Hybrid Concurrent Learning for Hybrid Linear Regression

Ryan S. Johnson and Ricardo G. Sanfelice

***Abstract*— We consider the problem of estimating a vector of unknown constant parameters for a linear regression model whose input and output signals are hybrid – that is, they exhibit both continuous (flow) and discrete (jump) evolution. Using a hybrid systems framework, we propose a hybrid algorithm capable of operating during both flows and jumps, that utilizes current measurements alongside stored data for adaptation. We show that our algorithm guarantees exponential convergence of the parameter estimate to the true value under a new notion of excitation that relaxes both the classical continuous-time and discrete-time persistence of excitation conditions and a recently proposed hybrid persistence of excitation condition. Simulation results show the merits of our proposed approach.**

I. Introduction

The problem of parameter estimation arises in many engineering applications [1]. One such related problem is linear regression, which is typically solved using the gradient descent algorithm [2], [3]. To ensure convergence of the parameter estimation error to zero, gradient descent algorithms require a *persistence of excitation* (PE) condition [1], [3]. However, PE is often difficult to verify online. On the other hand, *concurrent learning* (CL) [4], [5], a recently introduced method of parameter estimation that uses stored data alongside current measurements for adaptation, ensures convergence of the parameter estimation error to zero without requiring the usually restrictive PE condition.

In this paper, we extend the CL approach to estimate the unknown parameters of a linear regression model whose input and output signals are hybrid. We refer to such problems as *hybrid linear regression*. We begin with a motivational example in Section I followed by a literature review of CL algorithms in Section II. The dynamics of our proposed hybrid parameter estimation scheme, called *hybrid concurrent learning* (hybrid CL), are described in Section III. In Section IV, we show that our proposed algorithm ensures exponential convergence of the parameter estimation error to zero without requiring the regressor to satisfy the hybrid persistence of excitation condition in [6]. Instead, we impose a new notion of excitation that incorporates the information provided by both the regressor and the stored data. Criteria for selecting data for storage during flows and jumps are proposed in Section V. Examples are given in Section VI and concluding remarks are in Section VII. Due to space constraints, proofs and other details are sketched and will be published elsewhere.

Research partially supported by NSF Grants no. ECS-1710621, CNS-2039054, and CNS-2111688, by AFOSR Grants no. FA9550-19-1-0053, FA9550-19-1-0169, and FA9550-20-1-0238, and by ARO Grant no. W911NF-20-1-0253.

The authors are with the Department of Electrical and Computer Engineering, University of California, Santa Cruz, CA 95064, USA; rsjohnson@ucsc.edu, ricardo@ucsc.edu

A. Notation

We denote the set of real, nonnegative real, and positive real numbers as $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, and $\mathbb{R}_{>0}$, respectively. We denote the set of natural numbers (including zero) as $\mathbb{N}$. The matrix I denotes the identity matrix of appropriate dimension. For $x, y \in \mathbb{R}^n$, we write $[x^\top \;\; y^\top]^\top$ as (x, y). The Euclidean norm of vectors and the associated induced matrix norm is denoted by $|\cdot|$. The distance of a point x to a nonempty set S is denoted by $|x|_S = \inf_{y \in S} |y - x|$. The closure of a set S is denoted by $\mathrm{cl}(S)$. Given a matrix $A \in \mathbb{R}^{n\times n}$, $\mathrm{eig}(A)$ denotes the set of all eigenvalues of A, $\lambda_{\min}(A) := \min\{\lambda/2 : \lambda \in \mathrm{eig}(A + A^\top)\}$, and $\lambda_{\max}(A) := \max\{\lambda/2 : \lambda \in \mathrm{eig}(A + A^\top)\}$. The set of all singular values of A is denoted by $\sigma(A)$, with $\sigma_{\min}(A) := \min\{\sigma(A)\}$ and $\sigma_{\max}(A) := \max\{\sigma(A)\}$.

B. Hybrid dynamical systems

In this paper, a hybrid system $\mathcal{H}$ is modeled as [7]

$$\mathcal{H} = \begin{cases} \;\dot{\xi} = F(\xi) & \xi \in C \\ \xi^+ = G(\xi) & \xi \in D \end{cases} \tag{1}$$

where $\xi \in \mathbb{R}^n$ is the state, $F : C \to \mathbb{R}^n$ is the flow map defining a differential equation capturing the continuous dynamics, and $C \subset \mathbb{R}^n$ defines the flow set on which flows are permitted. The mapping $G : D \to \mathbb{R}^n$ is the jump map defining the law resetting ξ at jumps, and $D \subset \mathbb{R}^n$ is the jump set on which jumps are permitted.

A solution ξ to $\mathcal{H}$ is a *hybrid arc* [7] that is parameterized by $(t, j) \in \mathbb{R}_{\geq 0} \times \mathbb{N}$, where t is the elapsed ordinary time and j is the number of jumps that have occurred. The domain of ξ, denoted by $\mathrm{dom}\,\xi \subset \mathbb{R}_{\geq 0} \times \mathbb{N}$, is a *hybrid time domain*, in the sense that for every $(T, J) \in \mathrm{dom}\,\xi$, there exists a nondecreasing sequence $\{t_j\}_{j=0}^{J+1}$ with $t_0 = 0$ such that $\mathrm{dom}\,\xi \cap ([0, T] \times \{0, 1, \ldots, J\}) = \bigcup_{j=0}^{J} ([t_j, t_{j+1}], \{j\})$. A solution ξ to $\mathcal{H}$ is called *maximal* if it cannot be extended, and is called *complete* if its domain is unbounded. The operations $\sup_t \mathrm{dom}\,\xi$ and $\sup_j \mathrm{dom}\,\xi$ return the supremum of the t and j coordinates, respectively, of points in $\mathrm{dom}\,\xi$. The length of $\mathrm{dom}\,\xi$ is $\sup_t \mathrm{dom}\,\xi + \sup_j \mathrm{dom}\,\xi$.

