Implicit Regularization and Momentum Algorithms in Nonlinearly Parameterized Adaptive Control and Prediction

ARTICLE **Communicated by Andre Wibisono**

Implicit Regularization and Momentum Algorithms in Nonlinearly Parameterized Adaptive Control and Prediction

Nicholas M. Boffi
boffi@g.harvard.edu
John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138, U.S.A.

Jean-Jacques E. Slotine
jjs@mit.edu
Nonlinear Systems Laboratory, MIT, Cambridge, MA 02139, U.S.A.

Stable concurrent learning and control of dynamical systems is the subject of adaptive control. Despite being an established field with many practical applications and a rich theory, much of the development in adaptive control for nonlinear systems revolves around a few key algorithms. By exploiting strong connections between classical adaptive nonlinear control techniques and recent progress in optimization and machine learning, we show that there exists considerable untapped potential in algorithm development for both adaptive nonlinear control and adaptive dynamics prediction. We begin by introducing first-order adaptation laws inspired by natural gradient descent and mirror descent. We prove that when there are multiple dynamics consistent with the data, these non-Euclidean adaptation laws implicitly regularize the learned model. Local geometry imposed during learning thus may be used to select parameter vectors—out of the many that will achieve perfect tracking or prediction—for desired properties such as sparsity. We apply this result to regularized dynamics predictor and observer design, and as concrete examples, we consider Hamiltonian systems, Lagrangian systems, and recurrent neural networks. We subsequently develop a variational formalism based on the Bregman Lagrangian. We show that its Euler Lagrange equations lead to natural gradient and mirror descent-like adaptation laws with momentum, and we recover their first-order analogues in the infinite friction limit. We illustrate our analyses with simulations demonstrating our theoretical results.

1 Introduction

Adaptation is an online learning problem concerned with control or prediction of the dynamics of an unknown nonlinear system. This task is

Neural Computation **33, 590–673** (2021)
https://doi.org/10.1162/neco_a_01360

accomplished by constructing an approximation to the true dynamics through the online adjustment of a vector of parameter estimates under the assumption that there exists a fixed vector of parameters that globally fits the dynamics. The overarching goal is provably safe, stable, and concurrent learning and control of nonlinear dynamical systems.

Adaptive control theory is a mature field, and many results exist tailored to specific system structures (Ioannou & Sun, 2012; Narendra & Annaswamy, 2005; Slotine & Li, 1991). An adaptive control algorithm typically consists of a parameter estimator coupled in feedback to the controlled system, and the estimator is often strongly inspired by gradient-based optimization algorithms. A significant difference between standard optimization algorithms and adaptive control algorithms is that the parameter estimator must not only converge to a set of parameters that leads to perfect tracking of the desired trajectory, but the system must remain stable throughout the process. The additional requirement of stability prevents the immediate application of optimization algorithms as adaptive control algorithms, and stability must be proved by jointly analyzing the closed-loop system and estimator.

Significant progress has been made in adaptive control even for nonlinear systems in the *linearly parameterized* setting, where the dynamics approximation is of the form $\hat{\mathbf{f}} = \mathbf{Y}(\mathbf{x}, t)\hat{\mathbf{a}}$ for some known regressor matrix $\mathbf{Y}(\mathbf{x}, t)$ and vector of parameter estimates $\hat{\mathbf{a}}(t)$. Examples include the adaptive robot trajectory controller of Slotine and Li (1987) and the neural network-based controller of Sanner and Slotine (1992), which employs a mathematical expansion in physical nonlinear basis functions to uniformly approximate the unknown dynamics.

Unlike its linear counterpart, solutions to the adaptive control problem in the general nonlinearly parameterized setting $\hat{\mathbf{f}} = \mathbf{f}(\mathbf{x}, \hat{\mathbf{a}}, t)$ have remained elusive. Intuitively, this is unsurprising: guarantees for gradient-based optimization algorithms typically rely on convexity, with a few notable exceptions such as the Polyak-Lojasiewicz condition (Polyak, 1963). In the linearly parameterized setting, the underlying optimization problem will be convex. When the parameters appear nonlinearly, the problem is in general nonconvex and difficult to provide guarantees for.

In this work, we provide new provably globally convergent algorithms for both the linearly and nonlinearly parameterized adaptive control problems, along with new insight into existing adaptive control algorithms for the linearly parameterized setting. Our results for nonlinearly parameterized systems are valid under the monotonicity assumptions of Tyukin, Prokhorov, and van Leeuwen (2007) and the convexity assumptions of Fradkov (1980). These monotonicity assumptions are equivalent to those commonly satisfied by generalized linear models in statistics (Kakade, Kalai, Kanade, & Shamir, 2011).

1.1 Description of Primary Contributions. Our contributions can be categorized into two main advances.

1. We further develop a class of natural gradient and mirror descent-like algorithms that have recently appeared in the literature in the context of physically consistent inertial parameter learning in robotics (Lee, Kwon, & Park, 2018) and geodesically convex optimization (Wensing & Slotine, 2020). We prove that these algorithms implicitly regularize the learned system model in both the linearly parameterized and nonlinearly parameterized settings.
2. We construct a general class of higher-order in-time adaptive control algorithms that incorporate momentum into existing adaptation laws. We prove that our new momentum algorithms are stable and globally convergent for both linearly parameterized and nonlinearly parameterized systems.

Unlike standard problems in optimization and machine learning, explicit regularization terms cannot be naively added to adaptive control algorithms without affecting stability and performance. Our approach enables a provably stable and globally convergent implementation of regularization in adaptive control. We demonstrate the utility of these results through examples in the context of dynamics prediction, such as sparse estimation of a physical system's Hamiltonian or Lagrangian function, and estimating the weights of a continuous-time recurrent neural network model.

It is well known in adaptive control that the true parameters are only recovered when the desired trajectory satisfies a strong condition known as *persistent excitation* (Narendra & Annaswamy, 2005; Slotine & Li, 1991). In general, an adaptation law need only find parameters that enable perfect tracking, and very little is known about what parameters are found when the estimator converges without persistent excitation. Our proof of implicit regularization provides an answer and shows that standard Euclidean adaptation laws lead to parameters of minimum l_2 norm.

For the second contribution, we utilize the Bregman Lagrangian (Betancourt, Jordan, & Wilson, 2018; Wibisono, Wilson, & Jordan, 2016; Wilson, Recht, & Jordan, 2016) in tandem with the velocity gradient methodology (Andrievskii, Stotskii, & Fradkov, 1988; Fradkov, 1980, 1986; Fradkov, Miroshnik, & Nikiforov, 1999) to define a general formalism that generates higher-order in-time (Morse, 1992) velocity gradient algorithms. Our key insight is that the velocity gradient formalism provides an optimization-like framework that encompasses many well-known adaptive control algorithms and that the velocity gradient "loss function" can be placed directly in the Bregman Lagrangian.

