Received: 24 April 2018 | Revised: 20 September 2018 | Accepted: 24 September 2018

DOI: 10.1002/acs.2945

SPECIAL ISSUE ARTICLE

WILEY

Integral concurrent learning: Adaptive control with parameter convergence using finite excitation

Anup Parikh[1] | Rushikesh Kamalapurkar[2] | Warren E. Dixon[3]

[1]Robotics Research & Development, Sandia National Laboratories, Albuquerque, New Mexico

[2]School of Mechanical and Aerospace Engineering, Oklahoma State University, Stillwater, Oklahoma

[3]Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, Florida

Correspondence
Anup Parikh, Robotics Research & Development, Sandia National Laboratories, Albuquerque, NM 87123.
Email: aparikh@sandia.gov

Funding information
NEEC, Grant/Award Number: N00174-18-1-0003; AFOSR, Grant/Award Number: FA9550-18-1-0109; Office of Naval Research, Grant/Award Number: N00014-13-1-0151; NSF, Grant/Award Number: 1509516 and 1762829

Summary

Concurrent learning (CL) 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 CL 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.

KEYWORDS

adaptive control, nonlinear systems, system identification, uncertain systems

1 | 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 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 the works of Duarte and Narendra,[4] Krstić et al,[5] and Chowdhary and Johnson[6]), a new adaptive update scheme known as concurrent learning (CL) was recently developed in the pioneering works of Chowdhary et al.[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

 © 2018 John Wiley & Sons, Ltd.

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, eg, the works of Mühlegg et al[9] and Kamalapurkar et al.[10] However, these methods typically require tuning parameters such as an observer gain and switching threshold in the case of the online derivative estimator and basis, basis order, covariance, and time window in the case of smoothing, to produce satisfactory results.

In this note, we reformulate the derivative-based CL method (DCL) 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 (see, eg, the works of Slotine and Li,[11] Volyanskyy et al,[12] and Pan et al[13]); however, they still require PE to ensure exponential convergence. Recently, results in other works[14-18] have shown convergence using an interval or finite excitation condition, although they either require measurements of state derivatives (see, eg, the work of Pan and Yu[15]), require determining the analytical Jacobian of the regressor (see, eg, the work of Pan et al[14]), or are developed in a model reference adaptive control context,[16,17,19-21] which essentially assume that desired trajectories are generated from an LTI system and may rely on a matching condition, rather than the general nonlinear systems considered here without such assumptions. Some results[22-27] have theoretical analogs to those presented here, and some use filtering techniques to avoid data storage requirements, although 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 analogous to the smoothing buffer window that is already required for smoothing-based techniques. Despite the reformulation, the stability results still hold (ie, parameter convergence) and Monte Carlo simulation results suggest greater robustness to noise compared to DCL implementations. The technique is also applied to Euler-Lagrange (EL) systems to demonstrate the use of integral CL (ICL) for systems with unmatched uncertainties, similar to the very recent results in the work of Pan and Yu[18] that uses interval excitation (IE). Compared to some other similar approaches applied to EL systems, our approach selectively collects data rather than incorporating the entire history so as to consider the most informative data for parameter estimation, in contrast to, eg, the work of Roy et al.[28] Furthermore, our approach can learn from multiple exciting intervals, some of which may not be sufficiently exciting (ie, do not have a nonzero minimum eigenvalue) but, in aggregate, are sufficiently rich, in contrast to, eg, the work of Pan and Yu.[18] To develop the robustness that single interval IE methods natively provide, ICL can be integrated with a purging technique.[10]

2 | CONTROL OBJECTIVE

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

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

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 parameterized in the unknown parameters, ie,

$$f(x, t) = Y(x, t)\theta, \tag{2}$$

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

$$e(t) \triangleq x(t) - x_d(t) \tag{3}$$

$$\tilde{\theta}(t) \triangleq \theta - \hat{\theta}(t), \tag{4}$$

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, eg, $x(t)$ 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) yield the closed-loop error dynamics

$$\begin{aligned}\dot{e} &= Y(x,t)\theta + \dot{x}_d - Y(x,t)\hat{\theta} - Ke - \dot{x}_d \\ &= Y(x,t)\tilde{\theta} - Ke. \end{aligned} \tag{6}$$