We employ the following notion of stability [7]:

Definition 1.1: Given a hybrid system $\mathcal{H}$ with data as in (1), a nonempty closed set $\mathcal{A} \subset \mathbb{R}^n$ is said to be *globally pre-exponentially stable*[1] for $\mathcal{H}$ if there exist $\kappa > 0$ and $\lambda > 0$ such that each solution ξ to $\mathcal{H}$ satisfies

$$|\xi(t, j)|_{\mathcal{A}} \leq \kappa \mathrm{e}^{-\lambda(t+j)} |\xi(0, 0)|_{\mathcal{A}} \quad \forall (t, j) \in \mathrm{dom}\,\xi.$$

[1] The term “pre-exponential,” as opposed to “exponential,” indicates the possibility of a maximal solution that is not complete. This allows for separating the conditions for completeness from the conditions for stability and attractivity.

C. Motivational Example

To motivate our proposed parameter estimation algorithm, consider a hybrid arc $\phi : E \to \mathbb{R}^2$ with hybrid time domain

$$E = \bigcup_{k=0}^{\infty} \Big(\big([2\pi k, \quad \pi(2k+2)] \times \{2k\} \big) \cup \big(\{\pi(2k+2)\} \times \{2k+1\} \big) \Big).$$

During flows, the value of ϕ is

$$\phi(t,j) = \begin{cases} [\sin(t) \quad 0]^\top & \text{if} \quad t \le 6\pi \\ [0 \quad 0]^\top & \text{if} \quad t > 6\pi \end{cases} \tag{2}$$

and, each time ϕ jumps, the value of ϕ after the jump is

$$\phi(t,j+1) = \begin{cases} [0.5 \quad 1]^\top & \text{if} \quad j \in \{0,2,4\} \\ [0 \quad 0]^\top & \text{if} \quad j \in \mathbb{N} \setminus \{0,2,4\}. \end{cases} \tag{3}$$

Next, consider the hybrid linear regression model

$$y(t,j) = \theta^\top \phi(t,j) \quad \forall\, (t,j) \in E \tag{4}$$

where $(t,j) \mapsto y(t,j) \in \mathbb{R}$ is known, $(t,j) \mapsto \phi(t,j) \in \mathbb{R}^2$ is known and given by (2), (3), and $\theta = [1 \quad 1]^\top$ is unknown. Suppose our goal is to estimate θ. We first apply the hybrid gradient descent (GD) algorithm in [6], which addresses a similar problem. Denoting the parameter estimate as $\hat{\theta}$, the parameter estimation error for the hybrid GD algorithm, shown in blue in Figure 1, fails to converge to zero since ϕ does not satisfy the hybrid PE condition in [6]. On the other hand, the hybrid CL algorithm proposed in this paper successfully estimates θ by leveraging stored data alongside current measurements, as shown in green in Figure 1.[2]

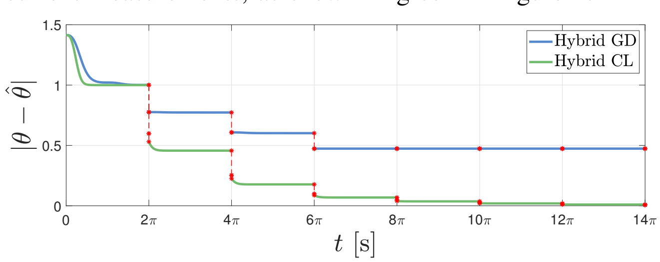

Fig. 1: The projection onto t of the estimation error for the hybrid GD algorithm in [6] and our hybrid CL algorithm. The hybrid GD algorithm produces nonzero steady state error, whereas the error for our algorithm converges to zero.

II. Preliminaries

In preparation for our proposed hybrid CL algorithm, we review the continuous-time and discrete-time CL algorithms.

A. Review of Concurrent Learning

- In continuous time, $t \mapsto y(t)$ is generated by the linear regression model

$$y(t) = \theta^\top \phi(t) \quad \forall\, t \in \mathbb{R}_{\ge 0} \tag{5}$$

where $t \mapsto y(t) \in \mathbb{R}$ and $t \mapsto \phi(t) \in \mathbb{R}^n$ are known, and $\theta \in \mathbb{R}^n$ is a vector of unknown constant parameters.

[2]Code at https://github.com/HybridSystemsLab/HybridCL_Motivation

The continuous-time CL algorithm operates by sampling ϕ and y at time instants $\{\tilde{t}_i\}_{i=1}^{N(t)}$ satisfying $0 \le \tilde{t}_1 < \tilde{t}_2 < \cdots \le t$, where $t \mapsto N(t)$ indicates a time-dependent number of samples, which is to be designed. To store the samples, we define

$$Z(t) := [z_1(t), \quad z_2(t), \quad \cdots, \quad z_p(t)] \in \mathbb{R}^{n \times p}$$
$$Y(t) := [y_1(t), \quad y_2(t), \quad \cdots, \quad y_p(t)] \in \mathbb{R}^{1 \times p}$$

where $p \in \mathbb{N} \setminus \{0\}$ is a design parameter satisfying $p \ge n$. The columns of Z and Y are initially empty (zero), and are populated by the samples of ϕ and y, respectively. In [4], samples are stored in empty columns or, if no empty column is available, by replacing the data in the column that maximizes $\sigma_{\min}(Z)$. Thus, the elements of Z and Y are piecewise constant right-continuous signals that change only at the sample times and, for all $t \in \mathbb{R}_{\ge 0}$,

$$y_\ell(t) = \theta^\top z_\ell(t) \quad \forall\, \ell \in \{1,2,\cdots,p\}. \tag{6}$$

The continuous-time CL algorithm [4] for $\hat{\theta}$ is

$$\begin{aligned} \dot{\hat{\theta}} &= \gamma_c \phi(t)(y(t) - \phi(t)^\top \hat{\theta}) \\ &\quad + \lambda_c \sum_{\ell=1}^{p} z_\ell(t)(y_\ell(t) - z_\ell(t)^\top \hat{\theta}) \end{aligned} \tag{7}$$

where $\gamma_c, \lambda_c > 0$ are design parameters. Denote the parameter estimation error as $\tilde{\theta} := \theta - \hat{\theta}$. Using (6), we express the dynamics of $\tilde{\theta}$ as

$$\dot{\tilde{\theta}} = -\gamma_c \phi(t)\phi(t)^\top \tilde{\theta} - \lambda_c Z(t) Z(t)^\top \tilde{\theta}. \tag{8}$$

The parameter update law for the continuous-time CL algorithm is equivalent to that of the classical continuous-time gradient descent algorithm [3] plus a term associated with the stored data.

