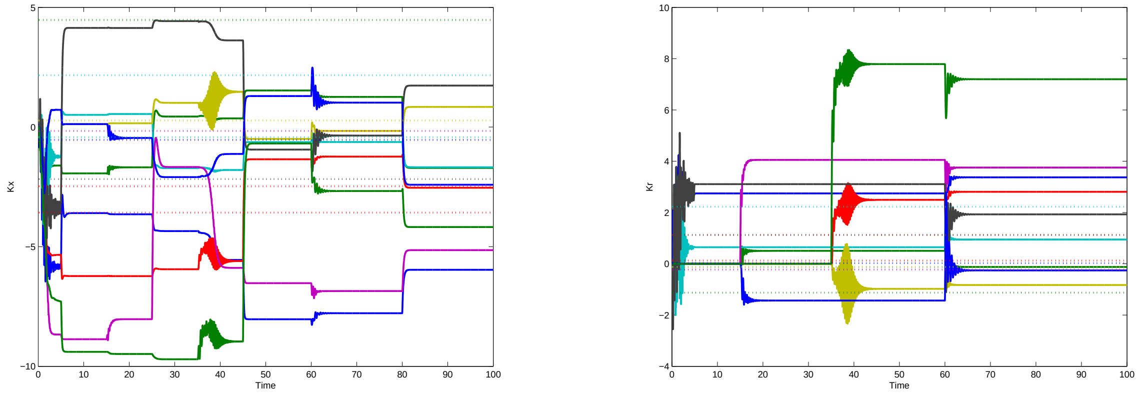

Figure 8. Adaptive Weights over Time (No Concurrent Learning nor Estimation of B Matrix)

simulation with the adaptive law with no concurrent learning or estimation of B causes unbounded output.

In Figs. 18-20, the system is controlled while the B matrix is estimated. This is a particularly difficult problem because the sign of the initial estimate of the B matrix ($\hat{B}$) is not correct. Correspondingly it can be seen that the controller, including the adaptive weights in Fig. 20, diverges in the beginning, but converges after the estimate $\hat{B}$ converges to the right B matrix. The control input in Fig. 19 is larger initially than in Figs. 7, 11, or 15, but then settles down to comparitively similar magnitudes as $\hat{B}$ converges to B. Once again, the adaptive weights converge to their ideal values in Fig. 20.

IV. Conclusions

We have shown an algorithm to safely switch between controllers while the new model is estimated and displayed results of using that algorithm on a helicopter model and a fixed wing unmanned aerial vehicle. We have also shown a that concurrent learning adaptive control in the context of a model reference adaptive controller identifies the control allocation matrix even when the sign of the estimate, $\hat{B}$, is not equal to the sign of the B matrix. Concurrent learning adaptive control also drives adaptive weights to their ideal values and causes exponentially quick convergence with less error than adaptive control without concurrent learning.

V. Appendix

Proof of Theorem 1

Proof. Let $\zeta = [e, \texttt{vec}(K), K_r]^T$ and define the following Lyapunov candidate,

$$V(\zeta) = \frac{1}{2}e^T Pe + \frac{1}{2}tr(\tilde{K}^T\Gamma_x^{-1}\tilde{K}) + \frac{1}{2}tr(\tilde{K_r}^T\Gamma_r^{-1}\tilde{K_r}). \tag{19}$$

It is possible to bound the Lyapunov candidate above and below with the following positive definite functions.

$$\frac{1}{2}\min(\lambda_{\min}(P), \lambda_{\min}(\Gamma_x^{-1}), \Gamma_r^{-1})\|\zeta\|^2 \leq V(\zeta)$$
$$\leq \frac{1}{2}\max(\lambda_{\max}(P), \lambda_{\max}(\Gamma_x^{-1}), \Gamma_r^{-1})\|\zeta\|^2$$