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

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

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

$$\dot{\hat{\theta}}(t) \triangleq \Gamma Y(x,t)^T e + k_{\text{CL}}\Gamma \sum_{i=1}^{N} \mathcal{Y}_i^T \left(x(t_i) - x(t_i - \Delta t) - \mathcal{U}_i - \mathcal{Y}_i\hat{\theta}\right), \tag{8}$$

where $k_{\text{CL}} \in \mathbb{R}$ 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, $\mathcal{Y}_i \triangleq \mathcal{Y}(t_i)$, $\mathcal{U}_i \triangleq \mathcal{U}(t_i)$,

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

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

Furthermore, $0_{n\times m}$ 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 CL term (ie, 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 the work of Chowdhary[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 $\lceil mnhb\Delta t \rceil$and $\lceil nhb\Delta t \rceil$ bytes, respectively, where h is the control loop rate in cycles per second and b is the number of bytes per value (eg, 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 ICL-based adaptive update law in (8) differs from traditional DCL update laws given in, eg, the works of Chowdhary et al.[6-8] Specifically, the state derivative, control, and regressor terms, ie, $\dot{x}$, u, and Y, respectively, used in the aforementioned works[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),

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

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

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

Note that (9) and (10) are piecewise continuous in time, the CL 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.

3 | 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 $\eta(t) \triangleq \left[\, e^T \ \ \tilde{\theta}^T \,\right]^T$. In addition, let $\lambda_{\min}\{\cdot\}$ and $\lambda_{\max}\{\cdot\}$ represent 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, $\exists \underline{\lambda} > 0,\ \exists T > \Delta t : \forall t \geq T,\ \lambda_{\min}\left\{\sum_{i=1}^{N} \mathcal{Y}_i^T \mathcal{Y}_i\right\} \geq \underline{\lambda}$.

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

$$\int_t^{t+\Delta t} Y^T(x(\tau),\tau)\, Y(x(\tau),\tau)\, d\tau \geq \alpha I > 0,\ \ \forall t > 0, \tag{13}$$

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 (ie, $\lambda_{\min}\left\{\sum_{i=1}^{N} \mathcal{Y}_i^T \mathcal{Y}_i\right\}$ can be calculated), whereas in PE, Δt 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, eg, the singular value maximization algorithm in the work of Chowdhary[7] to ensure that 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 (eg, fourth-order Runge-Kutta methods have $\mathcal{O}(h^5)$ local truncation errors), the threshold should also be selected sufficiently large to ensure that 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) ensure bounded tracking and parameter estimation errors.*

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

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

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 theorem 8.4 in the work of Khalil.[29] Furthermore, 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 $\beta_1 \triangleq \frac{1}{2} \min\left\{1, \lambda_{\min}\left\{\Gamma^{-1}\right\}\right\}$ and $\beta_2 \triangleq \frac{1}{2} \max\left\{1, \lambda_{\max}\left\{\Gamma^{-1}\right\}\right\}$. □

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

$$\|\eta(t)\| \leq \left(\frac{\beta_2}{\beta_1}\right) \exp(\lambda_1 T)\, \|\eta(0)\| \exp(-\lambda_1 t),\ \ \forall t \in [0, \infty). \tag{15}$$

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

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

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 η. Invoking theorem 4.10 in the work of Khalil,[29] e and $\tilde{\theta}$ are globally exponentially stable, ie, $\forall t \in [T, \infty)$,

$$\|\eta(t)\| \leq \sqrt{\frac{\beta_2}{\beta_1}}\, \|\eta(T)\| \exp\left(-\lambda_1 (t - T)\right),$$

where $\lambda_1 \triangleq \frac{1}{2\beta_2} \min\left\{\lambda_{\min}\{K\}, k_{\mathrm{CL}}\underline{\lambda}\right\}$. 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 (eg, the singular value maximization algorithm in the work of Chowdhary[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[30] as data is continuously added to the history stack.

