Removing Erroneous History Stack Elements in Concurrent Learning

Stefan Kersting and Martin Buss

***Abstract*— This paper is concerned with erroneous history stack elements in concurrent learning. Concurrent learning-based update laws make concurrent use of current measurements and recorded data. This replaces persistence of excitation by a less restrictive linear independence of the recorded data. However, erroneous or outdated data prevents convergence to the true parameters. We present insights into the convergence properties of concurrent learning and propose a routine to recognize and remove erroneous data online. We characterize erroneous data based on its inconsistency with the current measurement-based update. We numerically validate that the proposed routine restores the tracking ability and improves the convergence properties of concurrent learning.**

I. Introduction

Many control engineering applications are characterized by unknown and time-varying dynamics. In such cases, control systems with adaptive mechanisms enable estimation and tracking of the unknown parameters. Most adaptive systems, however, only guarantee convergence under the assumption that the system is persistently excited. Maintaining persistence of excitation (PE) throughout the entire adaptation period is very restrictive and is not easily monitored at runtime. Also, from a practitioner's point of view, persistent excitation introduces unwanted oscillations into the system.

The central idea in concurrent learning [1]–[3] is to use current measurements concurrently with past measurements, which are stored in history stacks. The repetitive use of past measurements bears a major advantage: PE is replaced by the simpler requirement of linear independence of the recorded data. Furthermore, linearly independent measurements can be obtained in a limited period of time with low or even no external excitation.

A connection between concurrent learning and Recursive Least Squares (RLS) [4, chap. 11] is indicated by [1]. The difference between the two concepts is in the update laws. In RLS, the parameter estimates are updated in discrete steps with every measurement taken. Concurrent learning, on the other hand, utilizes the measurements in the history stacks for a gradient descent. As we will show later, this step allows to evaluate each measurement individually and reject erroneous data quickly.

Concurrent learning-based update laws have been applied in various fields over the last years. In [5], concurrent learning enhances a model predictive control architecture for nonlinear systems. The synchronization of agents with uncertain dynamics in a network is discussed in [6] and [7], where concurrent learning is applied to learn the desired policies in the presence of unknown dynamics. Another promising field of application for concurrent learning-based update laws are switched systems. Both adaptive control of switched systems [8] and adaptive identification of piecewise affine systems [9] benefit from concurrent learning. The use of memory allows to continuously update all subsystems, even those that are currently not active.

Despite growing interest in concurrent learning and its advantages, the use of memory also introduces drawbacks: erroneous data in the history stacks deteriorates the general performance. In [2], it is stated that the tracking error is uniformly ultimately bounded and that the adaptive weights converge to a compact ball around the ideal weights in the presence of erroneous history stacks. While this is a desirable property for small errors, it implies that we converge to the wrong parameters in case of larger errors, which are for instance induced by parameter changes of the system.

Techniques proposed in [8] or [10] remove outdated and therefore erroneous history stack elements by repeatedly replacing the entire history stack. Those techniques however fail to selectively remove a single erroneous element. In this paper, we propose a routine which recognizes and replaces erroneous history stack elements online. The suggested routine exploits the fact that the erroneous stack elements exhibit the largest disagreement in terms of update directions to the current measurement-based term. We validate numerically that 1.) the proposed routine restores tracking properties of concurrent learning and 2.) reduces the parameter error of the obtained estimates.

This paper is organized as follows. In section II, we discuss concurrent learning-based parameter identifiers, which were used to identify piecewise affine systems in [9]. We propose a routine to remove erroneous history stack elements in section III and present arguments to support the algorithm. We validate the algorithm numerically in section IV with two simulation studies. Conclusions follow in section V.

II. Concurrent Learning Adaptive Identification

This section outlines the concurrent learning-based adaptive identification algorithm, which will be further discussed in the subsequent section. The algorithm is motivated by [2] and was previously applied for the identification of PWA systems in [9]. We consider the affine state space system

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

The research leading to these results has received funding partly from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. [267877] and partly from the Technische Universität München - Institute for Advanced Study (www.tum-ias.de), funded by the German Excellence Initiative.

S. Kersting and M. Buss are with the Chair of Automatic Control Engineering and with the TUM Institute for Advanced Study, Technische Universität München, Lichtenbergstrasse 2a, 85748 Garching, Germany {stefan.kersting, mb}@tum.de

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^p$ are the state and input vectors of the system. We wish to recursively identify the unknown system parameters $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times p}$ and $f \in \mathbb{R}^n$ based on measurements of $x(t)$ and $u(t)$. Let us therefore introduce the time-varying estimates $\hat{A}(t) \in \mathbb{R}^{n\times n}$, $\hat{B}(t) \in \mathbb{R}^{n\times p}$ and $\hat{f}(t) \in \mathbb{R}^n$. With carefully designed update laws, these estimates must provably converge to the true parameters.