- In the discrete-time case, $j \mapsto y(j)$ is generated by the linear regression model

$$y(j) = \theta^\top \phi(j) \quad \forall\, j \in \mathbb{N}.$$

The discrete-time CL algorithm operates similarly to the continuous-time CL algorithm. In [5], samples of ϕ and y are stored in time-varying matrices $j \mapsto Z(j) \in \mathbb{R}^{n \times p}$ and $j \mapsto Y(j) \in \mathbb{R}^{1 \times p}$, respectively, in order to maximize $\sigma_{\min}(Z)/\sigma_{\max}(Z)$. The discrete-time CL algorithm [5] for $\hat{\theta}$ is

$$\begin{aligned} \hat{\theta}(j+1) &= \hat{\theta}(j) + \Gamma \frac{\phi(j)(y(j) - \phi(j)^\top \hat{\theta}(j))}{\gamma_d + |\phi(j)|^2} \\ &\quad + \Gamma \frac{\sum_{\ell=1}^{p} z_\ell(j)(y_\ell(j) - z_\ell(j)^\top \hat{\theta}(j))}{\lambda_d + |Z(j)|^2} \end{aligned} \tag{9}$$

where $\gamma_d, \lambda_d > 0$ and $\Gamma \in (0,1]$ are design parameters. The estimation error $\tilde{\theta}$ has dynamics

$$\begin{aligned} \tilde{\theta}(j+1) &= \tilde{\theta}(j) \\ &\quad - \Gamma \frac{\phi(j)\phi(j)^\top}{\gamma_d + |\phi(j)|^2} \tilde{\theta}(j) - \Gamma \frac{Z(j)Z(j)^\top}{\lambda_d + |Z(j)|^2} \tilde{\theta}(j). \end{aligned} \tag{10}$$

Convergence of $\hat{\theta}$ for the CL algorithms in (7) (resp. (9)) is achieved when $\tilde{\theta}$ in (8) (resp. (10)) converges to zero. Next, we review conditions that ensure $\tilde{\theta}$ converges to zero.

B. Excitation Conditions

Suppose the signal $t \mapsto \phi(t)$ in (5) is *exciting over a finite interval*, i.e., there exist $t^* > 0$ and $\mu_1 > 0$ such that

$$\int_0^{t^*} \phi(s)\phi(s)^\top ds \geq \mu_1 I. \quad (11)$$

The condition in (11) is not sufficient to ensure convergence of $\tilde{\theta}$ to zero for the classical continuous-time gradient descent algorithm [3]. However, if (11) is satisfied, then the sequence of sample times $\{\tilde{t}_i\}_{i=1}^{N(t)}$ can be chosen so that $t \mapsto Z(t)Z(t)^\top$ is uniformly positive definite for all $t \geq t^*$ [4]. Hence, $t \mapsto Z(t)$ is *persistently exciting*, that is,

(C1) there exist $\eta_2, \mu_2 > 0$ such that

$$\int_t^{t+\eta_2} Z(s)Z(s)^\top ds \geq \mu_2 I \quad \forall t \geq t^*$$

and it follows that $\tilde{\theta}$ in (8) converges exponentially to zero.

Similarly, in the discrete-time case, excitation over a finite interval $\{0, 1, \cdots, j^*\}$ with $j^* \in \mathbb{N} \setminus \{0\}$ is not sufficient to ensure convergence of $\tilde{\theta}$ to zero for the classical discrete-time gradient algorithm [3]. However, ϕ and y can be sampled such that $j \mapsto Z(j)$ is *persistently exciting* [5]. That is, when

(C2) there exist $\eta_3 \in \mathbb{N} \setminus \{0\}$ and $\mu_3 > 0$ such that

$$\sum_{i=j}^{j+\eta_3} Z(i)Z(i)^\top \geq \mu_3 I \quad \forall j \geq j^*$$

it follows that $\tilde{\theta}$ in (10) converges exponentially to zero.

III. Hybrid Concurrent Learning Algorithm

Motivated by the limitations of the hybrid gradient descent algorithm highlighted in Section I-C, we develop a hybrid concurrent learning algorithm for estimating unknown parameters of hybrid linear regression systems of the form

$$y(t,j) = \theta^\top \phi(t,j) \quad (12)$$

where the regressor $(t,j) \mapsto \phi(t,j)$ and the output $(t,j) \mapsto y(t,j)$ are hybrid arcs as defined in Section I-B.

Since ϕ in (12) may exhibit both flows and jumps, it is important to update $\hat{\theta}$ continuously whenever ϕ flows, and to update $\hat{\theta}$ discretely each time ϕ jumps, which is possible when jumps are detected instantaneously. Given hybrid arcs $\phi : E \to \mathbb{R}^n$ and $y : E \to \mathbb{R}$, where $E := \operatorname{dom}\phi = \operatorname{dom} y$, our proposed hybrid CL algorithm is implemented as follows.