4 | EXTENSION TO EL SYSTEMS

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

4.1 | Control development

Consider EL dynamics of the form in chapter 2.3 in the work of Lewis et al[31] and in chapter 9.3 in the work of Spong and Vidyasagar[32]

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

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 chapter 2.3 in the work of Lewis et al[31]), which hold for a large class of physical systems.

Property 1. The system in (16) can be linearly parameterized, ie, the left-hand side of (16) can be rewritten as

$$Y_1\left(q, \dot{q}, \ddot{q}\right)\theta = M(q)\ddot{q} + V_m\left(q, \dot{q}\right)\dot{q} + F_d\dot{q} + G(q), \tag{17}$$

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:

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

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:

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

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 (ie, $q_d(t) \in \boldsymbol{C}^2$). To quantify the parameter identification objective, the parameter estimation error, $\tilde{\theta}(t) \in \mathbb{R}^m$, is again defined as

$$\tilde{\theta}(t) = \theta - \hat{\theta}(t), \tag{20}$$

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 $V_m(q, \dot{q})r$ result in the following open-loop error dynamics:

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

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

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

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

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

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

$$\mathcal{U}(t_i) \triangleq \int_{\max\{t-\Delta t,\, 0\}}^{t}\tau(\sigma)d\sigma,$$

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

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

$$Y_3\left(q(t), \dot{q}(t), q(t-\Delta t), \dot{q}(t-\Delta t)\right)\theta = M\left(q(t)\right)\dot{q}(t) - M\left(q(t-\Delta t)\right)\dot{q}(t-\Delta t)$$

$$Y_4\left(q, \dot{q}\right)\theta = -\dot{M}(q)\dot{q} + V_m(q, \dot{q})\dot{q} + F_d\dot{q} + G(q).$$

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

$$\int_{t-\Delta t}^{t}\tau(\sigma)d\sigma = Y_3\theta + \int_{t-\Delta t}^{t} Y_4\left(q(\sigma), \dot{q}(\sigma)\right)d\sigma\theta$$

$$\mathcal{U} = \mathcal{Y}\theta. \tag{25}$$

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

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

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

$$M(q)\dot{r} = Y_2\tilde{\theta} - e - V_m(q,\dot{q})r - k_1 r. \tag{27}$$

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

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

4.2 | Stability analysis

Similar to the analysis in Section 3, 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 3, an excitation condition is required to guarantee that the transition to the second phase happens in finite time, ie,

$$\exists \underline{\lambda},\ T > 0 : \forall t \geq T,\ \lambda_{\min}\left\{\sum_{i=1}^{N}\mathcal{Y}_i^T\mathcal{Y}_i\right\} \geq \underline{\lambda}. \tag{29}$$

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

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

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) yield

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

Simplifying and noting that $\sum_{i=1}^{N}\mathcal{Y}_i^T\mathcal{Y}_i$ is always positive semidefinite, $\dot{V}$ 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 theorem 8.4 in the work of Khalil.[29] 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 $\beta_1 \triangleq \frac{1}{2}\min\left\{1,\ m_1,\ \lambda_{\min}\{\Gamma^{-1}\}\right\}$ and $\beta_2 \triangleq \frac{1}{2}\max\left\{1,\ m_2,\ \lambda_{\max}\{\Gamma^{-1}\}\right\}$. □

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*

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

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), and (28) and simplifying yield

$$\dot{V}(\eta) = -\alpha e^T e - k_1 r^T r - k_2\tilde{\theta}^T\left[\sum_{i=1}^{N}\mathcal{Y}_i^T\mathcal{Y}_i\right]\tilde{\theta}. \tag{32}$$

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

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

Invoking theorem 4.10 in the work of Khalil,[29] $\eta(t)$ is globally exponentially stable, ie, $\forall t \in [T, \infty)$,

$$\|\eta(t)\| \leq \sqrt{\frac{\beta_2}{\beta_1}} \|\eta(T)\| \exp\left(-\lambda_1 (t - T)\right).$$

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

Remark 2. Similar to Section 3, using an appropriate data selection algorithm (eg, the singular value maximization algorithm in the work of Chowdhary[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.[30]