The update laws in concurrent learning-based adaptive identification (and control) consist of two parts. The first part depends on current measurements of states and control inputs as well as estimation errors. The second part is calculated with past measurements stored in history stacks. The update laws thus make concurrent use of instantaneous and recorded data. We first introduce both parts of the update laws in detail. Afterwards, we discuss their interplay.

A. Update with Current Data

Parameter identifiers [11, Ch. 5] enable the recursive identification of the state space system (1) based on estimation errors. The state of the system is estimated with

$$\dot{\hat{x}} = A_{\mathrm{m}}\hat{x} + \left(\hat{A} - A_{\mathrm{m}}\right)x + \hat{B}u + \hat{f}, \tag{2}$$

where $A_{\mathrm{m}} \in \mathbb{R}^{n\times n}$ is a stable (Hurwitz) design matrix which ensures boundedness of $\hat{x}$. Since A_{m} is stable, we always find a symmetric, positive definite matrix $P \in \mathbb{R}^{n\times n}$ such that $A_{\mathrm{m}}^T P + PA_{\mathrm{m}} = -Q_e$ for a positive definite matrix $Q_e \in \mathbb{R}^{n\times n}$. The following update laws are solely based on current measurements (hence the superscript C):

$$\dot{\hat{A}}^{\mathrm{C}} = -\mathit{\Gamma}_1 Pex^T, \quad \dot{\hat{B}}^{\mathrm{C}} = -\mathit{\Gamma}_2 Peu^T, \quad \dot{\hat{f}}^{\mathrm{C}} = -\mathit{\Gamma}_3 Pe, \tag{3}$$

with estimation error $e = \hat{x} - x$ and scaling constants $\mathit{\Gamma}_1, \mathit{\Gamma}_2, \mathit{\Gamma}_3 \in \mathbb{R}^+$. The update laws (3) share the same limitation as most adaptive algorithms: they require PE. The estimated parameters only converge to the true parameters if the system is persistently excited (see e.g. [11] and [12]).

B. Update with Recorded Data

Concurrent learning improves the performance of adaptive update laws, such as (3), by storing selected measurements in history stacks. The recorded data is then repeatedly employed to update the system parameters in a gradient descent. This increases convergence rates and in the case of switched systems even allows to update inactive subsystem parameters (see [8], [9]).

We record data in form of triplets. By ($x_j \in \mathbb{R}^n$, $u_j \in \mathbb{R}^p$, $\dot{x}_j \in \mathbb{R}^n$), we denote the j-th triplet consisting of state, control input and state derivative, respectively. A total of $q \in \mathbb{N}$ such triplets are recorded and stored in the three history stacks $X \triangleq [x_1, x_2, \ldots, x_q]$, $U \triangleq [u_1, u_2, \ldots, u_q]$ and $\dot{X} \triangleq [\dot{x}_1, \dot{x}_2, \ldots, \dot{x}_q]$. The number of recorded triplets q is upper bounded by practical considerations, i.e. memory limitations only allow a finite number of data points. The lower bound on q, given by $n+p+1 \leq q$, is due to theoretical insights which we discuss shortly.

Note that the stack elements x_j, u_j and $\dot{x}_j$ need not be constant. As a matter of fact, the opposite is the desirable case. History stack elements are flexible and should be replaced appropriately. In [10], it is shown that replacing old data points actually increases the convergence rate if elements are replaced such that the linear independence between history stack elements grows. During all manipulations of the recorded data, the following two assumptions on the history stacks should be fulfilled.

Assumption 1: The history stacks X and U contain elements, such that there exist $n + p + 1$ linearly independent vectors $[x_j^T, u_j^T, 1]^T$.

Assumption 2: Given X and U, all elements $j \in \{1, \ldots, q\}$ of history stack $\dot{X}$ fulfill $Ax_j + Bu_j + f = \dot{x}_j$.

Assumption 1 ensures that the recorded data contains enough information for the identification task. The existence of linearly independent history stack elements for Assumption 1 is guaranteed under PE. Note however that PE is by no means necessary for the collection of the history stack elements.

Assumption 2 requires that the recorded data is correct. In other words, the recorded derivative $\dot{x}_j$ must match the derivative $\dot{x}$ for the corresponding state x_j and input u_j. Note that Assumption 2 is very strict and usually not fulfilled in practice. Particularly in cases where the state derivative cannot be measured, one resorts to techniques such as fixed point smoothing to obtain a sufficiently accurate estimate (see e.g. [13, chap. 9]). We will show in Theorem 1, however, that small errors in the state derivative are acceptable such that estimates of $\dot{x}$ obtained with fixed point smoothing are sufficiently accurate. Nonetheless, the violation of Assumption 2 constitutes the main motivation of this paper as more severe errors can be introduced to the history stacks either due to parameter changes of the system or outliers in the smoothing process. The ability to detect and remove such outliers thus improves the tracking performance and precision of concurrent learning.