1.2 Summary of Related Work. Our work continues in a long-standing tradition that utilizes a continuous-time view to analyze optimization

algorithms, and here we consider a nonexhaustive list. Diakonikolas and Jordan (2019) develop momentum algorithms from the perspective of Hamiltonian dynamics, while Maddison, Paulin, Teh, O'Donoghue, and Doucet (2018) use Hamiltonian dynamics to prove linear convergence of new optimization algorithms without strong convexity. Muehlebach and Jordan (2019, 2020) study momentum algorithms from the viewpoint of dynamical systems and control. Boffi and Slotine (2020) analyze distributed stochastic gradient descent algorithms via dynamical systems and nonlinear contraction theory. Su, Boyd, and Candès (2016) provide an intuitive justification for Nesterov's accelerated gradient method (Nesterov, 1983) through a limiting differential equation. Continuous-time differential equations were used as early as 1964 by Polyak to derive the classical momentum or "heavy ball" optimization method (Polyak, 1964). In all cases, continuous time often affords simpler proofs, and it enables the application of physical intuition when reasoning about optimization algorithms. Given the gradient-based nature of many adaptive control algorithms, the continuous-time view of optimization provides a natural bridge from modern optimization to modern adaptive control.

Despite the simplicity of proofs in continuous time, finding a discretization that provably retains the convergence rates of a given differential equation is challenging. In a significant advance, Wibisono et al. (2016) showed that many accelerated methods in optimization can be derived via a variational point of view from a single mathematical object known as the Bregman Lagrangian. The Bregman Lagrangian leads to second-order mass-spring-damper-like dynamics, and careful discretization provides discrete-time algorithms such as Nesterov's celebrated accelerated gradient method (Nesterov, 1983). We similarly use the Bregman Lagrangian to generate our new adaptive control algorithms, which generalize and extend a recently developed algorithm due to Gaudio, Gibson, Annaswamy, and Bolender (2019).

Progress has been made in nonlinearly parameterized adaptive control in a number specific cases. Annaswamy, Skantze, and Loh (1998), Ai-Poh Loh, Annaswamy, and Skantze (1999), and Kojić and Annaswamy (2002) develop stable adaptive control laws for convex and concave parameterizations, though they may be overly conservative and require solving optimization problems at each time step. Astolfi and Ortega (2003) and Liu, Ortega, Su, and Chu (2010) develop the immersion and invariance (I&I) approach, and prove global convergence if a certain monotone function can be constructed. Ortega, Gromov, Nuño, Pyrkin, and Romero (2019) use a similar approach for system identification. Tyukin et al. (2007) consider dynamical systems satisfying a monotonicity assumption that is essentially identical to conditions required for learning generalized linear models in machine learning and statistics (Goel & Klivans, 2017; Goel, Klivans, & Meka, 2018; Kakade et al., 2011), and develop provably stable adaptive control algorithms for nonlinearly parameterized systems in this setting.

Fradkov (1980), Andrievskii et al. (1988), Fradkov (1986), and Fradkov et al. (1999) develop the velocity gradient methodology, an optimization-like framework for adaptive control that allows for provably global convergence under a convexity assumption. As mentioned in section 1.1, this framework, in tandem with the Bregman Lagrangian, is central to our development of momentum algorithms.

Our work is strongly related to and inspired by a line of recent work that analyzes the implicit bias of optimization algorithms in machine learning. Soudry, Hoffer, Nacson, Gunasekar, and Srebro (2018) and Gunasekar, Lee, Soudry, and Srebro (2018b, 2018a) characterize implicit regularization of common gradient-based optimization algorithms such as gradient descent with and without momentum, as well as natural gradient descent and mirror descent in the settings of regression and classification. Azizan, Lale, and Hassibi (2019) and Azizan and Hassibi (2019) arrive at similar results via a different derivation based on results from $\mathcal{H}_\infty$ control. Similarly, Belkin, Hsu, Ma, and Mandal (2019) consider the importance of implicit regularization in the context of the successes of deep learning. Our results are the adaptive control analogues of those presented in these papers.

1.3 Paper Outline. The paper is organized as follows. In section 2, we present some required mathematical background on direct adaptive control in the linearly and nonlinearly parameterized settings. In section 3 we analyze the implicit bias of adaptive control algorithms, while in section 4 we consider general observer and dynamics predictor design, Hamiltonian dynamics prediction, control of Lagrangian systems, and estimation of recurrent neural networks. In section 5 we provide background for our development of momentum algorithms, including a review of the velocity gradient formalism (section 5.1) and the Bregman Lagrangian (section 5.2). In section 6 we present adaptive control algorithms with momentum, and we extend them to the non-Euclidean setting in section 7. We illustrate our results via simulation in section 8, and we conclude with some closing remarks and future directions in section 9.

2 Direct Adaptive Control

In this section, we provide an introduction to direct adaptive control for both linearly parameterized and nonlinearly parameterized systems, along with a description of some natural gradient-like adaptive laws that have appeared in the recent literature.

2.1 Linearly Parameterized Dynamics. For simplicity, we restrict ourselves to the class of nth-order nonlinear systems,

$$x^{(n)} + f(\mathbf{x}, \mathbf{a}, t) = u \tag{2.1}$$

where $x^{(i)} \in \mathbb{R}$ denotes the ith derivative of x, $\mathbf{x} = \left(x, x^{(1)}, \ldots, x^{(n-1)}\right)^{\mathrm{T}} \in \mathbb{R}^n$ is the system state, $\mathbf{a} \in \mathbb{R}^p$ is a vector of unknown parameters, $f : \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}_+ \to \mathbb{R}$ is of known functional form but is unknown due to its dependence on $\mathbf{a}$, and $u \in \mathbb{R}$ is the control input. We seek to design a feedback control law $u = u(\mathbf{x}, \hat{\mathbf{a}})$ that depends on a set of adjustable parameters $\hat{\mathbf{a}} \in \mathbb{R}^p$ and ensures that $\mathbf{x}(t) \to \mathbf{x}_d(t)$ where $\mathbf{x}_d(t) \in \mathbb{R}^n$ is a known desired trajectory. Along the way, we require that all system signals remain bounded. The estimated parameters $\hat{\mathbf{a}}$ are updated according to a learning rule or adaptation law,

$$\dot{\hat{\mathbf{a}}} = \mathbf{g}(\mathbf{a}, \hat{\mathbf{a}}, \mathbf{x}), \tag{2.2}$$

where $\mathbf{g} : \mathbb{R}^p \times \mathbb{R}^p \times \mathbb{R}^n \to \mathbb{R}^p$ must be implementable solely in terms of known system signals despite its potential dependence on $\mathbf{a}$. For nth-order systems as considered in equation 2.1, a common approach is to define the *sliding variable* (Slotine & Li, 1991),

$$s = \left(\frac{d}{dt} + \lambda\right)^{n-1} \tilde{x} = \tilde{x}^{(n-1)} - \tilde{x}_r^{(n-1)}, \tag{2.3}$$

where $\lambda > 0$ is a constant and $\tilde{x}(t) = x(t) - x_d(t)$. We have defined $\tilde{x}^{(i)}(t) = x^{(i)}(t) - x_d^{(i)}(t)$ and $\tilde{x}_r^{(n-1)}$ as the remainder based on the definition of s. According to the definition, equation 2.3, s obeys the differential equation,

$$\dot{s} = u - f(\mathbf{x}, \mathbf{a}, t) - \tilde{x}_r^{(n)}. \tag{2.4}$$

Hence, from equation 2.4, we may choose

$$u = f(\mathbf{x}, \hat{\mathbf{a}}, t) + \tilde{x}_r^{(n)} - \eta s \tag{2.5}$$

to obtain the stable first-order linear filter:

$$\dot{s} = -\eta s + f(\mathbf{x}, \hat{\mathbf{a}}, t) - f(\mathbf{x}, \mathbf{a}, t). \tag{2.6}$$

For future convenience, we define $\tilde{f}(\mathbf{x}, \hat{\mathbf{a}}, \mathbf{a}, t) = f(\mathbf{x}, \hat{\mathbf{a}}, t) - f(\mathbf{x}, \mathbf{a}, t)$ and will omit its arguments when clear from the context. From the definition of s in equation 2.3, $s = 0$ defines the *dynamics*

$$\left(\frac{d}{dt} + \lambda\right)^{n-1} \tilde{x} = 0. \tag{2.7}$$

Equation 2.7 is a stable $(n-1)$th-order filter which ensures that $\tilde{x} \to 0$ exponentially. For systems of the form in equation 2.1, it is thus sufficient to consider the two first-order dynamics, equations 2.2 and 2.6. The adaptive

control problem has thus been reduced to finding a learning algorithm that ensures $s \to 0$.

