# Integral Concurrent Learning: Adaptive Control with Parameter Convergence using Finite Excitation

1

Anup Parikh<sup>1</sup>, Rushikesh Kamalapurkar<sup>2</sup> and Warren E. Dixon<sup>3</sup>

# Abstract

Concurrent learning is a recently developed adaptive update scheme that can be used to guarantee parameter convergence without requiring persistent excitation. However, this technique requires knowledge of state derivatives, which are usually not directly sensed and therefore must be estimated. A novel integral concurrent learning method is developed in this paper that removes the need to estimate state derivatives while maintaining parameter convergence properties. Data recorded online is exploited in the adaptive update law, and numerical integration is used to circumvent the need for state derivatives. The novel adaptive update law results in negative definite parameter error terms in the Lyapunov analysis, provided an online-verifiable finite excitation condition is satisfied. A Monte Carlo simulation illustrates improved robustness to noise compared to the traditional derivative formulation. The result is also extended to Euler-Lagrange systems, and simulations on a two-link planar robot demonstrate the improved performance compared to gradient based adaptation laws.

#### **Index Terms**

Adaptive Control, System Identification, Nonlinear Systems, Uncertain Systems

#### I. INTRODUCTION

Adaptive control methods provide a technique to achieve a control objective despite uncertainties in the system model. Adaptive estimates are developed through insights from a Lyapunov-based analysis as a means to yield a desired objective. Although a regulation or tracking objective can be achieved with this scheme, it is well known that the parameter estimates may not approach the true parameters using a least-squares or a gradient based online update law without persistent excitation (PE) [1]–[3]. However, the PE condition cannot be guaranteed *a priori* for nonlinear systems, and is difficult to check online, in general.

Motivated by the desire to learn the true parameters, or at least to gain the increased robustness and improved transient performance that parameter convergence provides (see [4]–[6]), a new adaptive update scheme known as concurrent learning (CL) was recently developed in the pioneering work of [6]–[8]. The principle idea of CL is to use recorded input and output data of the system dynamics to apply batch-like updates to the parameter estimate dynamics. These updates yield a negative definite, parameter estimation error term in the stability analysis, which allows parameter convergence to be established provided a finite excitation condition is satisfied. The finite excitation condition is an alternative condition compared to PE, and only requires excitation for a finite amount of time. Furthermore, the condition can be checked online by verifying the positivity of the minimum singular value of a function of the regressor matrix, as opposed to PE, which cannot be verified online, in general, for nonlinear systems. However, all current CL methods require that the output data include the state derivatives, which may not be available for all systems. Since the naive approach of finite difference of the state measurements leads to noise amplification, and since only past recorded data, opposed to real-time data, is needed for CL, techniques such as online state derivative estimation or smoothing have been employed, e.g., [9], [10]. However, these methods typically require tuning parameters such as an observer gain, switching threshold, etc. in the case of the online derivative estimator, and basis, basis order, covariance, time window, etc. in the case of smoothing, to produce satisfactory results.

In this note, we reformulate the CL method in terms of an integral, removing the need to estimate state derivatives. Other methods such as composite adaptive control also use integration based terms to improve parameter convergence (e.g., [11]–[13]), however they still require PE to ensure exponential convergence. Recently, results such as [14]–[17] have shown convergence using an interval or finite excitation condition, though they either require measurements of state derivatives (i.e., [15]), require determining the analytical Jacobian of the regressor (i.e., [14]) or are developed in a model reference adaptive control context (i.e., [16]–[20]), rather than the general nonlinear systems considered here. Results such as [21]–[26] have theoretical analogues to those presented here, and some use filtering techniques to avoid data storage requirements, though we show how the parameter estimation is performed alongside control development in a Lyapunov analysis. In our method, the only additional tuning parameter beyond what is needed for gradient-based adaptive control designs is the time window of integration, which is

<sup>&</sup>lt;sup>1</sup>Robotics Research & Development, Sandia National Laboratories, Albuquerque NM 87123, USA Email:aparikh@sandia.gov

<sup>&</sup>lt;sup>2</sup>School of Mechanical and Aerospace Engineering, Oklahoma State University, Stillwater OK 74078, USA Email:rushikesh.kamalapurkar@okstate.edu

<sup>&</sup>lt;sup>3</sup>Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville FL 32611-6250, USA Email:,wdixon@ufl.edu

This research is supported in part by a Task Order contract with the Air Force Research Laboratory, Munitions Directorate at Eglin AFB, and Office of Naval Research Grant N00014-13-1-0151. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the sponsoring agency.

analogous to the smoothing buffer window that is already required for smoothing based techniques. Despite the reformulation, the stability results still hold (i.e., parameter convergence) and Monte Carlo simulation results suggest greater robustness to noise compared to derivative based CL implementations. The technique is also applied to Euler-Lagrange (EL) systems to demonstrate the use of ICL for systems with unmatched uncertainties. Compared to other similar approaches applied to EL systems (e.g., [27]), our approach selectively collects data rather than incorporating the entire history so as to consider the most informative data for parameter estimation.