Let us now define ε_j as the error between estimated and recorded time derivative of the state for the j-th triplet:

$$\varepsilon_j(t) \triangleq \hat{A}(t)x_j + \hat{B}(t)u_j + \hat{f}(t) - \dot{x}_j. \tag{4}$$

As the history stack elements x_j, u_j and $\dot{x}_j$ in (4) are constant, changes in ε_j must stem from the time-varying estimates $\hat{A}$, $\hat{B}$ and $\hat{f}$. We enhance the update laws in (3) with an additional part which is based on the recorded data (hence the superscript R) and the errors ε_j:

$$\begin{aligned} &\dot{\hat{A}}^{\mathrm{R}} = -\mathit{\Gamma}_1 \sum_{j=1}^{q} \varepsilon_j x_j^T, \qquad \dot{\hat{B}}^{\mathrm{R}} = -\mathit{\Gamma}_2 \sum_{j=1}^{q} \varepsilon_j u_j^T, \\ &\dot{\hat{f}}^{\mathrm{R}} = -\mathit{\Gamma}_3 \sum_{j=1}^{q} \varepsilon_j. \end{aligned} \tag{5}$$

C. Concurrent Learning

For concurrent learning adaptive identification, we combine the current data-based update laws in (3) with the recorded data-based update laws in (5). We obtain the following theorem on the convergence of the parameter estimates.

Theorem 1: Consider the affine system (1) *and let the state of the system be observed by* (2)*. Update the estimates* $\hat{A}$*,* $\hat{B}$ *and* $\hat{f}$ *by*

$$\dot{\hat{A}} = \dot{\hat{A}}^C + \dot{\hat{A}}^R, \qquad \dot{\hat{B}} = \dot{\hat{B}}^C + \dot{\hat{B}}^R, \qquad \dot{\hat{f}} = \dot{\hat{f}}^C + \dot{\hat{f}}^R. \quad (6)$$

- *If Assumptions 1 and 2 hold, the estimates converge exponentially to the true parameters* A*,* B *and* f*.*
- *If only Assumption 1 holds, the estimates converge to a compact ball around the true parameters.*

We state now the essential steps of the proof for the erroneous case in which Assumption 2 is violated. The proof delivers insights which constitute the starting point for the intended detection of erroneous history stack elements.

Proof: We begin by modeling the violation of Assumption 2. We observe a discrepancy between the measured derivative and the true derivative of the state if Assumption 2 is violated. In order to model this, we define the measured/estimated derivative $\hat{\dot{x}}_j$ in our history stack as the true derivative $\dot{x}(x_j, u_j) = Ax_j + Bu_j + f$ at the corresponding state and control input minus an error term $\delta_j \in \mathbb{R}^n$:

$$\hat{\dot{x}}_j \triangleq \dot{x}(x_j, u_j) - \delta_j. \quad (7)$$

We insert (7) in the definition of ε_j in (4). This yields

$$\varepsilon_j = \tilde{A}x_j + \tilde{B}u_j + \tilde{f} + \delta_j, \quad (8)$$

with parameter errors $\tilde{A} = \hat{A} - A$, $\tilde{B} = \hat{B} - B$ and $\tilde{f} = \hat{f} - f$. While (4) specifies how to calculate ε_j online, the expression in (8) – which cannot be calculated due to the unknown parameter errors – is beneficial in the following convergence analysis.

In order to formulate a quadratic Lyapunov function in terms of the parameter errors $\tilde{A}$, $\tilde{B}$ and $\tilde{f}$, we combine them in a single column vector

$$\tilde{\theta} = \left[\mathrm{vec}(\tilde{A})^T, \mathrm{vec}(\tilde{B})^T, \tilde{f}^T\right]^T \in \mathbb{R}^{n(n+p+1)}. \quad (9)$$

With the Kronecker product $\otimes$, we introduce the matrices

$$\Gamma \triangleq \begin{bmatrix} \mathit{\Gamma}_1 I_{nn} & 0 & 0 \\ 0 & \mathit{\Gamma}_2 I_{np} & 0 \\ 0 & 0 & \mathit{\Gamma}_3 I_n \end{bmatrix}, \qquad \Psi \triangleq \begin{bmatrix} x \\ u \\ 1 \end{bmatrix} \otimes I_n,$$
$$\Xi \triangleq \left(\sum_{j=1}^{q} \begin{bmatrix} x_j \\ u_j \\ 1 \end{bmatrix} \begin{bmatrix} x_j^T & u_j^T & 1 \end{bmatrix} \right) \otimes I_n, \quad (10)$$

where I_n, I_{nn} and I_{np} are identity matrices of dimensions $n \times n$, $nn \times nn$ and $np \times np$, respectively. This allows us to express most equations in a more compact form, which in turn simplifies our later analysis. First, we rewrite (8) as

$$\varepsilon_j = \left(\begin{bmatrix} x_j \\ u_j \\ 1 \end{bmatrix} \otimes I_n \right)^T \tilde{\theta} + \delta_j. \quad (11)$$