Remark 1. Systems in the matched uncertainty form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}\left(u - f(\mathbf{x}, \mathbf{a}, t)\right),$$

where the constant pair $(\mathbf{A}, \mathbf{b})$ is controllable and the constant parameter vector $\mathbf{a}$ in the nonlinear function $f(\mathbf{x}, \mathbf{a}, t)$ is unknown, can always be put in the form of equation 2.1 by using a state transformation to the second controllability canonical form (see Luenberger, 1979, chap. 8.8). After such a transformation, the new state variables $\mathbf{z}$ satisfy $\dot{z}_i = z_{i+1}$ for $i < n$ and $\dot{z}_n = -\sum_{i=1}^{n-1} a_i z_i + u - f(\mathbf{x}, \mathbf{a}, t)$ for some fixed constants c_i. Defining s as in equation 2.3 and choosing u accordingly leads to equation 2.6. Hence, all results in this article extend immediately to such systems.

Remark 2. The fundamental utility of defining the variable s is its conversion of the adaptive control problem for the nth-order system, equation 2.1, to an adaptive control problem for the first-order system, equation 2.6. Our results may be simply extended to other error models (Ai-Poh Loh et al., 1999; Narendra & Annaswamy, 2005) of the form 2.6, or error models with similar input-output guarantees, as summarized by lemma 2.

Remark 3. We will use $\mathbf{f}$ to denote the equivalent first-order system to (2.1), $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{a}, t) + \mathbf{u}$, where $\mathbf{f} = (x_2, x_3, \ldots, f(\mathbf{x}, \mathbf{a}, t))$ and $\mathbf{u} = (0, 0, \ldots, u)$.

The classic setting for adaptive control assumes that the unknown nonlinear dynamics depends linearly on the set of unknown parameters, that is,

$$f(\mathbf{x}, \mathbf{a}, t) = \mathbf{Y}(\mathbf{x}, t)\mathbf{a},$$

with $\mathbf{Y} : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}^{1\times p}$ a known function. In this setting, a well-known algorithm is the adaptive controller of Slotine and Coetsee (1986), given by

$$\dot{\hat{\mathbf{a}}} = -\mathbf{P}\mathbf{Y}^{\mathrm{T}} s, \tag{2.8}$$

and its extension to multi-input adaptive robot control (Slotine & Li, 1987), where $\mathbf{P} = \mathbf{P}^{\mathrm{T}} > 0 \in \mathbb{R}^{p\times p}$ is a constant positive-definite matrix of learning rates. Consideration of the Lyapunov-like function $V = \frac{1}{2}s^2 + \frac{1}{2}\tilde{\mathbf{a}}^{\mathrm{T}}\mathbf{P}^{-1}\tilde{\mathbf{a}}$ shows stability of the feedback interconnection of equations 2.6 and 2.8 and convergence to the desired trajectory via an application of Barbalat's lemma (see lemma A.1). We will refer to equation 2.8 as the Slotine and Li controller.

In this work, we make a mild additional assumption that simplifies some of the proofs.

Assumption 1. *The dynamics $\hat{f}(\mathbf{x}, \hat{\mathbf{a}}, t)$ is locally bounded in $\hat{\mathbf{x}}$ and $\hat{\mathbf{a}}$ uniformly in t. That is, if $\|\mathbf{x}\| \leq \infty$ and $\|\hat{\mathbf{a}}\| < \infty$, then $\forall t \geq 0$, $|\hat{f}(\mathbf{x}, \hat{\mathbf{a}}, t)| < \infty$.*

2.2 Nonlinearly Parameterized Dynamics. While a difficult problem in general, significant progress has been made for the nonlinearly parameterized adaptive control problem under the assumption of *monotonicity*, and several notions of monotonicity have appeared in the literature (Astolfi & Ortega, 2003; Liu et al., 2010; Ortega et al., 2019; Tyukin, 2011; Tyukin et al., 2007). We consider one such notion as presented by Tyukin et al. (2007), which is captured in the following assumption.

Assumption 2. *There exists a known time- and state-dependent function* $\boldsymbol{\alpha} : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{R}^p$ *such that*

$$\tilde{\mathbf{a}}^{\mathrm{T}} \boldsymbol{\alpha}(\mathbf{x}, t) \left(f(\mathbf{x}, \hat{\mathbf{a}}, t) - f(\mathbf{x}, \mathbf{a}, t) \right) \geq 0, \tag{2.9}$$

$$|\boldsymbol{\alpha}(\mathbf{x}, t)^{\mathrm{T}} \tilde{\mathbf{a}}| \geq \frac{1}{D_1} |f(\mathbf{x}, \hat{\mathbf{a}}, t) - f(\mathbf{x}, \mathbf{a}, t)|, \tag{2.10}$$

where $D_1 > 0$ *is a positive scalar.*

This assumption is satisfied, for example, by all functions of the form

$$f(\mathbf{x}, \mathbf{a}, t) = \lambda(\mathbf{x}, t) f_m(\mathbf{x}, \boldsymbol{\phi}(\mathbf{x}, t)^{\mathrm{T}} \mathbf{a}, t), \tag{2.11}$$

where $\lambda : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{R}$, $\boldsymbol{\phi} : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{R}^p$, $f_m : \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}_{\geq 0} \to \mathbb{R}$, and where f_m is monotonic and Lipschitz in $\boldsymbol{\phi}(\mathbf{x}, t)^{\mathrm{T}} \mathbf{a}$. In this setting, $\boldsymbol{\alpha}(\mathbf{x}, t)$ may be taken as $\boldsymbol{\alpha}(\mathbf{x}, t) = (-1)^p D_1 \lambda(\mathbf{x}, t) \boldsymbol{\phi}(\mathbf{x}, t)$, where $p = 0$ if f_m is nondecreasing in $\boldsymbol{\phi}^{\mathrm{T}} \mathbf{a}$ and $p = 1$ if f_m is nonincreasing in $\boldsymbol{\phi}^{\mathrm{T}} \mathbf{a}$ (Tyukin, 2011; Tyukin et al., 2007).