- During flows, we sample ϕ and y at hybrid time instants $\{(\tilde{t}_i, \tilde{j}_i)\}_{i=1}^{N_c(t,j)}$ satisfying $0 \leq \tilde{t}_1 < \tilde{t}_2 < \cdots \leq t$, where $(t,j) \mapsto N_c(t,j)$ indicates a time-dependent number of samples, which is to be designed. To store the samples, we define, for all $(t,j) \in E$,

$$Z_c(t,j) := \begin{bmatrix} z_1^c(t,j), & z_2^c(t,j), & \cdots, & z_{p_c}^c(t,j) \end{bmatrix} \in \mathbb{R}^{n \times p_c}$$
$$Y_c(t,j) := \begin{bmatrix} y_1^c(t,j), & y_2^c(t,j), & \cdots, & y_{p_c}^c(t,j) \end{bmatrix} \in \mathbb{R}^{1 \times p_c}$$

  where $p_c \in \mathbb{N} \setminus \{0\}$ is a design parameter satisfying $p_c \geq n$. The columns of Z_c and Y_c are initially empty (zero), and are populated by the samples of ϕ and y, respectively. Thus, during flows, the elements of Z_c and Y_c are piecewise constant right-continuous signals, with values changing only at the sample times and, for all $(t,j) \in E$,

$$y_\ell^c(t,j) = \theta^\top z_\ell^c(t,j) \quad \forall\, \ell \in \{1, 2, \cdots, p_c\}. \quad (13)$$

- At jumps, we sample ϕ and y after the jump at hybrid time instants $\{(t_{\tilde{j}_i+1}, \tilde{j}_i+1)\}_{i=1}^{N_d(t,j)}$ satisfying $0 \leq \tilde{j}_1 < \tilde{j}_2 < \cdots \leq j$, where $(t,j) \mapsto N_d(t,j)$ indicates a time-dependent number of samples, which is to be designed. To store the samples, we define, for all $(t,j) \in E$,

$$Z_d(t,j) := \begin{bmatrix} z_1^d(t,j), & z_2^d(t,j), & \cdots, & z_{p_d}^d(t,j) \end{bmatrix} \in \mathbb{R}^{n \times p_d}$$
$$Y_d(t,j) := \begin{bmatrix} y_1^d(t,j), & y_2^d(t,j), & \cdots, & y_{p_d}^d(t,j) \end{bmatrix} \in \mathbb{R}^{1 \times p_d}$$

  where $p_d \in \mathbb{N} \setminus \{0\}$ is a design parameter satisfying $p_d \geq n$. The columns of Z_d and Y_d are initially empty, and are populated by the samples of ϕ and y, respectively. Thus, the elements of Z_d and Y_d change only after jumps associated with the sample times and, for all $(t,j) \in E$,

$$y_\ell^d(t,j) = \theta^\top z_\ell^d(t,j) \quad \forall\, \ell \in \{1, 2, \cdots, p_c\}. \quad (14)$$

For the given ϕ, y, Z_c, Y_c, Z_d, and Y_d, we express the dynamics of $\hat{\theta}$ as a hybrid system, denoted by $\mathcal{H}_{\text{CL}}$, with state $\xi := (\hat{\theta}, \tau, k) \in \mathcal{X} := \mathbb{R}^n \times E$ and dynamics

$$\begin{aligned} \dot{\xi} &= (f(\xi),\, 1,\, 0) =: F_{\text{CL}}(\xi) && \xi \in C_{\text{CL}} \\ \xi^+ &= (g(\xi),\, \tau,\, k+1) =: G_{\text{CL}}(\xi) && \xi \in D_{\text{CL}}. \end{aligned} \quad (15)$$

where the functions f and g, which give the dynamics of $\hat{\theta}$ during flows and jumps, respectively, and the flow set C_{CL} and jump set D_{CL}, are to be defined. The state components τ and k correspond to t and j, respectively, from the hybrid time domain E. Including τ and k in ξ allows ϕ, y, Z_c, Y_c, Z_d, and Y_d to be part of the definitions of F_{CL} and G_{CL}, rather than modeled as inputs to $\mathcal{H}_{\text{CL}}$, so we can express $\mathcal{H}_{\text{CL}}$ as an autonomous hybrid system and leverage recent results on stability properties for such systems [7].

During flows, we update $\hat{\theta}$ continuously with dynamics inspired by the continuous-time CL algorithm in (7), namely,

$$\begin{aligned} \dot{\hat{\theta}} &= \gamma_c \phi(\tau,k)(y(\tau,k) - \phi(\tau,k)^\top \hat{\theta}) \\ &\quad + \lambda_c \sum_{\ell=1}^{p_c} z_\ell^c(\tau,k)(y_\ell^c(\tau,k) - z_\ell^c(\tau,k)^\top \hat{\theta}) =: f(\xi) \end{aligned}$$

where $\gamma_c, \lambda_c > 0$ are design parameters. At jumps, we update $\hat{\theta}$ discretely using a reset map inspired by the discrete-time CL algorithm in (9), namely,

$$\begin{aligned} \hat{\theta}^+ &= \hat{\theta} \\ &\quad + \frac{\Gamma \phi(\tau,k+1)}{\gamma_d + |\phi(\tau,k+1)|^2}\Big(y(\tau,k+1) - \phi(\tau,k+1)^\top \hat{\theta}\Big) \\ &\quad + \frac{\Gamma}{\lambda_d + |Z_d(\tau,k+1)|^2} \sum_{\ell=1}^{p_d} \Big(z_\ell^d(\tau,k+1)\big(y_\ell^d(\tau,k+1) \\ &\quad - z_\ell^d(\tau,k+1)^\top \hat{\theta}\big)\Big) =: g(\xi) \end{aligned}$$

where $\gamma_d, \lambda_d > 0$ and $\Gamma \in (0, 1/2]$ are design parameters. The flow and jump sets of $\mathcal{H}_{\text{CL}}$ are defined so that $\mathcal{H}_{\text{CL}}$ flows when ϕ flows and jumps when ϕ jumps,

$$C_{\text{CL}} := \operatorname{cl}\big(\mathcal{X} \setminus D_{\text{CL}}\big), \quad D_{\text{CL}} := \{\xi \in \mathcal{X} : (\tau, k+1) \in E\}. \quad (16)$$

Remark 3.1: For simplicity, $\mathcal{H}_{\text{CL}}$ in (15) is expressed such that jumps in the estimator state coincide with jumps in ϕ. In practice, since measurements of ϕ^+ and y^+ are not available until after a jump, the corresponding jump in the estimator state will occur at a hybrid time instant after a jump in ϕ.