Moreover, we transform the update laws (6) with the help of (9), (10) and (11) and obtain

$$\dot{\tilde{\theta}} = \dot{\hat{\theta}} = -\Gamma \Psi P e - \Gamma \Xi \tilde{\theta} - \Gamma \underbrace{\left(\sum_{j=1}^{q} \begin{bmatrix} x_j \\ u_j \\ 1 \end{bmatrix} \otimes \delta_j \right)}_{\Delta}. \quad (12)$$

Also, the derivative of the estimation error is now given in terms of $\tilde{\theta}$ by

$$\dot{e} = A_{\mathrm{m}} e + \Psi^T \tilde{\theta}. \quad (13)$$

We choose the quadratic Lyapunov function $V = e^T P e + \tilde{\theta}^T \Gamma^{-1} \tilde{\theta}$ and with (12) and (13) obtain its time derivative:

$$\dot{V} = -e^T Q_e e - \tilde{\theta}^T Q_\theta \tilde{\theta} - 2\Delta^T \tilde{\theta}, \quad (14)$$

where $Q_e \triangleq -(A_{\mathrm{m}}^T P + P A_{\mathrm{m}})$ and $Q_\theta \triangleq \Xi^T + \Xi$. Recall that Q_e was chosen to be positive definite in the design of the update laws (3). Also, the matrix Ξ and in turn also Q_θ are positive definite because of Assumption 1.

Note that (14) also incorporates the ideal case in Theorem 1. If we set $\delta_j = 0, \ \forall j \in \{1, \ldots, q\}$, the Δ-term vanishes and the derivative of our candidate Lyapunov function is indeed negative definite. Hence, the estimation errors e and parameter errors $\tilde{\theta}$ converge to zero exponentially.

In the erroneous case however, we have $\delta_j \neq 0$. Therefore, (14) is quadratic in e and $\tilde{\theta}$ but is also linear in $\tilde{\theta}$. This divides the error space $[e^T \ \ \tilde{\theta}^T]^T \in \mathbb{R}^{n(n+p+2)}$ in two sets. For some small values of $\tilde{\theta}$ the linear term outweighs the quadratic terms, which results in a first closed set for which $\dot{V} > 0$. This first set is surrounded by a second set in which the quadratic terms outweigh the linear term and $\dot{V} < 0$. The outer set with $\dot{V} < 0$ guarantees boundedness of e and $\tilde{\theta}$.

Next, we approximate the compact set to which the parameter errors $\tilde{\theta}$ converge by analyzing the shape of the boundary between the two sets. This boundary is characterized by $\dot{V} = 0$. Rearranging (14) with $\dot{V} = 0$ yields

$$\begin{bmatrix} e^T & \tilde{\theta}^T \end{bmatrix} \begin{bmatrix} Q_e & 0 \\ 0 & Q_\theta \end{bmatrix} \begin{bmatrix} e \\ \tilde{\theta} \end{bmatrix} + \begin{bmatrix} 0 & 2\Delta^T \end{bmatrix} \begin{bmatrix} e \\ \tilde{\theta} \end{bmatrix} = 0. \quad (15)$$

With (15), we easily verify that the origin and the state with $e = 0$ and $\tilde{\theta} = -2(Q_\theta^{-1})^T \Delta$ belong to the boundary set. Furthermore, (15) defines an ellipsoid in the error space which is centered at $e_c = 0$ and $\tilde{\theta}_c = -(Q_\theta^{-1})^T \Delta$. Another representation of the ellipsoid (15) is

$$\begin{bmatrix} e^T - e_c^T & \tilde{\theta}^T - \tilde{\theta}_c^T \end{bmatrix} \underbrace{\frac{1}{\Delta^T Q_\theta^{-1} \Delta} \begin{bmatrix} Q_e & 0 \\ 0 & Q_\theta \end{bmatrix}}_{E} \begin{bmatrix} e - e_c \\ \tilde{\theta} - \tilde{\theta}_c \end{bmatrix} = 1.$$

The eigenvectors of the positive definite matrix E define the principal axes of the ellipsoid. The length of the semi-axes is given by the square root of the inverse eigenvalues of E. The maximal semi-principal axis $\nu_{\max}$ is thus characterized by the minimal eigenvalue $\lambda_{\min}(\cdot)$. Its length is

$$\|\nu_{\max}\| = \left(\lambda_{\min} \left(\frac{Q_\theta}{\Delta^T Q_\theta^{-1} \Delta} \right) \right)^{-\frac{1}{2}}.$$

Figure 1 visualizes the ellipsoid for an exemplary two-dimensional parameter vector $\theta = [\theta_1 \ \ \theta_2]^T$. We exploit the triangle inequality on the centroid $\tilde{\theta}_c$ of the ellipsoid and its maximal semi-principal axis $\nu_{\max}$. This approximates the ball around the true parameters to which our estimates converge in the presence of errors δ_j by