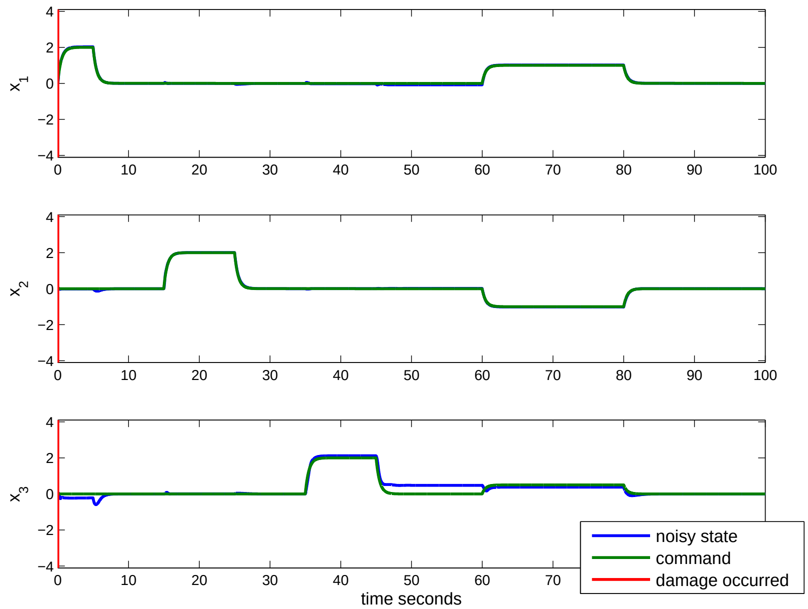

Figure 9. Tracking Performance over Time (Concurrent Learning but no Estimation of B Matrix)

Let $[t_1, t_2, ..., t_p], t_{i+1} > t_i$ be the sequence of times where each data was recorded. Taking the derivative of the Lyapunov candidate along trajectories of the system for each interval $[t_i, t_{i+1}]$ and performing simplifications, we get:

$$\dot{V}(\zeta) = -\frac{1}{2}e^TQe + (\tilde{K}^T x e^T + \tilde{K_r}^T r e^T)P(B - \hat{B})$$
$$-tr(\tilde{K}^T \sum_{j=1}^{p} x_j \hat{\epsilon}_{K_j}^T) - tr(\tilde{K_r}^T \sum_{j=1}^{p} r_j \hat{\epsilon}_{K_r}^T)$$

Define, $\tilde{\epsilon} = \hat{\epsilon} - \epsilon, \tilde{B} = \hat{B} - B$,

$$\dot{V}(\zeta) = -\frac{1}{2}e^TQe - (\tilde{K}^T x e^T + \tilde{K_r}^T r e^T)P\tilde{B}$$
$$-tr(\tilde{K}^T \sum_{j=1}^{p} x_j x_j^T \tilde{K}^T) - tr(\tilde{K_r}^T \sum_{j=1}^{p} r_j r_j^T \tilde{K_r}^T)$$
$$-tr(\tilde{K}^T \sum_{j=1}^{p} x_j \tilde{\epsilon}_{K_j}^T) - tr(\tilde{K_r}^T \sum_{j=1}^{p} r_j \tilde{\epsilon}_{K_r}^T).$$

Note that the matrix $\Omega = \sum_{j=1}^{p} x_j x_j^T$ is positive definite. Bounding the inequality above using norms and the triangle inequality,

$$\dot{V}(\zeta) \leq -\frac{1}{2}\lambda_{\min}(Q)\|e\|^2 - \frac{1}{2}\lambda_{\min}(\Omega)\|\tilde{K}\|^2$$
$$-\frac{1}{2}(\sum_{j=1}^{p} r_j^2)\|\tilde{K_r}\|^2 + \|\tilde{K}\|\|x_{rm}\|\|e\|\|P\|\|\tilde{B}\| \quad (20)$$
$$+\|\tilde{K}\|\|e\|^2\|P\|\|\tilde{B}\| + \|\tilde{K_r}\|\|r\|\|e\|\|P\|\|\tilde{B}\|$$
$$+\|\tilde{K}\|\|P\|\|\sum_{j=1}^{p} x_j \tilde{\epsilon}_K\| + \|\tilde{K_r}\|\|P\|\|\sum_{j=1}^{p} r_j \tilde{\epsilon}_{K_r}\|$$

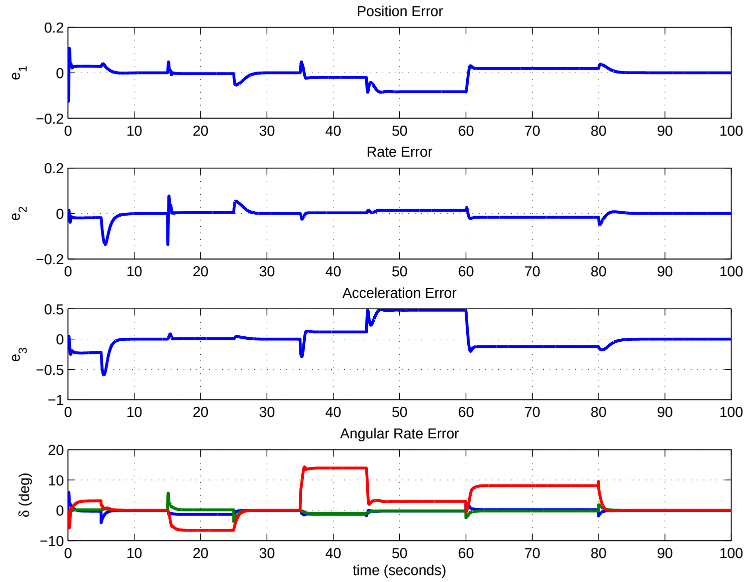

Figure 10. Tracking Error over Time (Concurrent Learning but no Estimation of B Matrix)