Under assumption 2, Tyukin et al. (2007) showed that the adaptation law,

$$\dot{\hat{\mathbf{a}}} = -\tilde{f}(\mathbf{x}, \hat{\mathbf{a}}, \mathbf{a}, t) \mathbf{P} \boldsymbol{\alpha}(\mathbf{x}, t), \tag{2.12}$$

with $\mathbf{P} = \mathbf{P}^{\mathrm{T}} > 0$ a positive-definite matrix of learning rates of appropriate dimensions ensures that $\tilde{f} \in \mathcal{L}_2$ over the maximal interval of existence of $\mathbf{x}$. Under suitable conditions on the error model, this then ensures that $\tilde{f} \in \mathcal{L}_2 \cap \mathcal{L}_\infty$, $\mathbf{x}(t)$ and $\hat{\mathbf{a}}(t)$ both remain bounded for all t, and that $\mathbf{x} \to \mathbf{x}_d$. The proof follows by consideration of the Lyapunov-like function $V = \frac{1}{2} \tilde{\mathbf{a}}^{\mathrm{T}} \mathbf{P}^{-1} \tilde{\mathbf{a}}$.

While $\tilde{f}$ itself is unknown, and hence equation 2.12 is not directly implementable, it is contained in $\dot{s}$. Intuitively, unknown quantities contained in $\dot{s}$ can be obtained in the adaptation dynamics through a proportional term in $\hat{\mathbf{a}}$ that contains s. This idea of gaining a "free" derivative is the basis of the reduced-order Luenberger observer for linear systems (Luenberger, 1979).[1] Proportional-integral adaptive laws of this type have been known as

[1] Similar concepts can be extended to nonlinear observers; see Lohmiller and Slotine (1998, sec. 4.1).

algorithms in finite form (Fradkov et al., 1999; Tyukin, 2003) and appear in the well-known I&I framework (Astolfi & Ortega, 2003; Liu et al., 2010). Following this prescription, equation 2.12 may be implemented in a proportional-integral form,

$$\boldsymbol{\xi}(\mathbf{x}, t) = -\mathbf{P}s(\mathbf{x}, t)\boldsymbol{\alpha}(\mathbf{x}, t), \tag{2.13}$$

$$\boldsymbol{\rho}(\mathbf{x}, t) = \mathbf{P}\int_{x_n(t_0)}^{x_n(t)} s(\mathbf{x}, t)\frac{\partial \boldsymbol{\alpha}(\mathbf{x}, t)}{\partial x_n} dx_n, \tag{2.14}$$

$$\hat{\mathbf{a}} = \overline{\mathbf{a}} + \boldsymbol{\xi}(\mathbf{x}, t) + \boldsymbol{\rho}(\mathbf{x}, t), \tag{2.15}$$

$$\dot{\tilde{\mathbf{a}}} = -\eta s\mathbf{P}\boldsymbol{\alpha} + \mathbf{P}s\sum_{i=1}^{n-1}\frac{\partial \boldsymbol{\alpha}}{\partial x_i}x_{i+1} - \sum_{i=1}^{n-1}\frac{\partial \boldsymbol{\rho}}{\partial x_i}x_{i+1} - \left(\frac{\partial \boldsymbol{\rho}}{\partial \mathbf{x}_d}\right)^{\mathrm{T}}\dot{\mathbf{x}}_d - \frac{\partial \boldsymbol{\xi}}{\partial t} - \frac{\partial \boldsymbol{\rho}}{\partial t}. \tag{2.16}$$

Algorithm 2.12 is similar to a gradient flow algorithm. If $f(\mathbf{x}, \mathbf{a}, t)$ has the form in equation 2.11 and is nondecreasing, gradient flow on the loss function $L(\mathbf{x}, \hat{\mathbf{a}}, \mathbf{a}, t) = \frac{1}{2}\tilde{f}^2(\mathbf{x}, \hat{\mathbf{a}}, \mathbf{a}, t)$ with a gain matrix $D_1\mathbf{P}$ leads to

$$\dot{\hat{\mathbf{a}}} = -\tilde{f}(\mathbf{x}, \hat{\mathbf{a}}, \mathbf{a}, t)f_m'(\mathbf{x}, \boldsymbol{\phi}^{\mathrm{T}}\hat{\mathbf{a}}, t)\mathbf{P}\boldsymbol{\alpha}(\mathbf{x}, t),$$

where $'$ denotes differentiation with respect to the second argument. $f_m'(\mathbf{x}, \boldsymbol{\phi}^{\mathrm{T}}\hat{\mathbf{a}}, t)$ is of known sign due to the monotonicity assumption but of unknown magnitude. It is sufficient to remove this quantity from the adaptation law and instead to follow the *pseudogradient* $\tilde{f}(\mathbf{x}, \hat{\mathbf{a}}, \mathbf{a}, t)\boldsymbol{\alpha}(\mathbf{x}, t)$ despite nonconvexity of the square loss in this setting. Similarly, if f is nonincreasing, we find

$$\dot{\hat{\mathbf{a}}} = \tilde{f}(\mathbf{x}, \hat{\mathbf{a}}, \mathbf{a}, t)f_m'(\mathbf{x}, \boldsymbol{\phi}^{\mathrm{T}}\hat{\mathbf{a}}, t)\mathbf{P}\boldsymbol{\alpha}(\mathbf{x}, t),$$

and it is sufficient to set f_m' to negative one.

2.3 The Bregman Divergence and Natural Adaptation Laws. Lee et al. (2018) introduced an elegant modification of the Slotine and Li adaptive robot controller, later generalized by Wensing and Slotine (2020). It consists of replacing the usual parameter estimation error term $\frac{1}{2}\tilde{\mathbf{a}}^{\mathrm{T}}\mathbf{P}^{-1}\tilde{\mathbf{a}}$ in the Lyapunov-like function $V = \frac{1}{2}s^2 + \frac{1}{2}\tilde{\mathbf{a}}^{\mathrm{T}}\mathbf{P}^{-1}\tilde{\mathbf{a}}$ with the Bregman divergence (Bregman, 1967),

$$d_\psi(\mathbf{y} \,\|\, \mathbf{x}) = \psi(\mathbf{y}) - \psi(\mathbf{x}) - (\mathbf{y} - \mathbf{x})^{\mathrm{T}}\nabla\psi(\mathbf{x})$$

to obtain the new "non-Euclidean" Lyapunov-like function,

$$V = \frac{1}{2}s^2 + d_\psi(\mathbf{a} \,\|\, \hat{\mathbf{a}}), \tag{2.17}$$

for an arbitrary strongly convex function ψ.