IV. Stability Analysis

Convergence of $\hat{\theta}$ to θ for $\mathcal{H}_{\text{CL}}$ is achieved when the parameter estimation error $\tilde{\theta} := \theta - \hat{\theta}$ converges to zero. Using (13) and (14), we express $\mathcal{H}_{\text{CL}}$ in error coordinates. The resulting system, denoted by $\widetilde{\mathcal{H}}_{\text{CL}}$, has dynamics

$$\begin{bmatrix}\dot{\tilde{\theta}}\\ \dot{\tau}\\ \dot{k}\end{bmatrix} = \begin{bmatrix}-\Phi_c(\tau,k)\tilde{\theta} - \mathcal{Z}_c(\tau,k)\tilde{\theta}\\ 1\\ 0\end{bmatrix} =: \widetilde{F}_{\text{CL}}(\xi) \qquad \xi \in \widetilde{C}_{\text{CL}}$$
$$\begin{bmatrix}\tilde{\theta}^+\\ \tau^+\\ k^+\end{bmatrix} = \begin{bmatrix}\tilde{\theta} - \Phi_d(\tau,k+1)\tilde{\theta} - \mathcal{Z}_d(\tau,k+1)\tilde{\theta}\\ \tau\\ k+1\end{bmatrix} =: \widetilde{G}_{\text{CL}}(\xi) \quad \xi \in \widetilde{D}_{\text{CL}} \tag{17}$$

where $\widetilde{C}_{\text{CL}} := C_{\text{CL}}$ and $\widetilde{D}_{\text{CL}} := D_{\text{CL}}$, with C_{CL}, D_{CL} in (16), and, omitting the (τ, k) arguments for readability,

$$\Phi_c := \gamma_c \phi\phi^\top, \qquad \Phi_d := \Gamma\frac{\phi\phi^\top}{\gamma_d + |\phi|^2} \tag{18}$$

$$\mathcal{Z}_c := \lambda_c Z_c Z_c^\top, \qquad \mathcal{Z}_d := \Gamma\frac{Z_d Z_d^\top}{\lambda_d + |Z_d|^2}. \tag{19}$$

We impose the following excitation condition.

Assumption 4.1: Given $\Phi_c, \Phi_d, \mathcal{Z}_c, \mathcal{Z}_d : E \to \mathbb{R}^{n\times n}$, where $E := \operatorname{dom}\Phi_c = \operatorname{dom}\Phi_d = \operatorname{dom}\mathcal{Z}_c = \operatorname{dom}\mathcal{Z}_d$ is a hybrid time domain, there exist $\Delta, \mu > 0$ such that, for each $(t', j'), (t^*, j^*) \in E$ satisfying

$$\Delta \le t^* - t' + j^* - j' \le \Delta + 1, \tag{20}$$

the following holds:

$$\begin{aligned}&\sum_{j=j'}^{j^*}\int_{\max\{t',t_j\}}^{\min\{t^*,t_{j+1}\}}\Big(\Phi_c(s,j) + \mathcal{Z}_c(s,j)\Big)ds\\ &+ \frac{1}{2}\sum_{j=j'}^{j^*-1}\Big(\Phi_d(t_{j+1}, j+1) + \mathcal{Z}_d(t_{j+1}, j+1)\Big) \ge \mu I\end{aligned} \tag{21}$$

where $\{t_j\}_{j=0}^J$ is the sequence defining E as in Section I-B and $t_{J+1} := T$, with $J := \sup_j E$ and $T := \sup_t E$.

The hybrid time instants (t', j') and (t^*, j^*) in Assumption 4.1 are the beginning and end, respectively, of a hybrid time interval with length as in (20), over which (21) holds.

Remark 4.2: The excitation condition in Assumption 4.1 relaxes the PE conditions in (C1) and (C2) for the continuous-time and discrete-time CL algorithms, respectively, and relaxes the hybrid PE condition in [6]. Indeed, if $\Phi_c, \Phi_d, \mathcal{Z}_c, \mathcal{Z}_d$ are scalar (i.e., $n = 1$), then Assumption 4.1 implies that either $\mathcal{Z}_c$ and $\mathcal{Z}_d$ are uniformly positive definite or Φ_c and Φ_d are PE in the hybrid sense of [6]. However, in the general case where $n > 1$, it is possible that Assumption 4.1 holds when neither $\mathcal{Z}_c$ nor $\mathcal{Z}_d$ are uniformly positive definite nor Φ_c and Φ_d are PE in the hybrid sense of [6]. An example of the latter case is shown in Section VI-A.

We now establish our main result stating conditions that ensure $\widetilde{\mathcal{H}}_{\text{CL}}$ induces global pre-exponential stability[3] of

$$\mathcal{A} := \left\{\xi \in \mathcal{X} \ : \ \tilde{\theta} = 0\right\}. \tag{22}$$

Global pre-exponential stability of $\mathcal{A}$ implies that, for each solution ξ to $\widetilde{\mathcal{H}}_{\text{CL}}$, the distance from ξ to the set $\mathcal{A}$ is bounded above by an exponentially decreasing function of the initial condition – see Definition 1.1. As a consequence, for each complete solution ξ to $\widetilde{\mathcal{H}}_{\text{CL}}$, the parameter estimation error $\tilde{\theta}$ converges exponentially to zero.

Theorem 4.3: Given $\phi : E \to \mathbb{R}^n$, $Z_c : E \to \mathbb{R}^{n\times p_c}$, and $Z_d : E \to \mathbb{R}^{n\times p_d}$, where $E := \operatorname{dom}\phi = \operatorname{dom}Z_c = \operatorname{dom}Z_d$ is a hybrid time domain, suppose that there exists $\phi_M > 0$ such that $|\phi(t,j)| \le \phi_M$ for all $(t,j) \in E$, and suppose that Assumption 4.1 holds with Φ_c, Φ_d in (18) *and $\mathcal{Z}_c, \mathcal{Z}_d$ in* (19). *Then, for each $\gamma_c, \gamma_d, \lambda_c, \lambda_d > 0$, $\Gamma \in (0, 1/2]$, each solution ξ to the hybrid system $\widetilde{\mathcal{H}}_{\text{CL}}$ in* (17) *satisfies*

$$|\xi(t,j)|_{\mathcal{A}} \le \kappa e^{-\lambda(t+j)}|\xi(0,0)|_{\mathcal{A}} \quad \forall (t,j) \in \operatorname{dom}\xi$$

where

$$\kappa := \sqrt{\frac{1}{1-\alpha}}, \qquad \lambda := \frac{1}{2(\Delta+1)}\ln\left(\frac{1}{1-\alpha}\right),$$

and

$$\alpha := \frac{2\mu}{\left(1 + \sqrt{(a_M+2)(\Delta+2)^3(a_M(\Delta+2)+1/2)}\right)^2} \tag{23}$$

with Δ, μ from Assumption 4.1 and $a_M := (\gamma_c + \lambda_c p_c)\phi_M^2$.