#### II. CONTROL OBJECTIVE

To illustrate the integral CL method, consider an example dynamic system modeled as

$$\dot{x}(t) = f(x(t), t) + u(t)$$
 (1)

where  $t \in [0, \infty)$ ,  $x : [0, \infty) \to \mathbb{R}^n$  are the measurable states,  $u : [0, \infty) \to \mathbb{R}^n$  is the control input and  $f : \mathbb{R}^n \times [0, \infty) \to \mathbb{R}^n$  represents the locally Lipschitz drift dynamics, with some unknown parameters. In the following development, as is typical in adaptive control, f is assumed to be linearly parametrized in the unknown parameters, i.e.,

$$f(x,t) = Y(x,t)\theta$$
<sup>(2)</sup>

where  $Y : \mathbb{R}^n \times [0, \infty) \to \mathbb{R}^{n \times m}$  is a regressor matrix and  $\theta \in \mathbb{R}^m$  represents the constant, unknown system parameters. To quantify the state tracking and parameter estimation objective of the adaptive control problem, the tracking error and parameter estimate error are defined as

$$e(t) \stackrel{\Delta}{=} x(t) - x_d(t) \tag{3}$$

$$\theta\left(t\right) \triangleq \theta - \theta\left(t\right) \tag{4}$$

where  $x_d : [0, \infty) \to \mathbb{R}^n$  is a known, continuously differentiable desired trajectory and  $\hat{\theta} : [0, \infty) \to \mathbb{R}^m$  is the parameter estimate. In the following, functional arguments will be omitted for notational brevity, e.g., x(t) will be denoted as x, unless necessary for clarity.

To achieve the control objective, the following controller is commonly used:

$$u(t) \triangleq \dot{x}_d - Y(x,t)\,\hat{\theta} - Ke\tag{5}$$

where  $K \in \mathbb{R}^{n \times n}$  is a positive definite constant control gain. Taking the time derivative of (3) and substituting for (1), (2), and (5), yields the closed loop error dynamics

$$\dot{e} = Y(x,t)\theta + \dot{x}_d - Y(x,t)\hat{\theta} - Ke - \dot{x}_d$$
  
=  $Y(x,t)\tilde{\theta} - Ke$  (6)

The parameter estimation error dynamics are determined by taking the time derivative of (4), yielding

$$\dot{\tilde{\theta}}(t) = -\dot{\tilde{\theta}}.$$
(7)

An integral CL-based update law for the parameter estimate is designed as

$$\dot{\hat{\theta}}(t) \triangleq \Gamma Y(x,t)^{T} e + k_{CL} \Gamma \sum_{i=1}^{N} \mathcal{Y}_{i}^{T} \left( x(t_{i}) - x(t_{i} - \Delta t) - \mathcal{U}_{i} - \mathcal{Y}_{i} \hat{\theta} \right)$$

$$\tag{8}$$

where  $k_{CL} \in \mathbb{R}$  and  $\Gamma \in \mathbb{R}^{m \times m}$  are constant, positive definite control gains,  $N \in \mathbb{Z}^+$  is a positive constant that satisfies  $N \geq \left\lceil \frac{m}{n} \right\rceil$ ,  $t_i \in [0, t]$  are time points between the initial time and the current time,  $\mathcal{Y}_i \triangleq \mathcal{Y}(t_i)$ ,  $\mathcal{U}_i \triangleq \mathcal{U}(t_i)$ ,

$$\mathcal{Y}(t) \triangleq \int_{\max\{t-\Delta t, 0\}}^{t} Y(x(\tau), \tau) d\tau,$$
(9)

$$\mathcal{U}(t) \triangleq \int_{\max\{t-\Delta t, 0\}}^{t} u(\tau) d\tau, \tag{10}$$

 $0_{n \times m}$  denotes an  $n \times m$  matrix of zeros, and  $\Delta t \in \mathbb{R}$  is a positive constant denoting the size of the window of integration. The concurrent learning term (i.e., the second term) in (8) represents saved data. The principal idea behind this design is to utilize recorded input-output data generated by the dynamics to further improve the parameter estimate. See [7] for a discussion on how to choose data points to record. In short, the data points should be selected to maximize the minimum eigenvalue of  $\sum_{i=1}^{N} \mathcal{Y}_i^T \mathcal{Y}_i$  since the minimum eigenvalue bounds the rate of convergence of the parameter estimation errors, as shown in the subsequent stability analysis. To calculate  $\mathcal{Y}(t)$  and  $\mathcal{U}(t)$ , one would store the values of Y and u over the interval  $[t - \Delta t, t]$ , which would require  $[mnhb\Delta t]$  and  $[nhb\Delta t]$  bytes, respectively, where h is the control loop rate in cycles per second and b

is the number of bytes per value (e.g., 8 bytes per double precision floating point number). Often, these storage requirements are easily satisfied by even modern embedded systems with somewhat limited memory.

The integral CL-based adaptive update law in (8) differs from traditional state derivative based CL update laws given in, e.g., [6]–[8]. Specifically, the state derivative, control, and regressor terms, i.e.,  $\dot{x}$ , u, and Y, respectively, used in [6]–[8] are replaced with the integral of those terms over the time window  $[t - \Delta t, t]$ .

Substituting (2) into (1), and integrating yields

$$\int_{t-\Delta t}^{t} \dot{x}(\tau) \, d\tau = \int_{t-\Delta t}^{t} Y(x,\tau) \, \theta d\tau + \int_{t-\Delta t}^{t} u(\tau) \, d\tau,$$

 $\forall t > \Delta t$ . Using the Fundamental Theorem of Calculus and the definitions in (9) and (10),

$$x(t) - x(t - \Delta t) = \mathcal{Y}(t)\theta + \mathcal{U}(t)$$
(11)

 $\forall t > \Delta t$ , where the fact that  $\theta$  is a constant was used to pull it outside the integral. Rearranging (11) and substituting into (8) yields

$$\dot{\hat{\theta}}(t) = \Gamma Y(x,t)^T e + k_{CL} \Gamma \sum_{i=1}^N \mathcal{Y}_i^T \mathcal{Y}_i \tilde{\theta}, \quad \forall t > \Delta t.$$
(12)

Note that (9) and (10) are piecewise continuous in time, the concurrent learning term in (8) is piecewise constant in time, and the simplified adaptive update law (12) is piecewise continuous in time. Hence, the right hand side of (7) is piecewise continuous in time.