Note that as long as the matrix Ω is full ranked (which can be guaranteed if the reference signal r is exciting over a finite interval $[0, T]$[23]) and $\texttt{sgn}(B) = \texttt{sgn}(\hat{B})$, the first three terms on the right hand side of the inequality above are negative-definite. Conservative bounds on the rest of the right hand side terms can be found as follows: The matrix A_{rm} of the reference model is assumed to be Hurwitz and the reference signal r is always bounded, therefore there exist scalars $c_r, c_{x_{rm}} > 0$ such that $\|x_{rm}\| < c_{rm}, \|r\| < c_r$. Note that $\|\tilde{B}\|$ is assumed to be bounded, therefore there exists a scalar c_B such that $\|P\|\|\tilde{B}\| < c_B$. Finally, note that the error terms $\hat{\epsilon}$ are functions of recorded data x_j, r_j, which are bounded by assumption and do not evolve with time. Therefore there exist scalars $c_{\epsilon_x}, c_{\epsilon_r} > 0$ such that $\|P\|\|\sum_{j=1}^{p} x_j \tilde{\epsilon}_K\| < c_{\epsilon_x}, \|P\|\|\sum_{j=1}^{p} r_j \tilde{\epsilon}_{K_r}\| < c_{\epsilon_r}$. Hence,

$$\begin{aligned} \dot{V}(\zeta) \leq & -\frac{1}{2}\lambda_{\min}(Q)\|e\|^2 - \frac{1}{2}\lambda_{\min}(\Omega)\|\tilde{K}\|^2 \\ & -\frac{1}{2}(\sum_{j=1}^{p} r_j^2)\|\tilde{K_r}\|^2 + c_{rm}c_B\|\tilde{K}\|\|e\| + c_B\|\tilde{K}\|\|e\|^2 \\ & + c_r c_B\|\tilde{K_r}\|\|e\| + c_{\epsilon_x}\|\tilde{K}\| + c_{\epsilon_r}\|\tilde{K_r}\|. \end{aligned} \tag{21}$$

Therefore, for sufficiently large $\lambda_{\min}(Q)$, $\lambda_{\min}(\Omega)$, and $\sum_{j=1}^{p} r_j^2$, $\dot{V}(\zeta) \leq 0$ outside of a compact set. To see that the set is indeed compact, note that the terms on the right hand side of Eq. 21 yield three quadratic inequalities in $\|e\|$, $\|\tilde{K}\|$ and $\|\tilde{K_r}\|$. A conservative estimate of the positively invariant set within which the solutions are bounded can be found by solving these quadratic inequalities for each variable while assuming that other two variables are non-zero. First we check the case where $\|\tilde{K}\| > 0, \|\tilde{K_r}\| > 0$. In this case,

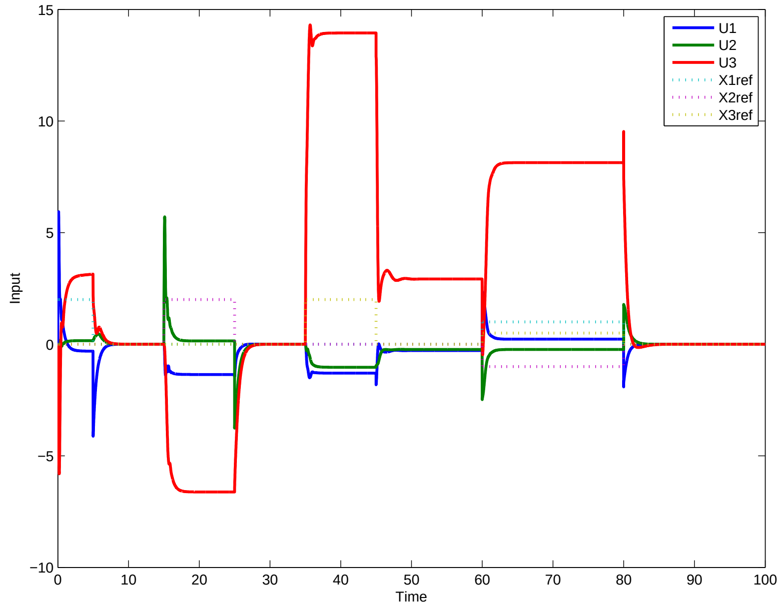

Figure 11. Control Input over Time (Concurrent Learning but no Estimation of B Matrix)