Sketch of Proof: To prove Theorem 4.3, we rely on a stability result in [6], which studies a class of hybrid systems with state $\xi = (\tilde{\theta}, \tau, k) \in \mathcal{X} := \mathbb{R}^n \times E$ and dynamics

$$\begin{aligned}\dot{\xi} &= (-A(\tau,k)\tilde{\theta},\, 1,\, 0) =: F(\xi) & \xi \in C\\ \xi^+ &= (\tilde{\theta} - B(\tau,k)\tilde{\theta},\, \tau,\, k+1) =: G(\xi) & \xi \in D\end{aligned} \tag{24}$$

where $A, B : E \to \mathbb{R}^{n\times n}$ are given and $E := \operatorname{dom}A = \operatorname{dom}B$ is a hybrid time domain, $C := \operatorname{cl}(E \setminus D)$, and $D := \{\xi \in \mathcal{X} : (\tau, k+1) \in E\}$. The hybrid system in (24) reduces to $\widetilde{\mathcal{H}}_{\text{CL}}$ in (17) when, for each solution ξ to $\widetilde{\mathcal{H}}_{\text{CL}}$,

$$A(t,j) := \Phi_c(t,j) + \mathcal{Z}_c(t,j), \tag{25}$$
$$B(t,j) := \begin{cases}\Phi_d(t_{j+1}, j+1) + \mathcal{Z}_d(t_{j+1}, j+1) & \text{if } j < J\\ \Phi_d(T,J) + \mathcal{Z}_d(T,J) & \text{if } j = J\end{cases}$$

for all $(t,j) \in \operatorname{dom}\xi$, where $T := \sup_t \operatorname{dom}\xi$ and $J := \sup_j \operatorname{dom}\xi$. It can be shown that, since Assumption 4.1 holds and $\phi(t,j) \le \phi_M$ for all $(t,j) \in \operatorname{dom}\xi$, with $\phi_M > 0$ coming from Theorem 4.3, the conditions of [6, Theorem 1] are satisfied. Then, from the equivalence between the data of $\widetilde{\mathcal{H}}_{\text{CL}}$ in (17) and the hybrid system in (24) with A, B in (25), the result follows from [6, Theorem 1]. $\square$

[3] Since each solution ξ to $\widetilde{\mathcal{H}}_{\text{CL}}$ inherits the hybrid time domain E, the use of "pre-exponential," as opposed to "exponential," stability means that $\operatorname{dom}E$ may be bounded.

V. Data Recording

In this section, we propose criteria to select data for storage during flows and jumps. Motivated by Theorem 4.3, we store data with the objective of satisfying Assumption 4.1. Moreover, we have from Theorem 4.3 that the rate of convergence of the parameter estimation error increases as μ in (23) increases and, from (21), μ increases as $\lambda_{\min}(\mathcal{Z}_c)$ and $\lambda_{\min}(\mathcal{Z}_d)$ increase. Omitting the (t,j) arguments for readability, we have from (19) that

$$\lambda_{\min}(\mathcal{Z}_c) = \lambda_c \sigma_{\min}(Z_c)^2, \quad \lambda_{\min}(\mathcal{Z}_d) = \Gamma \frac{\sigma_{\min}(Z_d)^2}{\lambda_d + \sigma_{\max}(Z_d)^2}.$$

Thus, maximizing $\lambda_{\min}(\mathcal{Z}_c)$ is equivalent to maximizing $\sigma_{\min}(Z_c)$, and maximizing $\lambda_{\min}(\mathcal{Z}_d)$ is equivalent to maximizing $\sigma_{\min}(Z_d)^2/(\lambda_d + \sigma_{\max}(Z_d)^2)$.

Our proposed data recording scheme is inspired by [5], [8]. However, in contrast to [5], [8], we do not assume that the data can be sampled so that Z_c and Z_d are full rank (see Remark 4.2). Hence, given a measurement of ϕ during flows, we select ϕ for storage based on the following criteria:

1. If Z_c has empty (zero) columns and ϕ is nonzero, then ϕ is stored in an empty column of Z_c.
2. If Z_c is full rank and ϕ increases $\sigma_{\min}(Z_c)$, then ϕ is stored in the column of Z_c that maximizes $\sigma_{\min}(Z_c)$.
3. If Z_c is not full rank and ϕ increases $\mathrm{rank}(Z_c)$, then ϕ is stored in the column of Z_c that maximizes $\mathrm{rank}(Z_c)$.
4. If Z_c is not full rank and ϕ increases the smallest nonzero singular value of Z_c while not decreasing $\mathrm{rank}(Z_c)$, then ϕ is stored in the column of Z_c that maximizes $\min(\sigma(Z_c) \setminus \{0\})$.
5. If none of the items above are satisfied, ϕ is not stored.

Whenever ϕ is stored in a column of Z_c, the current value of the hybrid signal y is stored in the corresponding column of Y_c. We implement this logic using Algorithm 1 during flows. The algorithm for storing measurements at jumps is similar, except with the objective of maximizing $\sigma_{\min}(Z_d)^2/(\lambda_d + \sigma_{\max}(Z_d)^2)$, and is omitted due to space constraints.