5 | SIMULATION

A Monte Carlo simulation was performed to demonstrate the application of the theoretical results presented in Section 3 and to illustrate the increased performance and robustness to noise compared to the traditional state DCL methods (see, eg, the works of Chowdhary et al[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 $(0.3, 3)$. In addition, the CL gain, k_{CL}, and the integration window, Δt, 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 integral-based, CL update laws, 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 ICL simulation, a buffer, with size based on Δt 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 (RMS) 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 Δt on trials where the sampled Δt was less than 0.5 seconds, ie, *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 $t \in [60, 100]$ (ie, after reaching steady-state behavior) were calculated for each trial, and

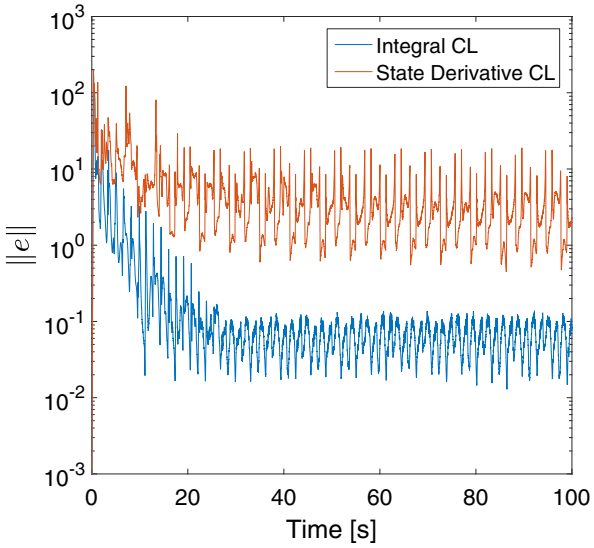

FIGURE 1 Mean state trajectory tracking errors across all trials [Colour figure can be viewed at wileyonlinelibrary.com]

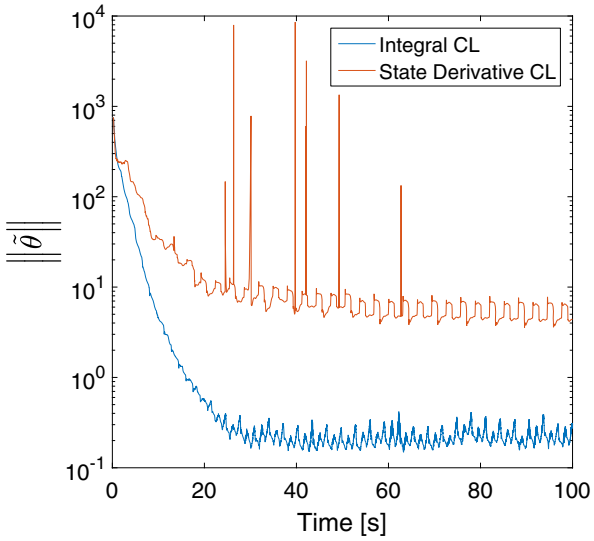

FIGURE 2 Mean parameter estimation errors across all trials [Colour figure can be viewed at wileyonlinelibrary.com]

then, the average RMS errors across all trials were determined. The final results of the Monte Carlo simulation are shown in Table 1, illustrating the improved performance of ICL versus state derivative CL.

$$\underbrace{\begin{bmatrix} p_1 + 2p_3c_2 & p_2 + p_3c_2 \\ p_2 + p_3c_2 & p_2 \end{bmatrix}}_{M(q)} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \underbrace{\begin{bmatrix} -p_3s_2\dot{q}_2 & -p_3s_2\left(\dot{q}_1 + \dot{q}_2\right) \\ p_3s_2\dot{q}_1 & 0 \end{bmatrix}}_{V_m(q,\dot{q})} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \underbrace{\begin{bmatrix} f_{d1} & 0 \\ 0 & f_{d2} \end{bmatrix}}_{F_d} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}. \tag{33}$$

A second set of simulations were performed to demonstrate the application of ICL to EL systems and verify the development in Section 4. 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

$$\begin{aligned} p_1 &= 3.473 & f_{d1} &= 5.3 \\ p_2 &= 0.196 & f_{d2} &= 1.1 \\ p_3 &= 0.242 \end{aligned}$$

TABLE 1 Average steady-state RMS tracking and RMS parameter estimation errors across all simulations, for ICL and traditional DCL

	e_1	e_2	$\tilde{\theta}_1$	$\tilde{\theta}_2$	$\tilde{\theta}_3$	$\tilde{\theta}_4$
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

Abbreviations: DCL, derivative-based concurrent learning; ICL, integral concurrent learning; RMS, root mean square.