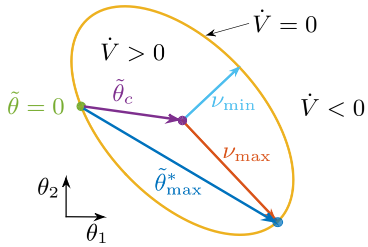

Fig. 1. Visualization of the ellipsoidal set for which $\dot{V} \geq 0$ and approximation of the maximum parameter error $\|\tilde{\theta}_{\max}\|$ in terms of the centroid $\tilde{\theta}_c$ and maximal semi-principal axis $\nu_{\max}$ of the ellipsoid ($\|\nu_{\max}\| = 1/\sqrt{\lambda_{\min}}$).

$\|\tilde{\theta}^*_{\max}\| \leq \|\Delta^T Q_\theta^{-1}\| + \|\nu_{\max}\|$,where the point $\tilde{\theta}^*_{\max}$ is also highlighted in Fig. 1. ■

The proof of Theorem 1 shows that history stack elements may be arbitrarily replaced as long as the two assumptions remain intact. From a control theoretic perspective, the replacement of stack elements amounts to switching between different stable dynamics that share the common Lyapunov function V. Note however that in the erroneous case, changing the history stack elements also changes the ellipsoidal region to which our estimates converge.

The size of the ellipsoidal region is a critical factor when reasoning whether the obtained estimates are sufficiently accurate. Findings in [9] suggest that errors introduced during the estimation of state derivatives usually lead to acceptable levels of accuracy. Single outliers in the smoothing process might however deteriorate the achievable accuracy. It is thus important to obtain good estimates of the state derivatives with $\delta_j \approx 0$. Therefore, obtaining measurements for the history stacks is considered expensive and carried out sparsely. This is a difference compared to RLS, where frequent measurements are needed to update the estimates.

While concurrent learning provides various advantages, such as faster convergence rates or replacing PE by more easily monitored linear independence, it comes at the price of one limitation. With the introduction of memory, the algorithm loses its adaptation capabilities. Assume that the history stack contains measurements taken when the system was accurately modeled by the parameter configuration θ'. Then, the system abruptly switches or gradually changes its parameters to θ'', which renders the history stack elements outdated. In that case, Assumption 2 is violated and the obtained parameter estimates veer away from the true parameters. In order to restore the adaptive capabilities of concurrent learning-based algorithms, we need mechanisms to identify errors in the history stacks online. We propose a solution to this problem in the next section.

III. Detecting Erroneous History Stack Elements

In this section, we present the main contribution of this paper. It is based on the analysis done in the previous section: if the history stacks contain erroneous elements, then our estimates converge to an ellipsoidal set in the parameter space (depicted in Fig. 1). Next, we provide a routine which detects and removes erroneous stack elements online.

We know that our estimates are wrong if $e \neq 0$ (or greater than some threshold in the presence of noise). We then detect erroneous stack elements by comparing the individual contributions in the update laws (6). Let us refer to the contribution of the current measurements and of the j-th history stack triplet by

$$\dot{\hat{\theta}}^C(x(t), u(t)) = -\Gamma \Psi P e \tag{16}$$

$$\dot{\hat{\theta}}^R_j(x_j, u_j, \dot{x}_j) = -\Gamma \Xi_j \tilde{\theta} - \Gamma \Delta_j, \quad j \in \{1, \ldots, q\}, \tag{17}$$

where

$$\Xi_j = \left(\begin{bmatrix} x_j \\ u_j \\ 1 \end{bmatrix} \begin{bmatrix} x_j^T & u_j^T & 1 \end{bmatrix} \right) \otimes I_n, \quad \Delta_j = \begin{bmatrix} x_j \\ u_j \\ 1 \end{bmatrix} \otimes \delta_j .$$

We are interested in the directional information in the update vectors (16) and (17). Recall that under the assumption of persistent excitation, the update law (16) with current data alone drives the parameter estimates to the true parameters. Therefore, $\dot{\hat{\theta}}^C$ serves as a reference point.

Note that the requirement of PE does not constitute a limitation at this point. Once we know that the history stack contains erroneous elements, we should in any case excite the system in order to obtain new measurements to fulfill Assumption 1. While collecting new measurements, we exploit the excitation to reason which elements are erroneous and need to be replaced. While PE guarantees that we obtain measurements satisfying Assumption 1, it is not a necessary condition. Future work could analyze how the following ideas perform without the PE condition. As shown in [12], we ensure PE by choosing uncorrelated input signals sufficiently rich of order $n + 1$.