Algorithm 1 Data Recording During Flows

Initialize: $\ell = 1$
Require: $(t,j) \in E$ and $(t, j-1) \notin E$
- **if** $\ell \le p_c$ **then** ▷ fill empty columns
  - **if** $|\phi(t,j)| > 0$ **then**
    - Store $\phi(t,j)$ in column ℓ of Z_c
    - Store $y(t,j)$ in column ℓ of Y_c
    - $\ell = \ell + 1$
  - **end if**
- **else** ▷ overwrite stored data
  - $S_{0,\text{old}} = \sigma_{\min}(Z_c)$
  - $S_{1,\text{old}} = \min(\sigma(Z_c) \setminus \{0\})$
  - $R_{\text{old}} = \mathrm{rank}(Z_c)$
  - Create empty vectors S_0 S_1, and R
  - **for** $r = 1$ to p_c **do**
    - $T = Z_c$; Store $\phi(t,j)$ in column r of T
    - Store $\sigma_{\min}(T)$ in column r of S_0
    - Store $\mathrm{rank}(T)$ in column r of R
    - Store $\min(\sigma(T) \setminus \{0\})$ in column r of S_1 if $\mathrm{rank}(T) \ge R_{\text{old}}$ and $\min(\sigma(T) \setminus \{0\}) \ne \varnothing$, otherwise store 0.
  - **end for**
  - **if** $\max S_0 > S_{0,\text{old}}$ **then**
    - Let q denote the column index of $\max S_0$
    - Store $\phi(t,j)$ in column q of Z_c
    - Store $y(t,j)$ in column q of Y_c
  - **else if** $\max R > R_{\text{old}}$ **then**
    - Let q denote the column index of $\max R$
    - Store $\phi(t,j)$ in column q of Z_c
    - Store $y(t,j)$ in column q of Y_c
  - **else if** $\max S_1 > S_{1,\text{old}}$ **then**
    - Let q denote the column index of $\max S_1$
    - Store $\phi(t,j)$ in column q of Z_c
    - Store $y(t,j)$ in column q of Y_c
  - **end if**
- **end if**

VI. Examples

In this section, we present examples that demonstrate the merits of our proposed algorithm. Simulations are performed using the Hybrid Equations Toolbox [9].

A. Motivational Example Revisited

Recall the motivational example in Section I-C. We employ $\mathcal{H}_{\text{CL}}$ in (15) to estimate θ in (4), with $\gamma_c = \lambda_c = 1$, $\gamma_d = \lambda_d = 1$, $\Gamma = 0.5$, and $p_c = p_d = n = 2$.

Let $\Delta = 2\pi + 2$. Then, for each $(t', j'), (t^*, j^*) \in E$ satisfying (20) , if $t' \le 4\pi$ and $j' \le 4$,

$$\sum_{j=j'}^{j^*} \int_{\max\{t',t_j\}}^{\min\{t^*,t_{j+1}\}} \Phi_c(s,j)ds + \frac{1}{2}\sum_{j=j'}^{j^*-1} \Phi_d(t_{j+1},j) \ge \mu_1 I \quad (26)$$

with Φ_c, Φ_d as in (18) and $\mu_1 = 0.11$. However, if $t' > 4\pi$ and $j' > 4$, (26) is satisfied only with $\mu_1 = 0$. Hence, Φ_c and Φ_d do not satisfy the hybrid PE condition in [6].

By sampling ϕ and y as described in Section V, we have that, for all $(t,j) \in E$ satisfying $t > 4\pi$ and $j > 4$,

$$Z_c(t,j) = \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix}, \qquad Z_d(t,j) = \begin{bmatrix} 0.5 & 0.5 \\ 1 & 1 \end{bmatrix}$$

which are not full rank. However, for each $(t', j'), (t^*, j^*) \in E$ satisfying (20) , if $t' > 4\pi$, $j' > 4$,

$$\sum_{j=j'}^{j^*} \int_{\max\{t',t_j\}}^{\min\{t^*,t_{j+1}\}} \mathcal{Z}_c(s,j)ds + \frac{1}{2}\sum_{j=j'}^{j^*-1} \mathcal{Z}_d(t_{j+1},j) \ge \mu_2 I \quad (27)$$

with $\mathcal{Z}_c, \mathcal{Z}_d$ as in (19) and $\mu_2 = 0.14$. Combining (26) and (27), we conclude that Assumption 4.1 is satisfied with $\Delta = 2\pi + 2$ and $\mu = \min\{\mu_1, \mu_2\} = 0.11$, and hence the conditions of Theorem 4.3 hold. Thus, our hybrid CL algorithm induces global pre-exponential stability of $\mathcal{A}$ in (22) in accordance with Theorem 4.3, as shown in Figure 1.

B. Clock Skew Estimation

Consider the problem of estimating the skew between a reference clock and a software clock that is used for timing aperiodic events. During events, the software clock counts ordinary time as $\dot{\tau}_s = 1 + \varepsilon$, where $\tau_s \in \mathbb{R}_{\geq 0}$ is the output of the software clock and $\varepsilon \in \mathbb{R}$ is the unknown clock skew. At the beginning and end of each event, the software clock is reset to zero and, between events, the software clock outputs zero. An event detector indicates resets of the software clock.

The dynamics of the software clock can be written as a hybrid system as in (1) with an added piecewise constant input[4] $u \in \{0, 1\}$, where $u = 1$ during events to be timed, and $u = 0$ otherwise. The software clock has state $(\tau_s, q) \in \mathbb{R}_{\geq 0} \times \{0, 1\}$, where q is a logic variable, and dynamics

$$\begin{aligned} \begin{bmatrix} \dot{\tau}_s \\ \dot{q} \end{bmatrix} &= \begin{bmatrix} (1+\varepsilon)q \\ 0 \end{bmatrix} \qquad (\tau_s, q, u) \in C_s \\ \begin{bmatrix} \tau_s^+ \\ q^+ \end{bmatrix} &= \begin{bmatrix} 0 \\ 1-q \end{bmatrix} \qquad (\tau_s, q, u) \in D_s \end{aligned} \tag{28}$$

where

$$C_s := \{(\tau_s, q, u) \in \mathbb{R}_{\geq 0} \times \{0,1\} \times \{0,1\} : q = u\}$$
$$D_s := \{(\tau_s, q, u) \in \mathbb{R}_{\geq 0} \times \{0,1\} \times \{0,1\} : q \in \{0,1\} \setminus \{u\}\}.$$

We express the τ_s component of each solution to (28) as the output of (12). Given $(t, j) \mapsto \tau_s(t, j)$ generated by (28), where t is provided by the reference clock, we define

$$y(t,j) := \tau_s(t,j), \quad \phi(t,j) := \begin{cases} 0 & \text{if} \quad \tau_s(t,j) = 0 \\ t - t_j & \text{if} \quad \tau_s(t,j) > 0. \end{cases} \tag{29}$$

for all $(t, j) \in \operatorname{dom} \tau_s$, where $\{t_j\}_{j=0}^{\sup_j \operatorname{dom} \tau_s}$ is the sequence defining $\operatorname{dom} \tau_s$ as in Section I-B. Then, y and ϕ in (29) satisfy (12) with $\theta = 1 + \varepsilon$ and, given a parameter estimate $\hat{\theta}$, the software clock skew estimate, denoted by $\hat{\varepsilon}$, is $\hat{\varepsilon} = \hat{\theta} - 1$.