The Bregman divergence may be understand as the error made when approximating $\psi(\mathbf{y})$ by a first-order Taylor expansion around $\mathbf{x}$. It is guaranteed to be nonnegative for strongly convex functions by the first-order characterization of convexity. While it is not a norm in general, it defines a distance-like function for ψ strongly convex related to the *Hessian metric* $\frac{1}{2}\|\mathbf{x}\|^2_{\nabla^2\psi} = \frac{1}{2}\mathbf{x}^\mathrm{T}\nabla^2\psi(\mathbf{x})\mathbf{x}$. As two simple examples, for $\psi(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|^2$, $d_\psi(\mathbf{x} \,\|\, \mathbf{y}) = \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|^2$. For $\psi(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\mathrm{T}\mathbf{Q}\mathbf{x}$ with $\mathbf{Q} > 0$ a positive-definite matrix, $d_\psi(\mathbf{x} \,\|\, \mathbf{y}) = \frac{1}{2}(\mathbf{x} - \mathbf{y})^\mathrm{T}\mathbf{Q}(\mathbf{x} - \mathbf{y})$. For general convex functions, $d_\psi(\cdot \,\|\, \cdot)$ can always be written via Taylor's formula with an integral remainder for multivariate functions as

$$d_\psi(\mathbf{y} \,\|\, \mathbf{x}) = (\mathbf{y} - \mathbf{x})^\mathrm{T}\left(\int_0^1 \nabla^2\psi\left(\mathbf{x} + s(\mathbf{y} - \mathbf{x})\right)(1 - s)ds\right)(\mathbf{y} - \mathbf{x}).$$

Indeed, a quick calculation shows that the derivative of the Bregman divergence is simply

$$\frac{d}{dt}d_\psi(\mathbf{a} \,\|\, \hat{\mathbf{a}}) = \tilde{\mathbf{a}}^\mathrm{T}\nabla^2\psi(\hat{\mathbf{a}})\dot{\hat{\mathbf{a}}}, \tag{2.18}$$

which can be directly used to show stability of the adaptation law

$$\dot{\hat{\mathbf{a}}} = -\left[\nabla^2\psi(\hat{\mathbf{a}})\right]^{-1}\mathbf{Y}^\mathrm{T}s.$$

This procedure replaces the gain matrix $\mathbf{P}$ in the adaptation law by the $\hat{\mathbf{a}}$-dependent *inverse Hessian* $\left[\nabla^2\psi(\hat{\mathbf{a}})\right]^{-1}$ of the strongly convex function ψ. In essence, this amounts to the adaptive control equivalent of the *natural gradient* algorithm of Amari (1998), so that the resulting adaptation law respects the underlying Riemannian geometry captured by the Hessian metric $\nabla^2\psi(\hat{\mathbf{a}})$. The standard adaptation law $\dot{\hat{\mathbf{a}}} = -\mathbf{P}\mathbf{Y}^\mathrm{T}s$ uses the constant metric $\mathbf{P}^{-1}$, which in turn explains the appearance of $\mathbf{P}$ in the natural gradient-like system.

The choice of ψ enables the design of adaptation algorithms that respect physical Riemannian constraints (Lee, Wensing, & Park, 2020) obeyed by the true parameters, as in the estimation of mass properties in robotics (Wensing, Kim, & Slotine, 2018). Similarly, it allows one to introduce a priori bounds on parameter estimates without resorting to parameter projection

techniques by choosing ψ to be a log-barrier function (Wensing & Slotine, 2020). In section 3.1, we further prove that the choice of ψ implicitly regularizes the learned system model.

Remark 4. The relation 2.18 shows that Tyukin's algorithm, equation 2.12, can be generalized to have a parameter estimate-dependent gain matrix. Indeed, consideration of the Lyapunov-like function $V = \frac{1}{\gamma} d_\psi(\mathbf{a} \,\|\, \hat{\mathbf{a}})$ shows that the algorithm

$$\dot{\hat{\mathbf{a}}} = -\gamma \tilde{f}(\mathbf{x}, \hat{\mathbf{a}}, \mathbf{a}, t) \left[\nabla^2 \psi(\hat{\mathbf{a}})\right]^{-1} \boldsymbol{\alpha}(\mathbf{x}, t),$$

with ψ strongly convex and $\gamma > 0$ ensures that $\tilde{f} \in \mathcal{L}_2$ over the maximal interval of existence of $\mathbf{x}(t)$ for nonlinearly parameterized systems categorized by assumption 2. The proof is identical to that of Tyukin et al. (2007). The implementation of this algorithm in PI form will be described in remark 8 and is based on a correspondence between mirror descent and natural gradient descent in continuous time. This algorithm can be seen as the adaptive control equivalent of a mirror descent or natural gradient extension of the GLMTron of Kakade et al. (2011), and this correspondence will be considered in greater detail in section 6.

Remark 5. In the linearly parameterized setting, rather than the Lyapunov-like function $V = \frac{1}{2}s^2 + d_\psi(\mathbf{a} \,\|\, \hat{\mathbf{a}})$, the Lyapunov-like function $V = \frac{1}{2}s^2 + d_\psi(\mathbf{Pa} \,\|\, \mathbf{P}\hat{\mathbf{a}})$ may be used for any positive-definite matrix $\mathbf{P}$. This shows stability of the adaptation law $\dot{\hat{\mathbf{a}}} = -\mathbf{P}^{-1} \left(\nabla^2 \psi(\mathbf{P}\hat{\mathbf{a}})\right)^{-1} \mathbf{P}^{-1} \mathbf{Y}^{\mathrm{T}} s$, where the choice of the matrix $\mathbf{P}$ offers an additional design flexibility.

Remark 6. In some practical applications, as in adaptive robot control, the estimated parameters $\hat{\mathbf{a}}$ may correspond to physical constants. In this case, the weighted parameter estimation error term $\frac{1}{2}\tilde{\mathbf{a}}^{\mathrm{T}}\mathbf{P}^{-1}\tilde{\mathbf{a}}$ not only provides additional design flexibility through the elements of $\mathbf{P}$ in the adaptation law, but is necessary for physical consistency of units. Indeed, the usual Lyapunov-like function $V = \frac{1}{2}s^2 + \frac{1}{2}\tilde{\mathbf{a}}^{\mathrm{T}}\mathbf{P}^{-1}\tilde{\mathbf{a}}$ shows that $\mathbf{P}^{-1}$ must be chosen so that the parameter estimation error term $\frac{1}{2}\tilde{\mathbf{a}}^{\mathrm{T}}\mathbf{P}^{-1}\tilde{\mathbf{a}}$ has the same units as the tracking error term $\frac{1}{2}s^2$. Similar considerations apply when replacing this standard parameter estimation error term with the Bregman divergence $d_\psi(\mathbf{a} \,\|\, \hat{\mathbf{a}})$, which has the same units as $\psi(\hat{\mathbf{a}})$. In this case, $\psi(\hat{\mathbf{a}})$ must be chosen to have the same units as the tracking error term, for example, by introducing a diagonal matrix of constants to ensure consistent dimensions.

3 Natural Gradient Adaptation and Implicit Regularization

In this section, we show that the "natural" adaptation algorithms of the previous section implicitly regularize the learned system model.

3.1 Implicit Regularization and Adaptive Control. With deep networks as the predominant example, modern machine learning often considers highly overparameterized models that are capable of interpolating the training data (achieving zero error on the training set) while still generalizing well to unseen examples. The classical principles of statistical learning theory emphasize a trade-off between generalization performance and model capacity, and predict that in the highly overparameterized regime, generalization performance should be poor due to a tendency of the model to fit noise in the training data. Nevertheless, empirical evidence indicates that deep networks and other modern machine learning models do not obey classical statistical learning wisdom (Belkin et al., 2019) and can even generalize with significant label noise (Zhang, Bengio, Hardt, Recht, & Vinyals, 2016).

More surprisingly, the ability to simultaneously fit label noise in the training data yet generalize to new examples has been observed in overparameterized linear models (Bartlett, Long, Lugosi, & Tsigler, 2020; Muthukumar, Vodrahalli, & Sahai, 2019). A possible explanation for the ability of highly overparameterized models to generalize when optimized using simple first-order algorithms is their implicit bias—that is, the tendency of an algorithm to converge to a particular (e.g., minimum norm) solution when there are many that interpolate the training data (Azizan & Hassibi, 2019; Azizan et al., 2019; Gunasekar et al., 2018a, 2018b; Soudry et al., 2018).