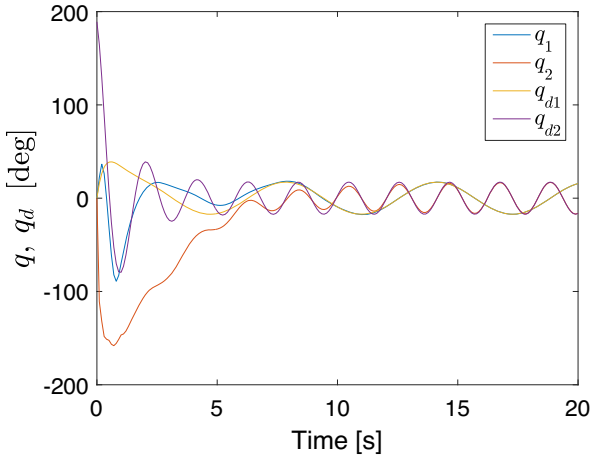

FIGURE 3 Evolution of the joint angles for the planar robot simulation using an integral concurrent learning adaptation law [Colour figure can be viewed at wileyonlinelibrary.com]

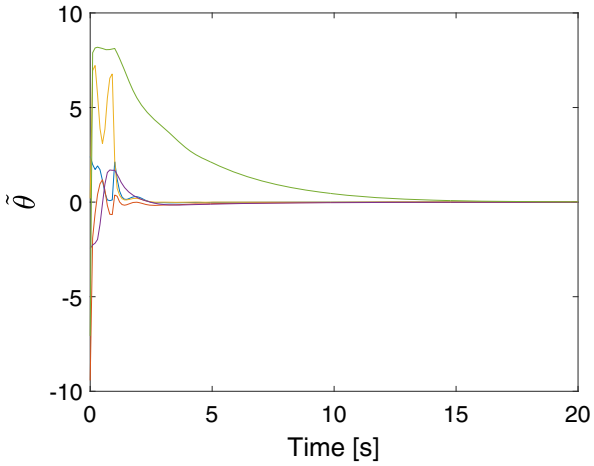

FIGURE 4 Evolution of the parameter estimation errors for the planar robot simulation using an integral concurrent learning adaptation law [Colour figure can be viewed at wileyonlinelibrary.com]

and the controller gains were selected as

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

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

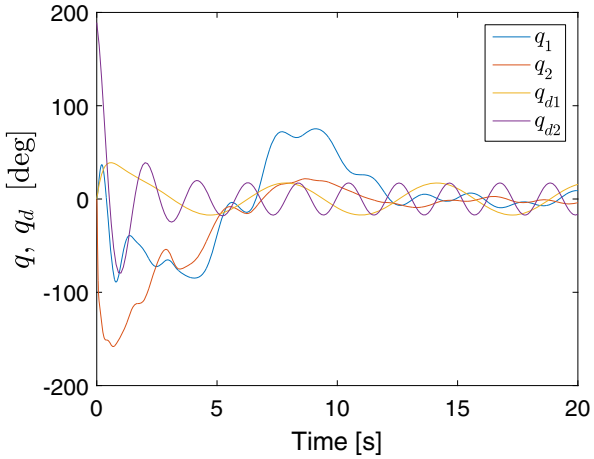

FIGURE 5 Evolution of the joint angles for the planar robot simulation using a gradient-based adaptation law [Colour figure can be viewed at wileyonlinelibrary.com]