#### **III. STABILITY ANALYSIS**

To facilitate the following analysis, let  $\eta : [0, \infty) \to \mathbb{R}^{n+m}$  represent a composite vector of the system states and parameter estimation errors, defined as  $\eta(t) \triangleq \begin{bmatrix} e^T & \tilde{\theta}^T \end{bmatrix}^T$ . Also, let  $\lambda_{\min} \{\cdot\}$  and  $\lambda_{\max} \{\cdot\}$  represents the minimum and maximum eigenvalues of  $\{\cdot\}$ , respectively.

In the following stability analysis, time is partitioned into two phases. During the initial phase, insufficient data has been collected to satisfy a richness condition on the history stack. In Theorem 1, it is shown that the controller and adaptive update law are still sufficient for the system to remain bounded for all time despite the lack of data. After a finite period of time, the system transitions to the second phase, where the history stack is sufficiently rich and the controller and adaptive update law are shown, in Theorem 2, to exponentially converge. To guarantee that the transition to the second phase happens in finite time, and therefore the overall system trajectories are ultimately bounded, we require the history stack be sufficiently rich after a finite period of time, as specified in the following assumption.

Assumption 1. The system is sufficiently excited over a finite duration of time. Specifically,  $\exists \underline{\lambda} > 0$ ,  $\exists T > \Delta t : \forall t \ge T$ ,  $\lambda_{\min} \left\{ \sum_{i=1}^{N} \mathcal{Y}_{i}^{T} \mathcal{Y}_{i} \right\} \ge \underline{\lambda}$ .

The condition in (1) requires that the system be sufficiently excited, though is weaker than the typical PE condition since excitation is only needed for a finite period of time. Specifically, PE requires

$$\int_{t}^{t+\Delta t} Y^{T}\left(x\left(\tau\right),\tau\right)Y\left(x\left(\tau\right),\tau\right)d\tau \ge \alpha I > 0, \ \forall t > 0$$
(13)

whereas Assumption 1 only requires the system trajectories to be exciting up to time T (at which point  $\sum_{i=1}^{N} \mathcal{Y}_{i}^{T} \mathcal{Y}_{i}$  is full rank), after which the exciting data recorded during  $t \in [0, T]$  is exploited for all t > T. Another benefit of the development in this paper is that the excitation condition is measurable (i.e.,  $\lambda_{\min} \left\{ \sum_{i=1}^{N} \mathcal{Y}_{i}^{T} \mathcal{Y}_{i} \right\}$  can be calculated), whereas in PE,  $\Delta t$  is unknown, and hence an uncountable number of integrals would need to be calculated at each of the uncountable number of time points, t, in order to verify PE. Assumption 1 is verified online by continually acquiring data (using e.g., the singular value maximization algorithm in [7] to ensure the minimum eigenvalue of  $\sum_{i=1}^{N} \mathcal{Y}_{i}^{T} \mathcal{Y}_{i}$  is always increasing) until  $\lambda_{\min} \left\{ \sum_{i=1}^{N} \mathcal{Y}_{i}^{T} \mathcal{Y}_{i} \right\}$  has reached a user selectable threshold. The threshold value is directly related to the exponential convergence rate of the system, as shown in the subsequent analysis. Since numerical integration may result in truncation errors (e.g., fourth order Runge-Kutta methods have  $\mathcal{O}(h^{5})$  local truncation errors), the threshold should also be selected sufficiently large to ensure the excitation condition is satisfied beyond the bounds of integration uncertainty to mitigate misidentification due to noise and truncation errors.

To encourage excitation of the system, a perturbation signal can be added to the desired trajectory. Notably, this perturbation signal (which distracts from the original state trajectory objective encoded in the original desired trajectory) would only need to be added to the system for a finite time before ensuring that sufficient data has been collected to learn the parameters. In other words, during implementation, the system only needs excitation initially, and then the original desired trajectory can

be tracked. In contrast, adaptive methods relying on PE might require perturbations for all time to ensure parameter estimate convergence, and hence the original state trajectory objective may never be achieved.

**Theorem 1.** For the system defined in (1) and (7), the controller and adaptive update law defined in (5) and (8) ensures bounded tracking and parameter estimation errors.

*Proof:* Let  $V : \mathbb{R}^{n+m} \to \mathbb{R}$  be a candidate Lyapunov function defined as

$$V(\eta) = \frac{1}{2}e^{T}e + \frac{1}{2}\tilde{\theta}^{T}\Gamma^{-1}\tilde{\theta}.$$
(14)

Taking the derivative of V along the trajectories of (1), substituting the closed loop error dynamics in (6) and the equivalent adaptive update law in (12), noting that  $\sum_{i=1}^{N} \mathcal{Y}_{i}^{T} \mathcal{Y}_{i}$  is positive semidefinite, and simplifying yields

$$\dot{V} \leq -e^T K e$$

which implies the system states remain bounded via [28, Theorem 8.4]. Further, since  $\dot{V} \leq 0$ ,  $V(\eta(t)) \leq V(\eta(0))$  and therefore  $\|\eta(t)\| \leq \sqrt{\frac{\beta_2}{\beta_1}} \|\eta(0)\|$ , where  $\beta_1 \triangleq \frac{1}{2} \min\{1, \lambda_{\min}\{\Gamma^{-1}\}\}$  and  $\beta_2 \triangleq \frac{1}{2} \max\{1, \lambda_{\max}\{\Gamma^{-1}\}\}$ .