$$
\begin{aligned}
\|e\| &\geq \frac{-b_e + \sqrt{b_e^2 - 4a_e c_e}}{2a_e} \\
a_e &= -\frac{1}{2}\lambda_{\min}(Q) + c_b\|\tilde{K}\| \\
b_e &= c_{rm}c_B\|\tilde{K}\| + c_r c_B\|\tilde{K}_r\| \\
c_e &= -\frac{1}{2}\lambda_{\min}(\Omega)\|\tilde{K}\|^2 - \frac{1}{2}(\sum_{j=1}^{p} r_j^2)\|\tilde{K}_r\|^2 \\
&\quad + c_{\epsilon_x}\|\tilde{K}\| + c_{\epsilon_r}\|\tilde{K}_r\|
\end{aligned} \tag{22}
$$

then $\dot{V}(\zeta) \leq 0$.

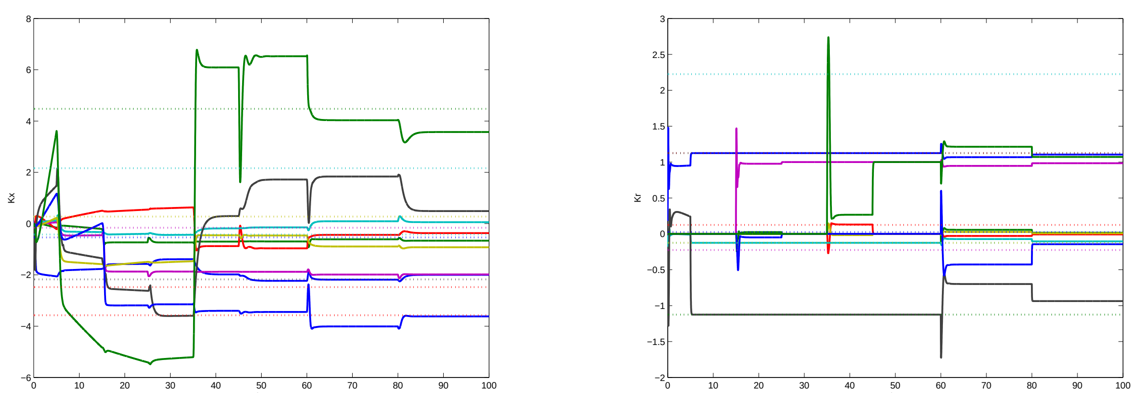

Figure 12. Adaptive Weights over Time (Concurrent Learning but no Estimation of B Matrix)

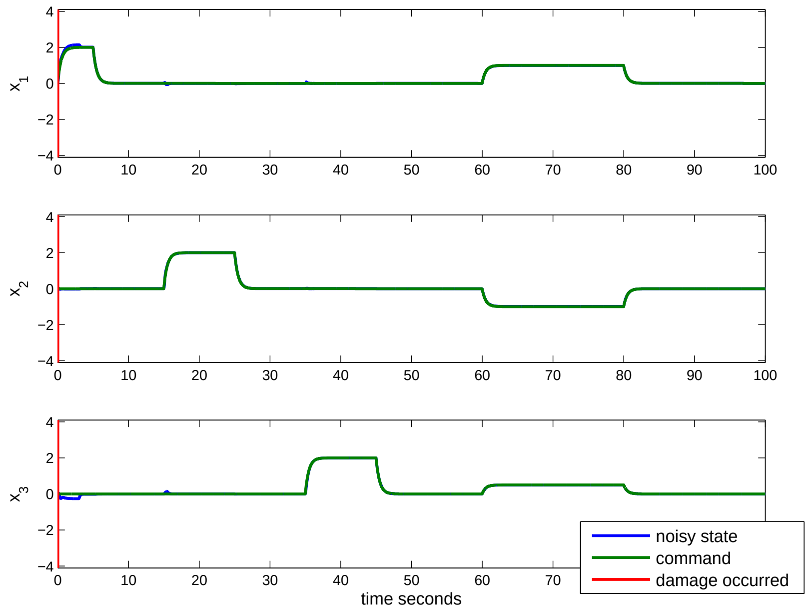

Figure 13. Tracking Performance over Time (Concurrent Learning and Estimation of B Matrix)