In adaptive control, the possibility of there being many possible parameter vectors $\hat{\mathbf{a}}$ that lead to zero tracking error is not unique to the overparameterized case. Unless the trajectory is *persistently exciting*[2] (Narendra & Annaswamy, 2005; Slotine & Li, 1991), it is well known that $\hat{\mathbf{a}}$ will not converge to the true parameters $\mathbf{a}$ in general. Depending on the complexity of the trajectory, there may even be many solutions in the *underparameterized* case where $\dim(\hat{\mathbf{a}}) < \dim(\mathbf{a})$. To achieve perfect tracking, the adaptation algorithm need only fit the unknown dynamics $f(\mathbf{x}(t), \mathbf{a}, t)$ along the trajectory rather than the whole state space, so that the effective number of parameters may be less than $\dim(\mathbf{a})$.

The wealth of possible solutions in the linearly parameterized case is captured by the time-dependent null space of $\mathbf{Y}(\mathbf{x}(t), t)$: when $\mathbf{x} \to \mathbf{x}_d$, we can conclude that $\mathbf{Y}(\mathbf{x}_d(t), t)\tilde{\mathbf{a}}(t) = 0$, and hence that $\hat{\mathbf{a}}(t) = \mathbf{a} + \hat{\mathbf{n}}(t)$ where $\mathbf{Y}(\mathbf{x}_d(t), t)\hat{\mathbf{n}}(t) = 0$ for all t. This observation also highlights that any element $\hat{\mathbf{n}}(t)$ of the null space may be added to the parameter estimates $\hat{\mathbf{a}}$ without affecting the value of $\hat{f}$.[3] In the overparameterized case when

[2] A typical characterization of persistent excitation in the linearly parameterized setting is that there exists some $\delta > 0$ and some $T > 0$ such that for all t, $\int_{t_3}^{t+T} \mathbf{Y}^{\mathrm{T}}(\mathbf{x}(\tau), \tau)\mathbf{Y}(\mathbf{x}(\tau), \tau)d\tau \geq \delta\mathbf{I}$.

[3] In principle, $\hat{\mathbf{n}}(t)$ could be chosen to shape the parameters $\hat{\mathbf{a}}(t)$ to satisfy some desired property.

$\dim(\hat{\mathbf{a}}) > \dim(\mathbf{a})$, the set of parameters that achieve zero tracking error is not unique regardless of the complexity of the desired trajectory. By deriving a continuous-time extension of a recent proof of the implicit bias of mirror descent algorithms (Azizan & Hassibi, 2019; Azizan et al., 2019), we now show that the natural adaptive laws of the previous section implicitly regularize $\hat{\mathbf{a}}$. This proof of implicit regularization provides an answer to the question, with infinitely many parameter vectors that achieve zero tracking error, which does adaptation choose?

Define the set

$$\mathcal{A} = \{\boldsymbol{\theta} | f(\mathbf{x}(t), \boldsymbol{\theta}, t) = f(\mathbf{x}(t), \mathbf{a}, t) \ \ \forall t\}, \tag{3.1}$$

that is, the set in equation 3.1 contains only parameters that interpolate the dynamics $f(\mathbf{x}(t), \mathbf{a}, t)$ along the entire trajectory. We are now in a position to state the following proposition.

Proposition 1. *Consider the natural gradient-like adaptation law for a linearly parameterized dynamics,*

$$\dot{\hat{\mathbf{a}}} = -\left[\nabla^2 \psi(\hat{\mathbf{a}})\right]^{-1} \mathbf{Y}^{\mathrm{T}} s, \tag{3.2}$$

where $\psi(\cdot)$ is a strongly convex function. Assume that $\hat{\mathbf{a}}(t) \to \hat{\mathbf{a}}_\infty \in \mathcal{A}$. Then

$$\hat{\mathbf{a}}_\infty = \arg\min_{\boldsymbol{\theta} \in \mathcal{A}} d_\psi(\boldsymbol{\theta} \,|\, \hat{\mathbf{a}}(0)).$$

In particular, if $\hat{\mathbf{a}}(0) = \arg\min_{\boldsymbol{\theta} \in \mathbb{R}^p} \psi(\boldsymbol{\theta})$, then

$$\hat{\mathbf{a}}_\infty = \arg\min_{\boldsymbol{\theta} \in \mathcal{A}} \psi(\boldsymbol{\theta}). \tag{3.3}$$

Proof. Let $\boldsymbol{\theta}$ be any constant vector of parameters. The Bregman divergence $d_\psi(\boldsymbol{\theta} \,\|\, \hat{\mathbf{a}}) = \psi(\boldsymbol{\theta}) - \psi(\hat{\mathbf{a}}) - \nabla\psi(\hat{\mathbf{a}})^{\mathrm{T}}(\boldsymbol{\theta} - \hat{\mathbf{a}})$ has the time derivative

$$\frac{d}{dt} d_\psi(\boldsymbol{\theta} \,\|\, \hat{\mathbf{a}}) = -\left(\frac{d}{dt}\nabla\psi(\hat{\mathbf{a}})\right)^{\mathrm{T}} (\boldsymbol{\theta} - \hat{\mathbf{a}}).$$

From equation 3.2, $\frac{d}{dt}\nabla\psi(\hat{\mathbf{a}}) = -\mathbf{Y}^{\mathrm{T}} s$, so that

$$\frac{d}{dt} d_\psi(\boldsymbol{\theta} \,\|\, \hat{\mathbf{a}}) = s\mathbf{Y}(\boldsymbol{\theta} - \hat{\mathbf{a}}).$$

Integrating both sides of the above shows that

$$d_\psi(\boldsymbol{\theta} \,|\, \hat{\mathbf{a}}(0)) = d_\psi(\boldsymbol{\theta} \,\|\, \hat{\mathbf{a}}(t)) + \int_0^t s(\tau)\mathbf{Y}(\mathbf{x}(\tau), \tau)(\hat{\mathbf{a}}(\tau) - \boldsymbol{\theta})\, d\tau.$$

If we now take $\boldsymbol{\theta} \in \mathcal{A}$, $\mathbf{Y}(\mathbf{x}(\tau), \tau)\boldsymbol{\theta} = f(\mathbf{x}(\tau), \mathbf{a}, \tau)$ and the integral term is independent of $\boldsymbol{\theta}$. Assuming that $\hat{\mathbf{a}} \to \hat{\mathbf{a}}_\infty \in \mathcal{A}$, we can take the limit as $t \to \infty$ and say that for any $\boldsymbol{\theta} \in \mathcal{A}$, $\hat{\mathbf{a}}_\infty \in \mathcal{A}$,

$$d_\psi(\boldsymbol{\theta} \mid \hat{\mathbf{a}}(0)) = d_\psi(\boldsymbol{\theta} \parallel \hat{\mathbf{a}}_\infty) + \int_0^\infty s(\tau)\left(\mathbf{Y}(\mathbf{x}(\tau), \tau)\hat{\mathbf{a}}(\tau) - f(\mathbf{x}(\tau), \mathbf{a}, \tau)\right) d\tau.$$

Because the only dependence of the right-hand side on $\boldsymbol{\theta}$ is in the first term and because this relation holds for any $\boldsymbol{\theta}$, the arg min of the two Bregman divergences must be identical. The minimum of the right-hand side over $\boldsymbol{\theta}$ is clearly obtained at $\hat{\mathbf{a}}_\infty$, while the minimum of the left-hand side is by definition obtained at $\arg\min_{\boldsymbol{\theta} \in \mathcal{A}} d_\psi(\boldsymbol{\theta}, \hat{\mathbf{a}}(0))$. From this, we conclude that

$$\hat{\mathbf{a}}_\infty = \arg\min_{\boldsymbol{\theta} \in \mathcal{A}} d_\psi(\boldsymbol{\theta} \mid \hat{\mathbf{a}}(0)),$$

which completes the proof. □

Equation 3.3 captures the implicit regularization imposed by the adaptation algorithm equation 3.2: out of all possible interpolating parameters, it chooses the $\hat{\mathbf{a}}$ that achieves the minimum value of ψ.