We analyze to which extent the memory-based update laws (17) coincide with this reference vector $\dot{\hat{\theta}}^C$. We define for each stack element j the angle φ_j between $\dot{\hat{\theta}}^C$ and $\dot{\hat{\theta}}^R_j$:

$$\varphi_j(t) = \sphericalangle \left(\dot{\hat{\theta}}^C(t),\ \dot{\hat{\theta}}^R_j(t) \right) . \tag{18}$$

Furthermore, note that the angles φ_j only allow for meaningful statements if the corresponding vectors are sufficiently long. We introduce the two thresholds ϑ^C and ϑ^R for $\dot{\hat{\theta}}^C$ and $\dot{\hat{\theta}}^R$, respectively. If $\|\dot{\hat{\theta}}^C\| < \vartheta^C$, then the current data-based update vector fails to qualify as a reliable reference point. This is usually the case if our parameter estimates are in the vicinity of the true parameters. If $\|\dot{\hat{\theta}}^R_j\| < \vartheta^R$, then the contribution of the j-th history stack element is very small. This is for instance the case if the current estimates $\hat{\theta}$ can not be falsified by the j-th history stack triplet $(x_j, u_j, \dot{x}_j)$. Therefore, falling below the two thresholds corresponds to good estimates. We thus set the angle φ_j to zero whenever falling below any of the two thresholds:

$$\varphi_j(t) \Leftarrow 0, \quad \text{if } \left(\|\dot{\hat{\theta}}^C\| < \vartheta^C \text{ or } \|\dot{\hat{\theta}}^R_j\| < \vartheta^R \right) . \tag{19}$$

The parameter estimates and angles φ_j are time-varying due to the excitation u. With a sinusoidal excitation in u, the parameter estimates and angles are forced into a limit

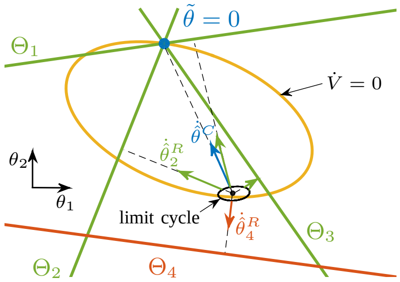

Fig. 2. Exemplary limit cycle in a two dimensional parameter space. History stack has 4 elements for which only $\delta_4 \neq 0$. The update vectors $\dot{\hat{\theta}}_j^R$ are an orthogonal projection onto Θ_j.

cycle. As we assume knowledge of the excitation u, we can approximate the frequency of this limit cycle. We are only interested in the update directions on average and therefore consider the low-pass filtered angles $\bar{\varphi}_j$, where the cut-off frequency of the filter is lower than the lowest excitation frequency in u. With the filtered angles $\bar{\varphi}_j$, we propose the following routine to detect and remove erroneous history stack elements.

Algorithm 1: Consider the concurrent learning-based update law in Theorem 1 with errors δ_j. Monitor the filtered angles $\bar{\varphi}_j$ according to (18) *and* (19)*. Whenever the parameter estimates enter a limit cycle (i.e. $e \neq 0$ and $\dot{\bar{\varphi}}_j \approx 0$), replace the k-th history stack triplet, where*

$$k = \arg\max_j \left\{ \bar{\varphi}_j \middle| \bar{\varphi}_j \geq \frac{\pi}{2} \right\}, \tag{20}$$

by a new measurement with $\bar{\varphi}_{new} < \bar{\varphi}_k$ in order to reduce the number and/ or magnitude of errors in the history stack.

Our Algorithm 1 is further justified with Fig. 2. First, note that each triplet in the history stack spans a subspace in the parameter space. The subspace Θ_j associated with the j-th triplet contains all parameter configurations that cannot be falsified by the measurements $(x_j, u_j, \dot{x}_j)$:

$$\Theta_j = \left\{ \hat{\theta} \mid \hat{A}(\hat{\theta})x_j + \hat{B}(\hat{\theta})u_j + \hat{f}(\hat{\theta}) = \dot{x}_j \right\}. \tag{21}$$

The parameter configurations in Θ_j form the solution of an underdetermined system of linear equations (21). From this system we derive the dimension of Θ_j to be $\dim(\Theta_j) = \dim(\theta) - n = n(n+p)$.

In the errorless case ($\delta_j = 0, \forall j$) and under Assumption 1, all subspaces intersect at a single point θ, where $\tilde{\theta} = 0$. In case of errors ($\delta_j \neq 0$), the subspaces most likely do not intersect at this common point. In Fig. 2, the subspaces of three correct history stack triplets are displayed as green lines. The red line is shifted away from the intersection point due to errors in the state derivative of the corresponding history stack element. Due to this error, the estimates do not converge to the true parameter configuration θ but enter a limit cycle in the ellipsoidal set discussed in the proof of Theorem 1.

Now note that the recorded data-based update laws (17) correspond to an orthogonal projection onto the subspaces Θ_j. For the correct history stack triplet, this projection points into the ellipsoid. Also the current data-based update (16) points towards the true parameter configuration on average. Wrong parameter estiamtes occure if there is at least one erroneous stack element pointing out of the ellipsoid. It follows that the averaged angle of at least one erroneous history stack element is greater than $\pi/2$. This insight affirms Algorithm 1.