Secondly, we consider the case $\|e\| \geq 0, \|\tilde{K}_r\| \geq 0$. In this case, $\dot{V}(\zeta) \leq 0$ if

$$\begin{aligned}
\|\tilde{K}\| &\geq \frac{-b_k + \sqrt{b_k^2 - 4a_k c_k}}{2a_k} \qquad (23)\\
a_k &= -\frac{1}{2}\lambda_{\min}(\Omega),\\
b_k &= c_{rm}c_B\|e\| + c_B\|e\|^2 + c_{\epsilon_x}\\
c_k &= -\frac{1}{2}\lambda_{\min}(Q)\|e\|^2 - \frac{1}{2}(\sum_{j=1}^{p} r_j^2)\|\tilde{K}_r\|^2\\
&\quad + c_r c_B\|\tilde{K}_r\|\|e\| + c_{\epsilon_r}\|\tilde{K}_r\| \qquad (24)
\end{aligned}$$

Then we check the case where $\|e\| \geq 0, \|\tilde{K}\| \geq 0$.

$$\begin{aligned}
\|\tilde{K}_r\| &\geq \frac{-b_{k_r} + \sqrt{b_{k_r}^2 - 4a_{k_r}c_{k_r}}}{2a_{k_r}} \qquad (25)\\
a_{k_r} &= -\frac{1}{2}(\sum_{j=1}^{p} r_j^2),\\
b_{k_r} &= c_r c_B\|e\| + c_{\epsilon_r} \qquad (26)\\
c_{k_r} &= -\frac{1}{2}\lambda_{\min}(Q)\|e\|^2 - \frac{1}{2}\lambda_{\min}(\Omega)\|\tilde{K}\|^2\\
&\quad c_{rm}c_B\|\tilde{K}\|\|e\| + c_B\|\tilde{K}\|\|e\|^2 + c_{\epsilon_x}\|\tilde{K}\|.
\end{aligned}$$

Inequalities 22–25 characterize the compact set outside of which $\dot{V}(\zeta) \leq 0$. Therefore, all solutions will eventually end up within this set, which in turn proves that the system $[e, \tilde{K}, \tilde{K}_r]$ is uniformly ultimately bounded. □

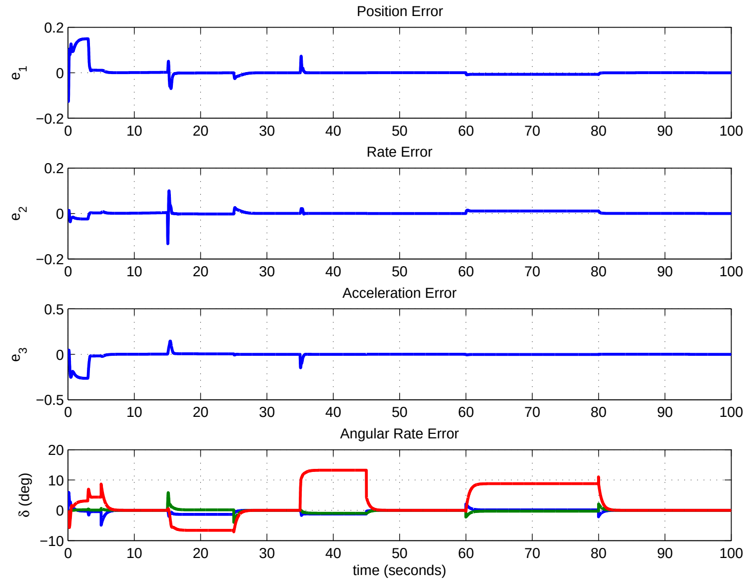

Figure 14. Tracking Error over Time (Concurrent Learning and Estimation of B Matrix)

References

[1] Kumpati S. Narendra and Anuradha M. Annaswamy. *Stable Adaptive Systems*. Prentice-Hall, Englewood Cliffs, 1989.

[2] Petros A. Ioannou and Peter V. Kokotovic. *Adaptive Systems with Reduced Models*. Springer Verlag, Secaucus, NJ, 1983.

[3] Karl Johan Aström and Björn Wittenmark. *Adaptive Control*. Addison-Weseley, Readings, 2nd edition, 1995.

[4] Gang Tao. *Adaptive Control Design and Analysis*. Wiley, New York, 2003.

[5] N. Hovakimyan, B. J. Yang, and A. Calise. An adaptive output feedback control methodology for non-minimum phase systems. *Automatica*, 42(4):513–522, 2006.

[6] Chengyu Cao and N. Hovakimyan. Design and analysis of a novel adaptive control architecture with guaranteed transient performance. *Automatic Control, IEEE Transactions on*, 53(2):586 –591, march 2008.

[7] Tansel Yucelen and Anthony Calise. Derivative-free model reference adaptive control. *AIAA Journal of Guidance, Control, and Dynamics*, 34(8):933–950, 2012. AIAA paper number 0731-5090, doi: 10.2514/3.19988.