Let $\varepsilon = 0.1$ and suppose that the events to be timed occur in the intervals $t \in [1, 2) \cup [3.5, 4.3)$. With these event times, ϕ in (29) does not satisfy the hybrid PE condition in [6]. However, by using the data recording criteria in Section V, it can be shown that Assumption 4.1 is satisfied and, since ϕ is bounded, the conditions of Theorem 4.3 hold.

We employ $\mathcal{H}_{\text{CL}}$ in (15) to estimate ε, with $\gamma_c = \lambda_c = 1$, $\gamma_d = \lambda_d = 1$, $\Gamma = 0.5$, $p_c = p_d = n = 1$, and $\hat{\theta}(0, 0) = 1$. For comparison, we estimate ε using the hybrid GD algorithm in [6] and, by considering ϕ and y as piecewise continuous signals, using the continuous-time recursive least squares (LS) algorithm [10]. To illustrate the robustness of our algorithm, we also simulate $\mathcal{H}_{\text{CL}}$ with noise $\nu(t, j) = 0.05 \sin(30t)$ added to measurements of ϕ.

The software clock output is shown in the top plot of Figure 2.[5] In the bottom plot, the estimation error is shown for the recursive LS algorithm in purple, for the hybrid GD algorithm in blue, and for our hybrid CL algorithm with and without noise in orange and green, respectively. The estimation errors for the recursive LS algorithm and the hybrid GD algorithm fail to converge to zero since the relevant excitation conditions are not satisfied. In contrast, when no noise is present, the estimation error for our hybrid CL algorithm converges exponentially to zero in accordance with Theorem 4.3 and, when noise is present, the error remains bounded, illustrating the robustness of our algorithm to measurement noise.

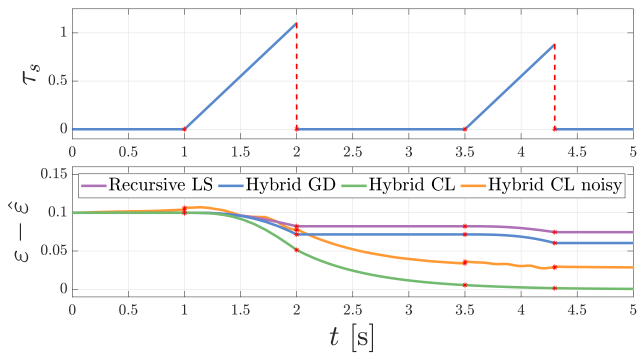

Fig. 2: The projection onto t of the software clock output (top) and the clock skew estimation errors (bottom).

VII. Conclusion

We proposed a hybrid CL algorithm that estimates unknown parameters of hybrid linear regression systems by using stored data alongside current measurements for adaptation. Future work on this topic includes exploring additional applications for our hybrid CL algorithm, formally studying the robustness of our algorithm to measurement noise, and extending the integral CL approach in [11] to estimate unknown parameters for classes of hybrid dynamical systems.

[4] See [7] for details on hybrid systems with inputs.

[5] Code at https://github.com/HybridSystemsLab/HybridCL_ClockSkew

References

[1] H. K. Khalil, *Nonlinear Systems; 3rd ed.* Upper Saddle River, NJ: Prentice-Hall, 2002.

[2] G. Tao, *Adaptive Control Design and Analysis (Adaptive and Learning Systems for Signal Processing, Communications and Control Series)*. USA: John Wiley & Sons, Inc., 2003.

[3] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems*. USA: Prentice-Hall, Inc., 1989.

[4] G. Chowdhary and E. Johnson, "Concurrent learning for convergence in adaptive control without persistency of excitation," in *Proceedings of the 2010 IEEE Conference on Decision and Control (CDC)*, pp. 3674–3679.

[5] O. Djaneye-Boundjou and R. Ordóñez, "Parameter identification in structured discrete-time uncertainties without persistency of excitation," in *Proceedings of the 2015 European Control Conference (ECC)*, pp. 3149–3154.

[6] A. Saoud, M. Maghenem, and R. G. Sanfelice, "A Hybrid Gradient Algorithm for Linear Regression with Hybrid Signals," in *Proceedings of the 2021 American Control Conference (ACC)*, pp. 4997–5002.

[7] R. G. Sanfelice, *Hybrid Feedback Control*. New Jersey: Princeton University Press, 2021.

[8] G. Chowdhary, T. Yucelen, M. Mühlegg, and E. Johnson, "Concurrent Learning Adaptive Control of Linear Systems with Exponentially Convergent Bounds," *International Journal of Adaptive Control and Signal Processing*, vol. 27, no. 4, pp. 280–301, Apr. 2013.

[9] R. G. Sanfelice, D. A. Copp, and P. Nanez, "A toolbox for simulation of hybrid systems in Matlab/Simulink: Hybrid Equations (HyEQ) Toolbox," in *Proceedings of the 2013 Hybrid Systems: Computation and Control Conference*, p. 101–106.

[10] P. A. Ioannou, *Robust Adaptive Control*. Upper Saddle River, NJ: PTR Prentice-Hall, 1996.

[11] A. Parikh, R. Kamalapurkar, and W. E. Dixon, "Integral concurrent learning: Adaptive control with parameter convergence using finite excitation," *International Journal of Adaptive Control and Signal Processing*, vol. 33, no. 12, pp. 1775–1787, 2019.