**Theorem 2.** Under Assumption 1, the controller and adaptive update law defined in (5) and (8) ensures globally exponential tracking of the system defined in (1) and (7) in the sense that

$$\|\eta(t)\| \le \left(\frac{\beta_2}{\beta_1}\right) \exp\left(\lambda_1 T\right) \|\eta(0)\| \exp\left(-\lambda_1 t\right), \ \forall t \in [0,\infty).$$

$$(15)$$

*Proof:* Let  $V : \mathbb{R}^{n+m} \to \mathbb{R}$  be a candidate Lyapunov function defined as

$$V(\eta) = \frac{1}{2}e^{T}e + \frac{1}{2}\tilde{\theta}^{T}\Gamma^{-1}\tilde{\theta}.$$

Taking the derivative of V along the trajectories of (1) during  $t \in [T, \infty)$ , substituting the closed loop error dynamics in (6) and the equivalent adaptive update law in (12), and simplifying yields

$$\dot{V} = -e^T K e - k_{CL} \tilde{\theta}^T \sum_{i=1}^N \mathcal{Y}_i^T \mathcal{Y}_i \tilde{\theta}, \ \forall t \in [T, \infty).$$

From Assumption 1,  $\lambda_{\min}\left\{\sum_{i=1}^{N} \mathcal{Y}_{i}^{T} \mathcal{Y}_{i}\right\} > 0$ ,  $\forall t \in [T, \infty)$ , which implies that  $\sum_{i=1}^{N} \mathcal{Y}_{i}^{T} \mathcal{Y}_{i}$  is positive definite and therefore  $\dot{V}$  is upper bounded by a negative definite function of  $\eta$ . Invoking [28, Theorem 4.10], e and  $\tilde{\theta}$  are globally exponentially stable, i.e.,  $\forall t \in [T, \infty)$ ,

$$\left|\eta\left(t\right)\right\| \leq \sqrt{\frac{\beta_{2}}{\beta_{1}}} \left\|\eta\left(T\right)\right\| \exp\left(-\lambda_{1}\left(t-T\right)\right)$$

where  $\lambda_1 \triangleq \frac{1}{2\beta_2} \min \{\lambda_{\min} \{K\}, k_{CL} \lambda\}$ . The composite state vector can be further upper bounded using the results of Theorem 1, yielding (15).

*Remark* 1. Using an appropriate data selection algorithm (e.g. the singular value maximization algorithm in [7]) ensures the minimum eigenvalue of  $\sum_{i=1}^{N} \mathcal{Y}_{i}^{T} \mathcal{Y}_{i}$  is always increasing, and therefore the Lyapunov function (14) is a common Lyapunov function [29] as data is continuously added to the history stack.

## IV. EXTENSION TO EULER-LAGRANGE SYSTEMS

The ICL technique can also be applied to systems with unmatched uncertainties. In this section, the ICL method is applied to Euler-Lagrange systems.

# A. Control Development

Consider Euler-Lagrange dynamics of the form [30, Chapter 2.3], [31, Chapter 9.3]

$$M(q(t))\ddot{q}(t) + V_m(q(t), \dot{q}(t))\dot{q}(t) + F_d\dot{q}(t) + G(q(t)) = \tau(t)$$
(16)

where q(t),  $\dot{q}(t)$ ,  $\ddot{q}(t) \in \mathbb{R}^n$  represent position, velocity and acceleration vectors, respectively,  $M : \mathbb{R}^n \to \mathbb{R}^{n \times n}$  represents the inertial matrix,  $V_m : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n \times n}$  represents centripetal-Coriolis effects,  $F_d \in \mathbb{R}^{n \times n}$  represents frictional effects,  $G : \mathbb{R}^n \to \mathbb{R}^n$  represents gravitational effects and  $\tau(t) \in \mathbb{R}^n$  denotes the control input. The system in (16) is assumed to have the following properties (see [30, Chapter 2.3]), which hold for a large class of physical systems. **Property 1.** The system in (16) can be linearly parameterized, i.e., the left hand side of (16) can be rewritten as

$$Y_{1}(q, \dot{q}, \ddot{q})\theta = M(q)\ddot{q} + V_{m}(q, \dot{q})\dot{q} + F_{d}\dot{q} + G(q)$$
(17)

where  $Y_1 : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n \times m}$  denotes the regression matrix, and  $\theta \in \mathbb{R}^m$  is a vector of uncertain parameters.

**Property 2.** The inertia matrix is symmetric and positive definite, and satisfies the following inequalities

$$m_1 \|\xi\|^2 \le \xi^T M(q) \, \xi \le m_2 \, \|\xi\|^2, \ \forall \xi \in \mathbb{R}^r$$

where  $m_1$  and  $m_2$  are known positive scalar constants, and  $\|\cdot\|$  represents the Euclidean norm.

Property 3. The inertia and centripetal-Coriolis matrices satisfy the following skew symmetric relation

$$\xi^{T}\left(\frac{1}{2}\dot{M}\left(q\right)-V_{m}\left(q,\dot{q}\right)\right)\xi=0, \ \forall \xi\in\mathbb{R}^{n}$$

where  $\dot{M}(q(t))$  is the time derivative of the inertial matrix. Equivalently,  $\dot{M}(q) = 2V_m(q;\dot{q})$ .

To quantify the tracking objective, the position tracking error,  $e(t) \in \mathbb{R}^n$ , and the filtered tracking error,  $r(t) \in \mathbb{R}^n$ , are defined as

$$e = q_d - q \tag{18}$$

$$r = \dot{e} + \alpha e \tag{19}$$

where  $q_d(t) \in \mathbb{R}^n$  represents the desired trajectory, whose first and second time derivatives exist and are continuous (i.e.,  $q_d(t) \in \mathbb{C}^2$ ). To quantify the parameter identification objective, the parameter estimation error,  $\tilde{\theta}(t) \in \mathbb{R}^m$ , is again defined as

$$\theta\left(t\right) = \theta - \theta\left(t\right) \tag{20}$$

where  $\hat{\theta}(t) \in \mathbb{R}^m$  represents the parameter estimate.