It is left for future work to analyze how Algorithm 1 performs in case the system is not persistently excited.

IV. Numerical Validation

In this section, we numerically validate Algorithm 1 for the detection of erroneous history stack elements in two settings. First, we show that Algorithm 1 allows concurrent learning to track parameter changes in the system. Afterwards, we compare the proposed method with RLS and with an alternative approach to enable tracking in concurrent learning [8].

A. Tracking Parameter Changes

We consider an exemplary switched affine system ($n = 2$, $p = 1$) which changes its parameters at $t = 1000\,\mathrm{s}$ from $\theta' = \begin{bmatrix} 0 & -2 & 1 & -3 & 0 & 1 & 0 & 0 \end{bmatrix}^T$ to $\theta'' = \begin{bmatrix} 0 & -5 & 1 & -1 & 0 & 3 & 0 & -0.2 \end{bmatrix}^T$.

We initialize the history stacks of size $q = 4$ with measurements of x_j, u_j and erroneous state derivatives $\dot{x}_j$ with normally distributed $\delta_j \sim \mathcal{N}(0, 0.5)$. We choose the design parameters $A_\mathrm{m} = -I_2$, $\Gamma = I_8$, $P = I_2$, $\vartheta^C = 0.05$ and $\vartheta^R(t) = 0.5 \max_j(\|\dot{\hat{\theta}}_j^R(t)\|)$ and excite the system with $u(t) = 0.5(\sin(\frac{5}{3}t) + \cos(t))$. During the simulation we generate estimates $\dot{x}_j$ with normally distributed errors $\delta_j \sim \mathcal{N}(0, 0.01)$. Furthermore, we add normally distributed noise with zero mean and a variance of 0.1 to the state measurements $x(t)$ in (2) and (3). Figure 3 shows the simulation results.

Within the first $100\,\mathrm{s}$, the proposed algorithm successfully replaces the initial history stack elements by new measurements with smaller variance in δ_j. Our estimates $\hat{\theta}$ converge to the true parameters θ'. The system parameters change to θ'' at time instance $t = 1000\,\mathrm{s}$ and the system consequently enters a limit cycle. This can be seen in Fig. 3 by strongly oscillating parameter estimates. These oscillations naturally trigger the proposed algorithm, which then replaces outdated history stack elements. After approximately $300\,\mathrm{s}$, the history stacks do not contain any outdated measurements taken with θ' and our estimates converge to the true parameters θ''.

B. Comparison with Cyclic Replacement and RLS

Note that the ability to reduce the parameter error $\tilde{\theta}$ in the presence of errors δ_j constitutes a major advantage of the proposed algorithm over other approaches. We show this for a stable, randomly chosen system of higher order ($n = 5, p = 2$). We excite the system with the input signals $u_1(t) = \sin(\frac{5}{3}t) + \cos(1.1t) + \sin(1.7t)$ and $u_2(t) = \sin(\frac{4}{3}t) + \cos(t) + \sin(\frac{7}{3}t)$. We initialize the history stacks with $q = 10$ measurements and draw the errors in the history stacks from a normal distribution $\delta_j \sim \mathcal{N}(0, 0.3)$. During a period of $2000\,\mathrm{s}$, the detection mechanism according to

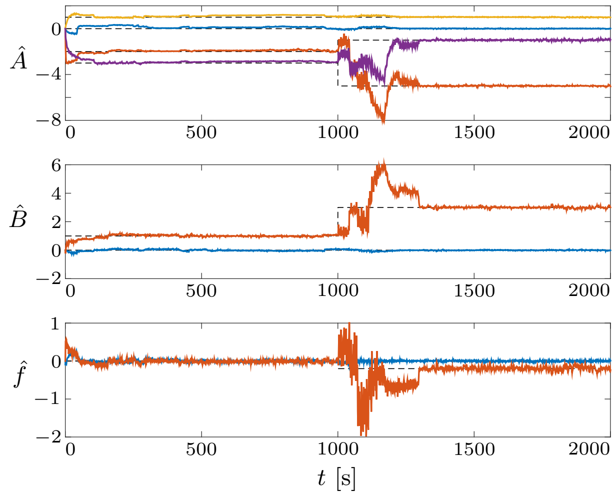

Fig. 3. Tracking a parameter change at $t = 1000\,\mathrm{s}$ with the proposed detection mechanism for erroneous stack elements in concurrent learning. Solid lines represent the parameter estimates and dotted lines indicate the true parameters.

Algorithm 1 replaces the stack elements with the greatest errors by new measurements. Note that the new elements also contain errors δ_j which follow the same normal distribution as the initial errors. The chosen design parameters are the same as in section IV-A.