[8] Nhan Nguyen, Kalamanje Krishnakumar, John Kaneshige, and Pascal Nespeca. Dynamics and adaptive control for stability recovery of damaged asymmetric aircraft. In *AIAA Guidance Navigation and Control Conference*, Keystone, CO, 2006.

[9] M. Steinberg. Historical overview of research in reconfigurable flight control. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 219(4):263–275, 2005.

[10] A. Calise, N. Hovakimyan, and M. Idan. Adaptive output feedback control of nonlinear systems using neural networks. *Automatica*, 37(8):1201–1211, 2001. Special Issue on Neural Networks for Feedback Control.

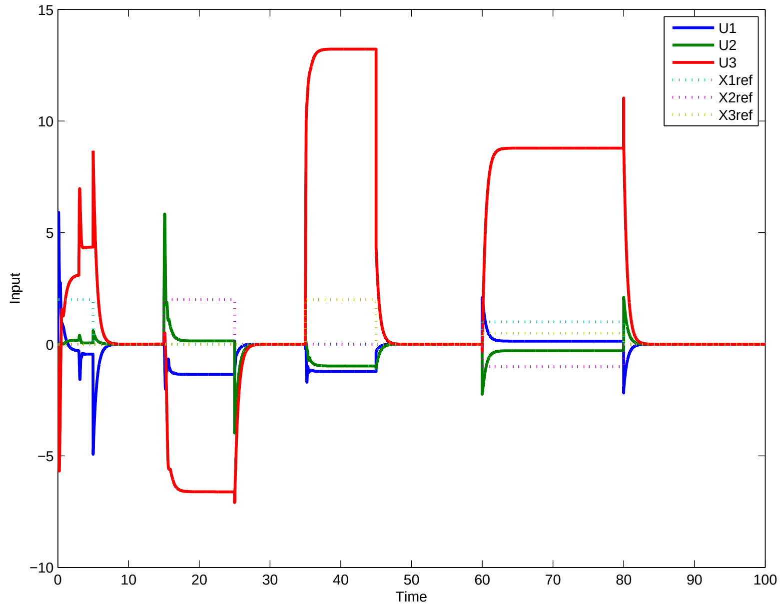

Figure 15. Control Input over Time (Concurrent Learning and Estimation of B Matrix)

[11] E. Johnson and S. Kannan. Adaptive flight control for an autonomous unmanned helicopter. In *Proceedings of the AIAA Guidance Navigation and Control Conference, held at Monterrery CA*, 2002.

[12] Eric N. Johnson and Seung Min Oh. Adaptive control using combined online and background learning neural network. In *Proceedings of CDC*, 2004.

[13] Eric N. Johnson. *Limited Authority Adaptive Flight Control*. PhD thesis, Georgia Institute of Technology, Atlanta Ga, 2000.

[14] Suresh K. Kannan, Adrian A. Koller, and Eric N. Johnson. Simulation and development environment for multiple heterogeneous uavs. In *AIAA Modeling and Simulation Technology Conference*, number AIAA-2004-5041, Providence, Rhode Island, August 2004.

[15] Suresh Kannan. *Adaptive Control of Systems in Cascade with Saturation*. PhD thesis, Georgia Institute of Technology, Atlanta Ga, 2005.

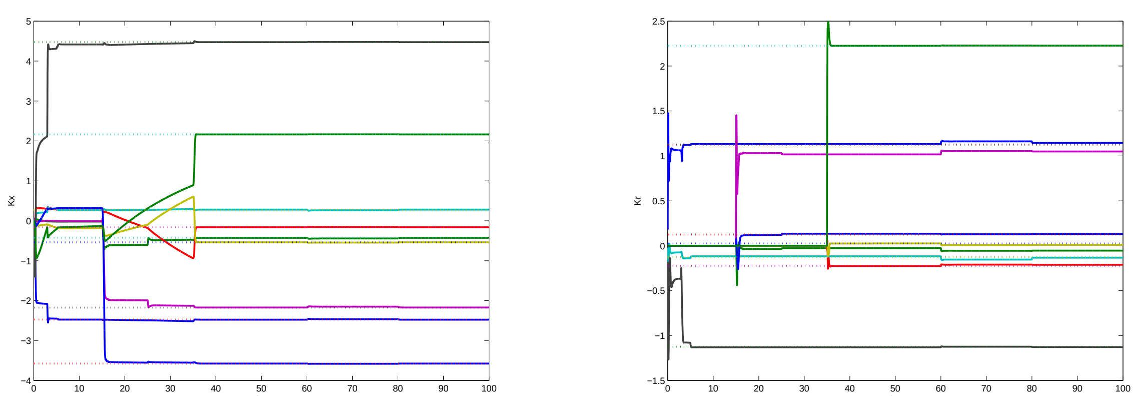

Figure 16. Adaptive Weights over Time (Concurrent Learning and Estimation of B Matrix)