Taking the time derivative of (19), premultiplying by M(q), substituting in from (16), and adding and subtracting  $V_m(q, \dot{q}) r$  results in the following open-loop error dynamics

$$M(q)\dot{r} = Y_2(q, \dot{q}, q_d, \dot{q}_d, \ddot{q}_d)\theta - V_m(q, \dot{q})r - \tau$$
(21)

where  $Y_2: \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n \times m}$  is defined based on the relation

$$Y_2(q, \dot{q}, q_d, \dot{q}_d, \ddot{q}_d) \theta \triangleq M(q) \, \ddot{q}_d + V_m(q, \dot{q}) \left( \dot{q}_d + \alpha e \right) + F_d \dot{q} + G(q) + \alpha M(q) \, \dot{e}. \tag{22}$$

To achieve the tracking objective, the controller is designed as

$$\tau = Y_2 \hat{\theta} + e + k_1 r \tag{23}$$

where  $k_1 \in \mathbb{R}$  is a positive constant. To circumvent the need for  $\ddot{q}(t)$ , the update law can be formulated in terms of an integral, as

$$\dot{\hat{\theta}} = \Gamma Y_2^T r + k_2 \Gamma \sum_{i=1}^N \mathcal{Y}_i^T \left( \mathcal{U}_i - \mathcal{Y}_i \hat{\theta}(t) \right)$$
(24)

where  $\mathcal{Y}_{i} \triangleq \mathcal{Y}(t_{i}), \mathcal{U}_{i} \triangleq \mathcal{U}(t_{i}), \mathcal{Y}: [0, \infty) \to \mathbb{R}^{n \times m}$  and  $\mathcal{U}: [0, \infty) \to \mathbb{R}^{n}$  are defined as

$$\mathcal{U}(t_i) \triangleq \int_{\max\{t-\Delta t, 0\}}^t \tau(\sigma) \, d\sigma,$$
$$\mathcal{Y}(t_i) \triangleq Y_3(q(t), \dot{q}(t), q(t-\Delta t), \dot{q}(t-\Delta t)) + \int_{\max\{t-\Delta t, 0\}}^t Y_4(q(\sigma), \dot{q}(\sigma)) \, d\sigma,$$

and the functions  $Y_3: \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n \times m}$ ,  $Y_4: \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n \times m}$  are defined based on the relations

$$\begin{split} Y_3\left(q\left(t\right), \dot{q}\left(t\right), q\left(t - \Delta t\right), \dot{q}\left(t - \Delta t\right)\right) \theta &= M\left(q\left(t\right)\right) \dot{q}\left(t\right) - M\left(q\left(t - \Delta t\right)\right) \dot{q}\left(t - \Delta t\right) \\ Y_4\left(q, \dot{q}\right) \theta &= -\dot{M}\left(q\right) \dot{q} + V_m\left(q, \dot{q}\right) \dot{q} + F_d \dot{q} + G\left(q\right). \end{split}$$

Integrating both sides of (16) and using integration by parts on the inertial term yields

$$\int_{t-\Delta t}^{t} \tau(\sigma) \, d\sigma = Y_{3}\theta + \int_{t-\Delta t}^{t} Y_{4}\left(q\left(\sigma\right), \dot{q}\left(\sigma\right)\right) \, d\sigma\theta$$
$$\mathcal{U} = \mathcal{Y}\theta \tag{25}$$

Using the relation in (25), (24) can be rewritten as

$$\dot{\hat{\theta}} = \Gamma Y_2^T r + k_2 \Gamma \sum_{i=1}^N \mathcal{Y}_i^T \left( \mathcal{Y}_i \theta - \mathcal{Y}_i \hat{\theta}(t) \right).$$
(26)

Substituting the controller from (23) into the error dynamics in (21) results in the following closed-loop tracking error dynamics

$$M(q)\dot{r} = Y_2\tilde{\theta} - e - V_m(q,\dot{q})r - k_1r.$$
(27)

Similarly, taking the time derivative of (20) and substituting the parameter estimate update law from (26) results in the following closed-loop parameter estimation error dynamics

$$\dot{\tilde{\theta}} = -\Gamma Y_2^T r - k_2 \Gamma \left[ \sum_{i=1}^N \mathcal{Y}_i^T \mathcal{Y}_i \right] \tilde{\theta}.$$
(28)

## B. Stability Analysis

Similar to the analysis in Section III, two periods of time are considered. In Theorem 3 it is shown that the designed controller and adaptive update law are sufficient for the system to remain bounded for all time despite the lack of data and in Theorem 4 exponential convergence is established given a sufficiently rich history stack. Similar to Section III, an excitation condition is required to guarantee that the transition to the second phase happens in finite time, i.e.,

$$\exists \underline{\lambda}, T > 0 : \forall t \ge T, \ \lambda_{\min} \left\{ \sum_{i=1}^{N} \mathcal{Y}_{i}^{T} \mathcal{Y}_{i} \right\} \ge \underline{\lambda}.$$
<sup>(29)</sup>

**Theorem 3.** For the system defined in (16), the controller and adaptive update law defined in (23) and (24) ensure bounded tracking and parameter estimation errors.