We compare our approach with the cyclic replacement approach proposed in [8] and RLS [4, chap. 11]. Following [8], we model the aging of data with an exponential decay ($e^{-0.01t}$) on the current history stack. Once a new history stack outperforms the current history stack, the entire history stack is replaced. RLS was implemented on the history stack measurements x_j, u_j and $\dot{x}_j$ with a forgetting factor $\gamma = 0.95$.

Figure 4 shows the estimation process for all three approaches. The parameter error for the cyclic replacement approach is greater than the error obtained with our error detection approach. This is due to the fact that the entire history stack is replaced at once. In that case, it is very likely that the history stack contains elements with similarly large errors. Even though RLS performs slightly better than concurrent learning with cyclic replacement, it is still affected by the errors δ_j. The proposed algorithm on the other hand selectively replaces the most erroneous history stack element. After about $700\,\mathrm{s}$, the algorithm has found history stack elements with sufficiently small errors which cause the update terms to fall below the thresholds ϑ^C and ϑ^R in (19). This terminates our detection algorithm and the history stack is left unchanged.

V. Conclusions

This paper is concerned with concurrent learning-based adaptive identification: a framework where instantaneous data and history stack elements are concurrently used to update estimates of the unknown system parameters. The introduction of memory however compromises the tracking ability which is essential for adaptive algorithms. Furthermore, erroneous stack elements restrict convergence

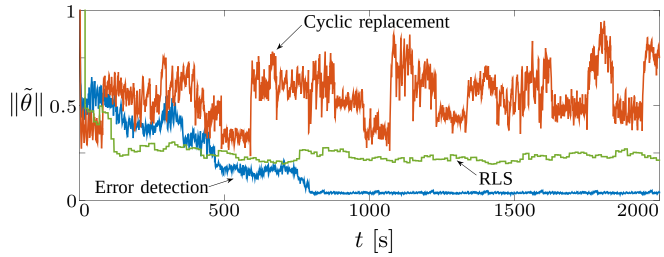

Fig. 4. Comparison of the proposed detection mechanism for erroneous stack elements (blue) with a cyclic replacement of the entire history stack (red) and Recursive Least Squares (green).

to a compact ball around the true parameters. This paper shows how to selectively detect erroneous or outdated history stack elements at runtime. The proposed routine utilizes the insight that the update directions of erroneous history stack elements exhibit the largest disagreement to the traditional update laws. Replacing the corresponding elements restores the tracking ability of concurrent learning and provides better estimates. Future work is devoted to the formulation of guidelines for the choice of the design parameters and the evaluation of the proposed detection mechanism for adaptive control systems with concurrent learning.

References

[1] G. Chowdhary and E. Johnson, “Concurrent learning for convergence in adaptive control without persistency of excitation,” in *Proc. of the 49th IEEE CDC*, 2010, pp. 3674–3679.

[2] 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*, vol. 27, pp. 280–301, 2013.

[3] G. Chowdhary, “Concurrent Learning for convergence in adaptive control without persistency of excitation,” Ph.D. dissertation, Georgia Institute of Technology, 2010.

[4] L. Ljung, *System identification: Theory for the user*. Englewood Cliffs, NJ: Prentice-Hall, 1987.

[5] G. Chowdhary, M. Mühlegg, J. How, and F. Holzapfel, “A Concurrent Learning Adaptive-Optimal Control Architecture for Nonlinear Systems,” in *Proc. of the 52nd IEEE CDC*, 2013, pp. 868–873.

[6] R. Kamalapurkar, P. Walters, and W. Dixon, “Concurrent learning-based approximate optimal regulation,” in *Proc. of the 52nd IEEE CDC*, 2013, pp. 6256–6261.

[7] J. Klotz, R. Kamalapurkar, and W. E. Dixon, “Concurrent learning-based network synchronization,” in *Proc. of the American Control Conference*, 2014, pp. 796–801.

[8] G. De La Torre, G. Chowdhary, and E. N. Johnson, “Concurrent learning adaptive control for linear switched systems,” in *Proc. of the American Control Conference*, 2013, pp. 854–859.

[9] S. Kersting and M. Buss, “Concurrent learning adaptive identification of piecewise affine systems,” in *Proc. of the 53rd IEEE CDC*, 2014, pp. 3930–3935.

[10] G. Chowdhary and E. Johnson, “A singular value maximizing data recording algorithm for concurrent learning,” in *Proc. of the American Control Conference*, 2011, pp. 3547–3552.

[11] P. Ioannou and J. Sun, *Robust adaptive control*. Upper Saddle River, NJ: Prentice-Hall, 1996.

[12] S. Kersting and M. Buss, “Online identification of piecewise affine systems,” in *UKACC 10th International Conference on (CONTROL 2014)*, 2014, pp. 90–95.

[13] D. Simon, *Optimal state estimation: Kalman, H [infinity] and nonlinear approaches*. Hoboken, N.J: Wiley-Interscience, 2006.