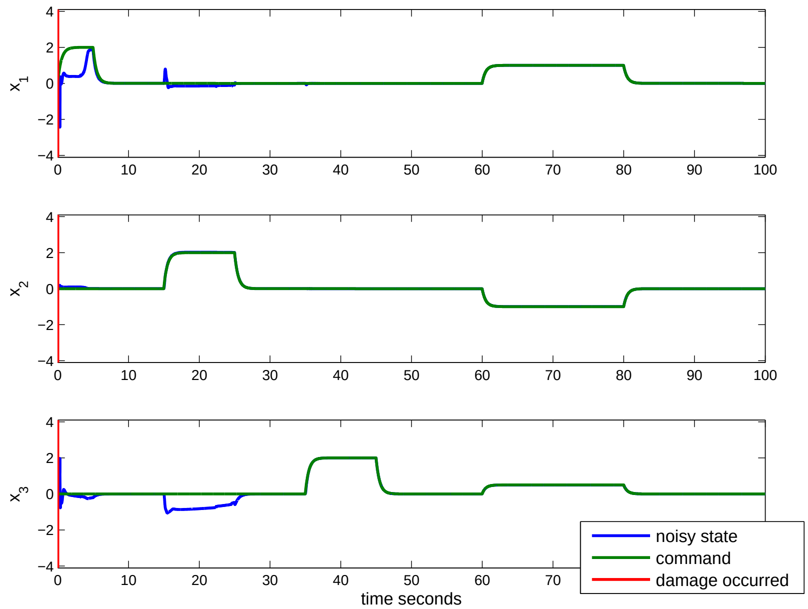

Figure 17. Tracking Performance over Time (Concurrent Learning and Estimation of B Matrix)

[16] Girish Chowdhary and Eric N. Johnson. Theory and flight test validation of a concurrent learning adaptive controller. *Journal of Guidance Control and Dynamics*, 34(2):592–607, March 2011.

[17] Eugene Lavretsky and Kevin Wise. Flight control of manned/unmanned military aircraft. In *Proceedings of American Control Conference*, 2005.

[18] Damien B Jourdan, Michael D Piedmonte, Vlad Gavrilets, and David W Vos. *Enhancing UAV Survivability Through Damage Tolerant Control*, pages 1–26. Number August. AIAA, 2010. AIAA-2010-7548.

[19] Girish Chowdhary, Eric N. Johnson, Rajeev Chandramohan, Scott M. Kimbrell, and Anthony Calise. Autonomous guidance and control of airplanes under actuator failures and severe structural damage. *Journal of Guidance Control and Dynamics*, 2012.

[20] Girish Chowdhary. *Concurrent Learning for Convergence in Adaptive Control Without Persistency of Excitation*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, 2010.

[21] Girish Chowdhary and Eric N. Johnson. Concurrent learning for convergence in adaptive control without persistency of excitation. In *49th IEEE Conference on Decision and Control*, pages 3674–3679, 2010.

[22] Girish Chowdhary and Eric N. Johnson. A singular value maximizing data recording algorithm for concurrent learning. In *American Control Conference*, San Francisco, CA, June 2011.

[23] Girish Chowdhary, Maximillian Muhlegg, Tansel Yucelen, and Eric Johnson. Concurrent learning adaptive control of linear systems with exponentially convergent bounds. *International Journal of Adaptive Control and Signal Processing*, 2012.

[24] E. Lavretsky. Combined/composite model reference adaptive control. *Automatic Control, IEEE Transactions on*, 54(11):2692 –2697, nov. 2009.

[25] Amit Somanath. Adaptive control of hypersonic vehicles in presence of actuation uncertainties. Sm, Massachusetts Institute of Technology, Cambridge, MA, June 2010.

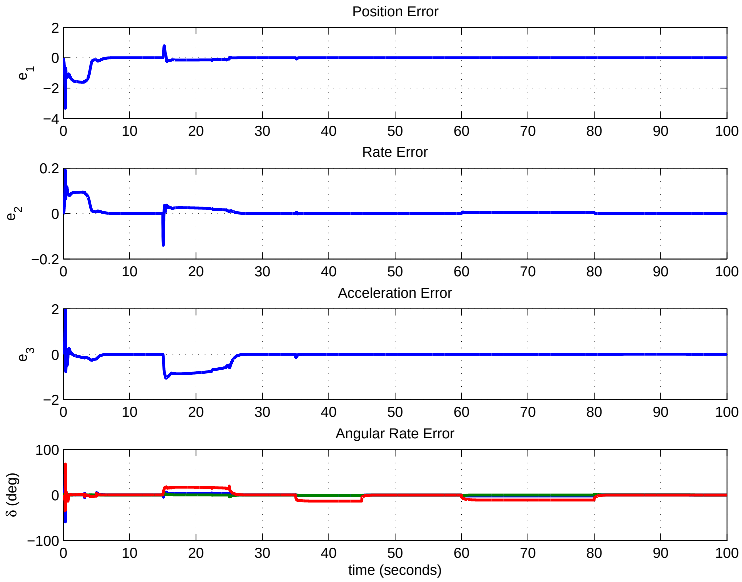

Figure 18. Tracking Error over Time (Concurrent Learning and Estimation of B Matrix)