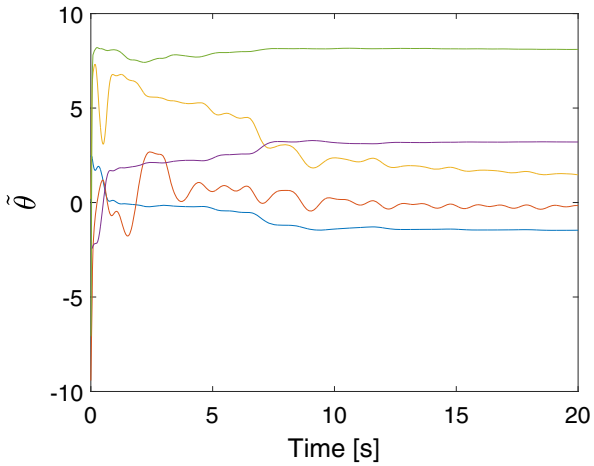

FIGURE 6 Evolution of the parameter estimation errors for the planar robot simulation using a gradient-based adaptation law [Colour figure can be viewed at wileyonlinelibrary.com]

trajectories in Figures 5 and 6 demonstrate the performance of a purely gradient-based adaptive update law (ie, $k_2 = 0$), in which trajectory tracking performance is degraded and the system parameters are not identified.

6 | CONCLUSION

A modified CL adaptive update law was developed, resulting in guarantees on the convergence of the parameter estimation errors without requiring PE or the estimation of state derivatives. The development in this paper represents a significant improvement in online system identification. While 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 CL 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, ie, has better tracking and estimation performance, compared to other CL designs, as demonstrated by the included Monte Carlo simulation.

A tuning parameter that results from this design is the integration time window, Δt. As the integration window increases, the difference between the prediction of the state evolution based on current parameter estimates (ie, $\mathcal{U}_i + \mathcal{Y}_i\hat{\theta}$) and the actual state evolution (ie, $x(t_i) - x(t_i - \Delta t)$) 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, eg, the work of Chowdhary et al[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.

ACKNOWLEDGEMENTS

This research was supported in part by the Naval Engineering Education Consortium (grant N00174-18-1-0003), the Air Force Office of Scientific Research (grant FA9550-18-1-0109), the Office of Naval Research (grant N00014-13-1-0151), and the National Science Foundation (grants 1509516 and 1762829). 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.