Remark 7. The assumptions of proposition 1 provide a setting where theoretical insight may be gained into the implicit regularization of adaptive control algorithms, but they are stronger than needed. In general, the parameters $\hat{\mathbf{a}}(t)$ found by an adaptive controller need not converge to a constant despite the fact that $\dot{\hat{\mathbf{a}}} \to 0$.[4] Similarly, even in the case that the parameters converge, it is not strictly required that $\mathbf{Y}(\mathbf{x}(t), t)\hat{\mathbf{a}}_\infty = f(\mathbf{x}(t), \mathbf{a}, t)$ along the entire trajectory, as this condition is satisfied asymptotically. Numerical simulations in section 8 will demonstrate the implicit regularization of parameters $\hat{\mathbf{a}}(t)$ found by adaptive control along the entire trajectory.

We may make a similar claim in the nonlinearly parameterized setting captured by assumption 2. To do so, we require an additional assumption.

Assumption 3. *For any vector of parameters $\boldsymbol{\theta}$ and the true parameters $\mathbf{a}$, $f(\mathbf{x}(t), \boldsymbol{\theta}, t) = f(\mathbf{x}(t), \mathbf{a}, t)$ implies that $\boldsymbol{\alpha}(\mathbf{x}(t), t)^{\mathrm{T}}\boldsymbol{\theta} = \boldsymbol{\alpha}(\mathbf{x}(t), t)^{\mathrm{T}}\mathbf{a}$.*

[4] Lyapunov function arguments based on a parameter estimation error term generally lead to the conclusion that the parameters remain bounded, and it is generally the case that $\dot{\hat{\mathbf{a}}} \to 0$ as it is driven by an error term. Nevertheless, $\hat{\mathbf{a}}$ may stay time-varying for all t. For instance, the function $f(t) = \sin(\sqrt{t})$ remains bounded and time-varying for all t, but has $f'(t) = \frac{1}{2\sqrt{t}}\cos(\sqrt{t}) \to 0$. A sufficient condition (by Barbalat's lemma [lemma 1 in this article]) for convergence to a constant $\hat{\mathbf{a}}_\infty$ is that $(\hat{\mathbf{a}} - \hat{\mathbf{a}}_\infty) \in \mathcal{L}_p$ for some p.

For the class of systems 2.11, a sufficient condition for assumption 3 is that $\lambda(\mathbf{x}(t), t) \neq 0$ and that the map $\boldsymbol{\phi}(\mathbf{x}, t)^{\mathrm{T}}\mathbf{a} \to f_m(\mathbf{x}(t), \boldsymbol{\phi}(\mathbf{x}, t)^{\mathrm{T}}\mathbf{a})$ is invertible at every t. We may now state our implicit regularization result for nonlinearly parameterized systems.

Proposition 2. *Consider the adaptation algorithm*

$$\dot{\hat{\mathbf{a}}} = -\left[\nabla^2\psi(\hat{\mathbf{a}})\right]^{-1} \tilde{f}(\mathbf{x}(t), \hat{\mathbf{a}}(t), \mathbf{a}, t)\boldsymbol{\alpha}(\mathbf{x}(t), t) \tag{3.4}$$

under assumptions 2 and 3. Assume $\hat{\mathbf{a}}(t) \to \hat{\mathbf{a}}_\infty \in \mathcal{A}$. *Then*

$$\hat{\mathbf{a}}_\infty = \arg\min_{\boldsymbol{\theta}\in\mathcal{A}} d_\psi(\boldsymbol{\theta} \mid \hat{\mathbf{a}}(0)).$$

Proof. The proof is much the same as proposition 1. The Bregman divergence $d_\psi(\boldsymbol{\theta} \parallel \hat{\mathbf{a}})$ for any fixed vector of parameters $\boldsymbol{\theta}$ verifies

$$\frac{d}{dt} d_\psi(\boldsymbol{\theta} \parallel \hat{\mathbf{a}}) = \tilde{f}(\mathbf{x}(t), \hat{\mathbf{a}}(t), \mathbf{a}, t)\boldsymbol{\alpha}(\mathbf{x}(t), t)^{\mathrm{T}}\left(\boldsymbol{\theta} - \hat{\mathbf{a}}\right),$$

so that, integrating both sides,

$$d_\psi(\boldsymbol{\theta} \parallel \hat{\mathbf{a}}(0)) = d_\psi(\boldsymbol{\theta} \parallel \hat{\mathbf{a}}(t)) - \int_0^t \tilde{f}(\mathbf{x}(\tau), \hat{\mathbf{a}}(\tau), \mathbf{a}, \tau)\boldsymbol{\alpha}(\mathbf{x}(\tau), \tau)^{\mathrm{T}}\left(\boldsymbol{\theta} - \hat{\mathbf{a}}(\tau)\right) d\tau.$$

Now take $\boldsymbol{\theta} \in \mathcal{A}$. By the assumptions of the proposition, $\boldsymbol{\alpha}(\mathbf{x}(\tau), \tau)^{\mathrm{T}}\boldsymbol{\theta} = \boldsymbol{\alpha}(\mathbf{x}(\tau), \tau)^{\mathrm{T}}\mathbf{a}$ is independent of $\boldsymbol{\theta}$. Hence, using that $\hat{\mathbf{a}}(t) \to \hat{\mathbf{a}}_\infty \in \mathcal{A}$, we can write

$$d_\psi(\boldsymbol{\theta} \parallel \hat{\mathbf{a}}(0)) = d_\psi(\boldsymbol{\theta} \parallel \hat{\mathbf{a}}_\infty) - \int_0^\infty \tilde{f}(\mathbf{x}(\tau), \hat{\mathbf{a}}(\tau), \mathbf{a}, \tau)\boldsymbol{\alpha}(\mathbf{x}(\tau), \tau)^{\mathrm{T}}\left(\mathbf{a} - \hat{\mathbf{a}}(\tau)\right) d\tau.$$

Optimizing both sides over $\boldsymbol{\theta} \in \mathcal{A}$ as in proposition 1 yields the result. □