*Proof:* Let  $V : \mathbb{R}^{2n+m} \to \mathbb{R}$  be a candidate Lyapunov function defined as

$$V(\eta) = \frac{1}{2}e^{T}e + \frac{1}{2}r^{T}M(q)r + \frac{1}{2}\tilde{\theta}^{T}\Gamma^{-1}\tilde{\theta}$$
(30)

where  $\eta(t) \triangleq \begin{bmatrix} e(t)^T & r(t)^T & \tilde{\theta}(t)^T \end{bmatrix}^T \in \mathbb{R}^{2n+m}$  is a composite state vector. Taking the time derivative of (30) and substituting (19), (27), and (28) yields

$$\dot{V}(\eta) = e^T \left( r - \alpha e \right) + \frac{1}{2} r^T \dot{M}(q) r - k_2 \tilde{\theta}^T \left[ \sum_{i=1}^N \mathcal{Y}_i^T \mathcal{Y}_i \right] \tilde{\theta} - \tilde{\theta}^T Y_2^T r + r^T \left( Y_2 \tilde{\theta}(t) - e - V_m(q, \dot{q}) r - k_1 r \right)$$

Simplifying and noting that  $\sum_{i=1}^{N} \mathcal{Y}_{i}^{T} \mathcal{Y}_{i}$  is always positive semi-definite,  $\dot{V}$  can be upper bounded as

$$\dot{V}(\eta) \leq -\alpha e^T e - k_1 r^T r.$$

Therefore,  $\eta(t)$  is bounded based on [28, Theorem 8.4]. Furthermore, since  $\dot{V}(\eta(t)) \leq 0$ ,  $V(\eta(T)) \leq V(\eta(0))$  and therefore  $\|\eta(T)\| \leq \sqrt{\frac{\beta_2}{\beta_1}} \|\eta(0)\|$ , where  $\beta_1 \triangleq \frac{1}{2} \min\left\{1, m_1, \lambda_{\min}\left\{\Gamma^{-1}\right\}\right\}$  and  $\beta_2 \triangleq \frac{1}{2} \max\left\{1, m_2, \lambda_{\max}\left\{\Gamma^{-1}\right\}\right\}$ .

**Theorem 4.** For the system defined in (16), the controller and adaptive update law defined in (23) and (24) ensure globally exponential tracking in the sense that

$$\|\eta(t)\| \le \left(\frac{\beta_2}{\beta_1}\right) \exp\left(\lambda_1 T\right) \|\eta(0)\| \exp\left(-\lambda_1 t\right), \ \forall t \in [0,\infty)$$
(31)

where  $\lambda_1 \triangleq \frac{1}{2\beta_2} \min \{ \alpha, k_1, k_2 \underline{\lambda} \}.$ 

*Proof:* Let  $V : \mathbb{R}^{2n+m} \to \mathbb{R}$  be a candidate Lyapunov function defined as in (30). Taking the time derivative of (30), substituting (19), (27), (28) and simplifying yields

$$\dot{V}(\eta) = -\alpha e^{T} e - k_{1} r^{T} r - k_{2} \tilde{\theta}^{T} \left[ \sum_{i=1}^{N} \mathcal{Y}_{i}^{T} \mathcal{Y}_{i} \right] \tilde{\theta}.$$
(32)

From the finite excitation condition,  $\lambda_{\min}\left\{\sum_{i=1}^{N} \mathcal{Y}_{i}^{T} \mathcal{Y}_{i}\right\} > 0$ ,  $\forall t \in [T, \infty)$ , which implies that  $\sum_{i=1}^{N} \mathcal{Y}_{i}^{T} \mathcal{Y}_{i}$  is positive definite, and therefore  $\dot{V}$  can be upper bounded as

$$\dot{V}(\eta) \leq -\alpha e^T e - k_1 r^T r - k_2 \underline{\lambda} \left\| \tilde{\theta} \right\|^2, \ \forall t \in [T, \infty).$$

Invoking [28, Theorem 4.10],  $\eta(t)$  is globally exponentially stable, i.e.,  $\forall t \in [T, \infty)$ ,

$$\left\|\eta\left(t\right)\right\| \leq \sqrt{\frac{\beta_{2}}{\beta_{1}}} \left\|\eta\left(T\right)\right\| \exp\left(-\lambda_{1}\left(t-T\right)\right)$$

The composite state vector can be further upper bounded using the results of Theorem 3, yielding (31).

*Remark* 2. Similar to Section III, using an appropriate data selection algorithm (e.g. the singular value maximization algorithm in [7]) ensures the minimum eigenvalue of  $\sum_{i=1}^{N} \mathcal{Y}_{i}^{T} \mathcal{Y}_{i}$  is always increasing, and therefore the Lyapunov function (30) is a common Lyapunov function [29].

# V. SIMULATION

A Monte Carlo simulation was performed to demonstrate the application of the theoretical results presented in Section III and to illustrate the increased performance and robustness to noise compared to the traditional state derivative based CL methods (e.g., [6]–[8]) across a wide variety of gain selections and noise realizations. The following example system was used in the simulations:

$$\dot{x}(t) = \begin{bmatrix} x_1^2 & \sin(x_2) & 0 & 0\\ 0 & x_2 \sin(t) & x_1 & x_1 x_2 \end{bmatrix} \theta + u(t)$$

where  $x: [0,\infty) \to \mathbb{R}^2$ ,  $u: [0,\infty) \to \mathbb{R}^2$ , the unknown parameters were selected as

$$\theta = \begin{bmatrix} 5 & 10 & 15 & 20 \end{bmatrix}^T$$

and the desired trajectory was selected as

$$x_d(t) = 10 \left(1 - e^{-0.1t}\right) \begin{bmatrix} \sin(2t) \\ 0.4\cos(3t) \end{bmatrix}.$$

For each of the 200 trials within the Monte Carlo simulation, the feedback and adaptation gains were selected as  $K = K_s I_2$ and  $\Gamma = \Gamma_s I_4$ , where  $K_s \in \mathbb{R}$  was sampled from a uniform distribution on (0.1, 15) and  $\Gamma_s \in \mathbb{R}$  was sampled from a uniform distribution on (0.3, 3). Also, the concurrent learning gain,  $k_{CL}$ , and the integration window,  $\Delta t$ , were sampled from uniform distributions with support on (0.002, 0.2) and (0.01, 1), respectively. After gain sampling, a simulation using each, the traditional state derivative based, and the integral based, CL update law was performed, with a simulation step size of 0.0004 seconds and additive white Gaussian noise on the measured state with standard deviation of 0.3. For each integral CL simulation, a buffer, with size based on  $\Delta t$  and the step size, was used to store the values of x, Y, and u during the time interval  $[t - \Delta t, t]$  and to calculate x(t),  $x(t - \Delta t)$ ,  $\mathcal{Y}(t)$  and  $\mathcal{U}(t)$ . Similarly, for the state derivative CL simulation, a buffer of the same size was used as the input to a moving average filter before calculating the state derivative via central finite difference. The size of the history stack and the simulation time span were kept constant across all trials at N = 20 and 100 seconds, respectively.