ORCID

Anup Parikh http://orcid.org/0000-0003-3120-1721

REFERENCES

1. Ioannou P, Sun J. *Robust Adaptive Control*. Upper Saddle River, NJ: Prentice Hall; 1996.
2. Narendra K, Annaswamy A. *Stable Adaptive Systems*. Upper Saddle River, NJ: Prentice Hall; 1989.
3. Sastry S, Bodson M. *Adaptive Control: Stability, Convergence, and Robustness*. Upper Saddle River, NJ: Prentice Hall; 1989.
4. Duarte MA, Narendra K. Combined direct and indirect approach to adaptive control. *IEEE Trans Autom Control*. 1989;34(10):1071-1075.
5. Krstić M, Kokotović PV, Kanellakopoulos I. Transient-performance improvement with a new class of adaptive controllers. *Syst Control Lett*. 1993;21(6):451-461.
6. Chowdhary GV, Johnson EN. Theory and flight-test validation of a concurrent-learning adaptive controller. *J Guid Control Dyn*. 2011;34(2):592-607.
7. Chowdhary G. *Concurrent Learning for Convergence in Adaptive Control without Persistency of Excitation* [PhD dissertation]. Atlanta, GA: Georgia Institute of Technology; 2010.
8. Chowdhary G, Yucelen T, Mühlegg M, Johnson EN. Concurrent learning adaptive control of linear systems with exponentially convergent bounds. *Int J Adapt Control Signal Process*. 2013;27(4):280-301.
9. Mühlegg M, Chowdhary G, Johnson E. Concurrent learning adaptive control of linear systems with noisy measurements. Paper presented at: AIAA Guidance, Navigation, and Control Conference; 2012; Chicago, IL.
10. Kamalapurkar R, Reish B, Chowdhary G, Dixon WE. Concurrent learning for parameter estimation using dynamic state-derivative estimators. *IEEE Trans Autom Control*. 2017;62(7):3594-3601.
11. Slotine JJ, Li W. Composite adaptive control of robot manipulators. *Automatica*. 1989;25(4):509-519.
12. Volyanskyy KY, Haddad WM, Calise AJ. A new neuroadaptive control architecture for nonlinear uncertain dynamical systems: beyond σ- and e-modifications. *IEEE Trans Neural Netw*. 2009;20(11):1707-1723.
13. Pan Y, Sun T, Yu H. Composite adaptive dynamic surface control using online recorded data. *Int J Robust Nonlinear Control*. 2016;26(18):3921-3936.
14. Pan Y, Li X, Yu H. Least-squares learning control with guaranteed parameter convergence. Paper presented at: 2016 22nd International Conference on Automation and Computing (ICAC); 2016; Colchester, UK.
15. Pan Y, Yu H. Composite learning from adaptive dynamic surface control. *IEEE Trans Autom Control*. 2016;61(9):2603-2609.
16. Pan Y, Zhang J, Yu H. Model reference composite learning control without persistency of excitation. *IET Control Theory Appl*. 2016;10(16):1963-1971.
17. Cho N, Shin HS, Kim Y, Tsourdos A. Composite model reference adaptive control with parameter convergence under finite excitation. *IEEE Trans Autom Control*. 2018;63(3):811-818.
18. Pan Y, Yu H. Composite learning robot control with guaranteed parameter convergence. *Automatica*. 2018;89:398-406.
19. Cho N, Kim Y. Basis integral concurrent learning model reference adaptive control. Paper presented at: European Control Conference; 2016; Aalborg, Denmark.
20. Roy SB, Bhasin S, Kar IN. Combined MRAC for unknown MIMO LTI systems with parameter convergence. *IEEE Trans Autom Control*. 2018;63(1):283-290.
21. Roy SB, Bhasin S, Kar IN. A UGES switched MRAC architecture using initial excitation. *IFAC Pap*. 2017;50(1):7044-7051.
22. Engel R, Kreisselmeier G. A continuous-time observer which converges in finite time. *IEEE Trans Autom Control*. 2002;47(7):1202-1204.
23. Ortega R. An on-line least-squares parameter estimator with finite convergence time. *Proc IEEE*. 1988;76(7):847-848.
24. Aranovskiy S, Bobtsov A, Ortega R, Pyrkin A. Parameters estimation via dynamic regressor extension and mixing. Paper presented at: American Control Conference; 2016; Boston, MA.
25. Young PC. An instrumental variable method for real-time identification of a noisy process. *Automatica*. 1970;6(2):271-287.

26. Aranovskiy S, Bobtsov A, Ortega R, Pyrkin A. Performance enhancement of parameter estimators via dynamic regressor extension and mixing. *IEEE Trans Autom Contol*. 2017;62(7):3546-3550.
27. Gromov VS, Borisov OI, Pyrkin AA, Bobtsov AA, Kolyubin SA, Aranovskiy SV. The DREM approach for chaotic oscillators parameter estimation with improved performance. *IFAC Pap*. 2017;50(1):7027-7031.
28. Roy SB, Bhasin S, Kar IN. Parameter convergence via a novel pi-like composite adaptive controller for uncertain Euler-Lagrange systems. Paper presented at: IEEE Conference on Decision and Control; 2016; Las Vegas, NV.
29. Khalil HK. *Nonlinear Systems*. 3rd ed. Upper Saddle River, NJ: Prentice Hall; 2002.
30. Liberzon D. *Switching in Systems and Control*. Basel, Switzerland: Birkhäuser; 2003.
31. Lewis FL, Dawson DM, Abdallah C. *Robot Manipulator Control: Theory and Practice*. Boca Raton, FL: CRC; 2003.
32. Spong M, Vidyasagar M. *Robot Dynamics and Control*. New York, NY: John Wiley & Sons; 1989.

How to cite this article: Parikh A, Kamalapurkar R, Dixon WE. Integral concurrent learning: Adaptive control with parameter convergence using finite excitation. *Int J Adapt Control Signal Process*. 2019;33:1775–1787. https://doi.org/10.1002/acs.2945