[26] G. Tao, S.M. Joshi, and X. Ma. Adaptive state feedback and tracking control of systems with actuator failures. *Automatic Control, IEEE Transactions on*, 46(1):78 –95, jan 2001.

[27] M. Muhlegg, G. Chowdhary, and Johnson E. Concurrent learning adaptive control of linear systems with noisy measurement. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, MN, August 2012.

[28] G. Chowdhary and R. Jategaonkar. Aerodynamic parameter estimation from flight data applying extended and unscented kalman filter. In *AIAA Atmospheric Flight Mechanics Conference*. Citeseer, 2006.

[29] Kirk Y. Scheper, Girish Chowdhary, and Eric N. Johnson. Aerodynamic system identification of fixed-wing UAV. In *Proceedings of AIAA AFM*, August 2013.

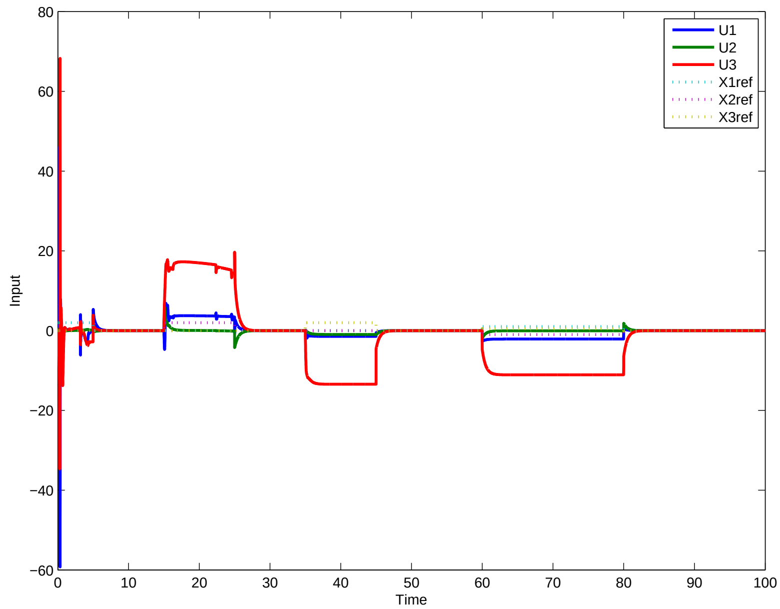

Figure 19. Control Input over Time (Concurrent Learning and Estimation of B Matrix)

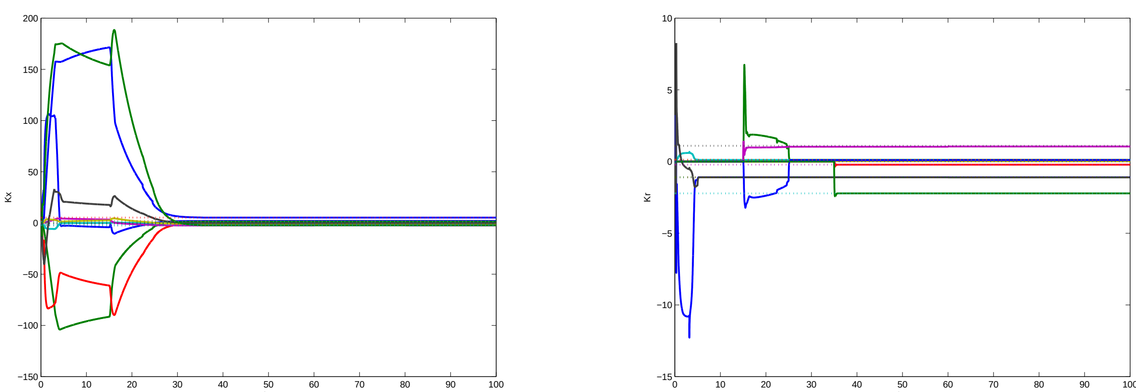

Figure 20. Adaptive Weights over Time (Concurrent Learning and Estimation of B Matrix)