Since the moving average filter window used in the state derivative CL simulations provides an extra degree of freedom, the optimal filter window size was determined *a priori* for a fair comparison. The optimal filtering window was calculated by adding Gaussian noise, with the same standard deviation as in the simulation, to the desired trajectory, and selecting the window size that minimizes the root mean square error between the estimated and true  $\dot{x}_d$ . This process yielded an optimal filtering window of 0.5 seconds; however, the filtering window was truncated to  $\Delta t$  on trials where the sampled  $\Delta t$  was less than 0.5 seconds, i.e., filter window = min {0.5,  $\Delta t$ }.

The mean tracking error trajectory and parameter estimation error trajectory across all trials are depicted in Figures 1 and 2. To compare the overall performance of both methods, the RMS tracking error and the RMS parameter estimation error during the time interval  $t \in [60, 100]$  (i.e., after reaching steady state behavior) were calculated for each trial, and then the average RMS errors across all trials was determined. The final results of the Monte Carlo simulation are shown in Table I, illustrating the improved performance of integral CL versus state derivative CL.



Fig. 1. Mean state trajectory tracking errors across all trials.

Fig. 2. Mean parameter estimation errors across all trials.

 TABLE I

 Average steady state RMS tracking and RMS parameter estimation errors across all simulations, for integral concurrent learning (ICL) and traditional derivative-based concurrent learning (DCL).

	$e_1$	$e_2$	$ ilde{ heta}_1$	$ ilde{ heta}_2$	$\tilde{ heta}_3$	$ ilde{ heta}_4$
ICL	0.1078	0.2117	0.0507	0.3100	0.1867	0.1121
DCL	0.2497	0.6717	0.1802	1.3376	0.3753	0.2382

A second set of simulations were performed to demonstrate the application of ICL to Euler-Lagrange systems and verify the development in Section IV. A two-link planar robot was simulated, with dynamics shown in (33), where  $c_2$  denotes  $cos(q_2)$  and  $s_2$  denotes  $sin(q_2)$ . The nominal parameters values of the model are

$$p_1 = 3.473 \qquad f_{d1} = 5.3 \\ p_2 = 0.196 \qquad f_{d2} = 1.1 \\ p_3 = 0.242$$

and the controller gains were selected as

 $\alpha = 1.0$   $\Gamma = 0.3I_5$  $k_1 = 1.0$   $k_2 = 3.0.$ 

The desired trajectory was selected as

$$q_{d1} = (1 + 10 \exp(-2t)) \sin(t),$$
  

$$q_{d2} = (1 + 10 \exp(-t)) \cos(3t),$$

and a history stack of up to 20 data points was used for ICL. The results of the simulation are shown in Figures 3 and 4, where it is clear that the tracking and parameter estimation error exponentially converge. In comparison, the error trajectories in Figures 5 and 6 demonstrate the performance of a purely gradient based adaptive update law (i.e.,  $k_2 = 0$ ), in which trajectory tracking performance is degraded and the system parameters are not identified.



Fig. 3. Evolution of the joint angles for the planar robot simulation using an ICL adaptation law.



Fig. 5. Evolution of the joint angles for the planar robot simulation using an gradient-based adaptation law.



Fig. 4. Evolution of the parameter estimation errors for the planar robot simulation using an ICL adaptation law.



Fig. 6. Evolution of the parameter estimation errors for the planar robot simulation using a gradient-based adaptation law.

#### VI. CONCLUSION

A modified concurrent learning adaptive update law was developed, resulting in guarantees on the convergence of the parameter estimation errors without requiring persistent excitation or the estimation of state derivatives. The development in this paper represents a significant improvement in online system identification. Whereas PE is required in the majority of adaptive methods for parameter estimation convergence (usually ensured through the use of a probing signal that is not considered in the Lyapunov analysis), the technique described in this paper does not require PE. Furthermore, the formulation of concurrent learning in this paper circumvents the need to estimate the unmeasurable state derivatives, therefore avoiding the design and tuning of a state derivative estimator. This formulation is more robust to noise, i.e., has better tracking and estimation performance, compared to other concurrent learning designs, as demonstrated by the included Monte Carlo simulation.

A tuning parameter that results from this design is the integration time window,  $\Delta t$ . As the integration window increases, the difference between the prediction of the state evolution based on current parameter estimates (i.e.,  $U_i + Y_i\hat{\theta}$ ) and the actual state evolution (i.e.,  $x(t_i) - x(t_i - \Delta t)$ ) should increase, therefore providing a larger error signal from which to learn. On the other hand, a larger integration window increases the effect of disturbances and noise since these signals would also be integrated, resulting in a larger ultimate error bound (see e.g., [8] for a discussion on the effects of disturbances on the ultimate error). Therefore, future efforts will investigate optimal selection of the integration window based on disturbance and noise characteristics, as well as identifying any unknown parameters in the control effectiveness.

#### REFERENCES

- [2] K. Narendra and A. Annaswamy, Stable Adaptive Systems. Prentice-Hall, Inc., 1989.
- [3] S. Sastry and M. Bodson, Adaptive Control: Stability, Convergence, and Robustness. Upper Saddle River, NJ: Prentice-Hall, 1989.
- [4] M. A. Duarte and K. Narendra, "Combined direct and indirect approach to adaptive control," *IEEE Trans. Autom. Control*, vol. 34, no. 10, pp. 1071–1075, Oct. 1989.
- [5] M. Krstić, P. V. Kokotović, and I. Kanellakopoulos, "Transient-performance improvement with a new class of adaptive controllers," Syst. Control Lett., vol. 21, no. 6, pp. 451–461, 1993.
- [6] G. V. Chowdhary and E. N. Johnson, "Theory and flight-test validation of a concurrent-learning adaptive controller," J. Guid. Control Dynam., vol. 34, no. 2, pp. 592–607, Mar. 2011.
- [7] G. Chowdhary, "Concurrent learning for convergence in adaptive control without persistency of excitation," Ph.D. dissertation, Georgia Institute of Technology, Dec. 2010.
- [8] G. Chowdhary, T. Yucelen, M. Mühlegg, and E. N. Johnson, "Concurrent learning adaptive control of linear systems with exponentially convergent bounds," Int. J. Adapt. Control Signal Process., vol. 27, no. 4, pp. 280–301, 2013.
- [9] M. Mühlegg, G. Chowdhary, and E. Johnson, "Concurrent learning adaptive control of linear systems with noisy measurements," in Proc. AIAA Guid. Navig. Control Conf., 2012.
- [10] R. Kamalapurkar, B. Reish, G. Chowdhary, and W. E. Dixon, "Concurrent learning for parameter estimation using dynamic state-derivative estimators," *IEEE Trans. Autom. Control*, submitted (see also arXiv:1507.08903).
- [11] J. J. Slotine and W. Li, "Composite adaptive control of robot manipulators," Automatica, vol. 25, no. 4, pp. 509–519, Jul. 1989.
- [12] K. Y. Volyanskyy, W. M. Haddad, and A. J. Calise, "A new neuroadaptive control architecture for nonlinear uncertain dynamical systems: Beyond σ-and e-modifications," *IEEE Transactions on Neural Networks*, vol. 20, no. 11, pp. 1707–1723, 2009.
- [13] Y. Pan, T. Sun, and H. Yu, "Composite adaptive dynamic surface control using online recorded data," *International Journal of Robust and Nonlinear Control*, vol. 26, no. 18, pp. 3921–3936, 2016.
- [14] Y. Pan, X. Li, and H. Yu, "Least-squares learning control with guaranteed parameter convergence," in Automation and Computing (ICAC), 2016 22nd International Conference on. IEEE, 2016, pp. 132–137.
- [15] Y. Pan and H. Yu, "Composite learning from adaptive dynamic surface control," *IEEE Transactions on Automatic Control*, vol. 61, no. 9, pp. 2603–2609, 2016.
- [16] Y. Pan, J. Zhang, and H. Yu, "Model reference composite learning control without persistency of excitation," *IET Control Theory & Applications*, vol. 10, no. 16, pp. 1963–1971, 2016.
- [17] N. Cho, H.-S. Shin, Y. Kim, and A. Tsourdos, "Composite model reference adaptive control with parameter convergence under finite excitation," *IEEE Trans. Autom. Control*, vol. 63, no. 3, pp. 811–818, Mar. 2017.
- [18] N. Cho and Y. Kim, "Basis integral concurrent learning model reference adaptive control," in Eur. Control Conf., Jul. 2016.
- [19] S. B. Roy, S. Bhasin, and I. N. Kar, "Combined mrac for unknown mimo lti systems with parameter convergence," *IEEE Trans. Autom. Control*, vol. 63, no. 1, pp. 283–290, Jan. 2018.
- [20] —, "A uges switched mrac architecture using initial excitation," in *IFAC-PapersOnLine*, vol. 50, no. 1, Jul. 2017, pp. 7044–7051.
- [21] R. Engel and G. Kreisselmeier, "A continuous-time observer which converges in finite time," *IEEE Trans. Autom. Control*, vol. 47, no. 7, pp. 1202–1204, 2002.
- [22] R. Ortega, "An on-line least-squares parameter estimator with finite convergence time," Proc. IEEE, vol. 76, no. 7, pp. 847-848, Jul. 1988.
- [23] S. Aranovskiy, A. Bobtsov, R. Ortega, and A. Pyrkin, "Parameters estimation via dynamic regressor extension and mixing," in *Am. Control Conf.*, Jul. 2016.
- [24] P. C. Young, "An instrumental variable method for real-time identification of a noisy process," Automatica, vol. 6, no. 2, pp. 271-287, Mar. 1970.
- [25] S. Aranovskiy, A. Bobtsov, R. Ortega, and A. Pyrkin, "Performance enhancement of parameter estimators via dynamic regressor extension and mixing," *IEEE Trans Autom. Contol*, vol. 62, no. 7, pp. 3546–3550, Jul. 2017.
- [26] V. S. Gromov, O. I. Borisov, A. A. Pyrkin, A. A. Bobtsov, S. A. Kolyubin, and S. V. Aranovskiy, "The drem approach for chaotic oscillators parameter estimation with improved performance," in *IFAC-PapersOnLine*, vol. 50, no. 1, Jul. 2017, pp. 7027–7031.
- [27] S. B. Roy, S. Bhasin, and I. N. Kar, "Parameter convergence via a novel pi-like composite adaptive controller for uncertain euler-lagrange systems," in IEEE Conf. Decis. Control, Dec. 2016.
- [28] H. K. Khalil, Nonlinear Systems, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2002.
- [29] D. Liberzon, Switching in Systems and Control. Birkhauser, 2003.
- [30] F. L. Lewis, D. M. Dawson, and C. Abdallah, Robot Manipulator Control Theory and Practice. CRC, 2003.
- [31] M. Spong and M. Vidyasagar, Robot Dynamics and Control. New York: John Wiley & Sons Inc., 1989.