Remark 8. Algorithm 3.4 must be implemented in PI form due to the appearance of $\tilde{f}$. The use of the PI form in $\hat{\mathbf{a}}$, equations 2.13 to 2.16, is complicated by the presence of the inverse Hessian of ψ. To implement equation 2.12, the Euclidean variant may be implemented through the usual PI form for an auxiliary variable $\dot{\hat{\mathbf{v}}} = -\tilde{f}(\mathbf{x}(t), \hat{\mathbf{a}}(t), \mathbf{a}, t)\boldsymbol{\alpha}(\mathbf{x}(t), t)$, and then the controller parameters may be computed by inverting the gradient of ψ, $\hat{\mathbf{a}}(t) = \left(\nabla\psi^{-1}\right)(\hat{\mathbf{v}}(t))$. This follows by the equivalence of mirror descent and natural gradient descent in continuous time. Concretely, the identity $\frac{d}{dt}\nabla\psi(\hat{\mathbf{a}}) = \nabla^2\psi(\hat{\mathbf{a}})\dot{\hat{\mathbf{a}}}$ shows that $\dot{\hat{\mathbf{a}}} = -\left(\nabla^2\psi(\hat{\mathbf{a}})\right)^{-1}\tilde{f}(\mathbf{x}, \hat{\mathbf{a}}, \mathbf{a}, t)\boldsymbol{\alpha}(\mathbf{x}, t)$

is equivalent to $\frac{d}{dt}\nabla\psi(\hat{\mathbf{a}}) = -\tilde{f}(\mathbf{x}, \hat{\mathbf{a}}, \mathbf{a}, t)\boldsymbol{\alpha}(\mathbf{x}, t)$. The auxiliary variable $\hat{\mathbf{v}}$ can then be identified with $\nabla\psi(\hat{\mathbf{a}})$.

Remark 9. If the inverse gradient of ψ is unknown but ψ is chosen to be strongly convex, the contracting (Lohmiller & Slotine, 1998) dynamics $\dot{\mathbf{w}} = -\frac{1}{\tau}(\nabla\psi(\mathbf{w}) - \nabla\psi(\hat{\mathbf{v}}))$ with $\tau > 0$ will converge to a ball around $\hat{\mathbf{v}}$ with radius set by $\|\frac{d}{dt}\nabla\psi(\hat{\mathbf{v}})\| \times \frac{\tau}{l}$ where l is the strong convexity parameter. By choosing τ so that this contracting dynamics is fast on the timescale of adaptation, $\mathbf{w}$ will represent a good approximation of the instantaneous $\hat{\mathbf{v}}$.

Remark 10. Our results highlight, through the equivalence of their continuous-time limits, that both mirror descent–like and natural gradient-like adaptive laws impose implicit regularization. This observation extends recent results on the implicit regularization of mirror descent (Azizan & Hassibi, 2019; Azizan et al., 2019) to natural gradient descent and, furthermore, applies to linearly parameterized and generalized linearly parameterized models in machine learning, not just in the context of adaptive control. This has previously been noted in Gunasekar et al. (2018a), where it was discussed that in discrete time, natural gradient descent only approximately imposes implicit regularization due to discretization errors.

Propositions 1 and 2 demonstrate for the first time the implicit bias of adaptive control algorithms. In doing so, they identify an additional design choice that may be exploited for the application of interest. Proposition 1 implies that the Slotine and Li controller, when initialized with the parameters at $\hat{\mathbf{a}}(0) = \mathbf{0}$, finds the interpolating parameter vector of minimum l_2 norm. Other norms, such as the l_1, l_∞, l_p for arbitrary p, or group norms will find alternative parameter vectors that may have desirable properties such as sparsity.[5] The usual Euclidean geometry–based adaptive laws can be seen as a form of ridge regression, while imposed l_1, l_2 and l_1 simultaneously, or l_p regularization through the choice of ψ can be seen as the adaptive control equivalents of LASSO (Tibshirani, 1996) or compressed sensing, elastic net, and bridge regression, respectively. In the context of adaptive control, this notion of implicit regularization is particularly interesting, as typical regularization terms such as l_1 and l_2 penalties cannot in general be added to the adaptation law directly without affecting the stability and performance of the algorithm.

Remark 11. Following remark 6, if $\psi(\cdot)$ is chosen as the lth power of a p norm, in practical applications it is necessary to include a matrix $\boldsymbol{\Gamma}$ to ensure that $\psi(\hat{\mathbf{a}}) = \|\boldsymbol{\Gamma}\hat{\mathbf{a}}\|_p^l$ has the same units as the tracking error

[5] Because the l_1 norm is not strongly convex, it may be replaced with a suitable approximation such as the $l_{1+\epsilon}$ norm for $\epsilon > 0$ and small (Azizan & Hassibi, 2019; Azizan et al., 2019).

component of the Lyapunov function. For example, if $l = 2$, then $\mathbf{\Gamma}$ may be chosen as $\mathbf{\Gamma} = \mathbf{P}^{-1/2}$ for consistency of units where $\mathbf{P}$ is a gain matrix tuned for the standard adaptive law $\dot{\hat{\mathbf{a}}} = -\mathbf{P}\mathbf{Y}^{\mathrm{T}}s$. In addition, $l = 2$ admits a simple inversion formula for $\nabla\psi$ for any p, as will be utilized in the simulations in section 8, although the corresponding inverse Hessian $\left(\nabla^2\psi(\cdot)\right)^{-1}$ is nondiagonal for $p \neq 2$. For $l = p$, the inverse Hessian is diagonal, but $\mathbf{\Gamma}$ must then be calibrated independently from $\mathbf{P}$ tuned for the standard l_2 law. Note that choosing ψ to be an l_1 norm will impose sparsity on $\mathbf{\Gamma}\hat{\mathbf{a}}$, so that $\mathbf{\Gamma}$ should be taken to be diagonal to ensure sparsity in $\hat{\mathbf{a}}$ itself.

3.2 Non-Euclidean Measure of the Tracking Error. The usual Lyapunov function incorporates a Euclidean tracking error term given by $\frac{1}{2}s^2$. In a similar vein to the derivation of the "natural" adaptive laws, for any strictly convex function $\phi : \mathbb{R} \to \mathbb{R}$, we may instead replace this tracking error term by the Bregman divergence $d_\phi(0 \,\|\, s)$. This quantity has time derivative

$$\frac{d}{dt}d_\phi(0 \,\|\, s) = -\eta s^2\phi''(s) + \phi''(s)\mathbf{Y}\tilde{\mathbf{a}}$$

in the linearly parameterized case. Because $\phi''(s) \geq 0$ for strictly convex ϕ, it is simple to see that this modification to the usual Lyapunov function in combination with a non-Euclidean measure of the parameter estimation error leads to a family of stable adaptation laws parameterized by ϕ and ψ of the form $\dot{\hat{\mathbf{a}}} = -\left[\nabla^2\psi(\hat{\mathbf{a}})\right]^{-1}\mathbf{Y}^{\mathrm{T}}\phi''(s)s$. This shows, for example, that any odd power of s may be stably employed in the adaptation law by taking $\phi = s^p$ for even some power p. Surprisingly, more exotic adaptation laws such as $\dot{\hat{\mathbf{a}}} = -\left[\nabla^2\psi(\hat{\mathbf{a}})\right]^{-1}\mathbf{Y}^{\mathrm{T}}e^{\lambda|s|}s$ for $\lambda > 0$ may also be used.

In the single-input case, these laws could be more simply obtained by replacing the $\frac{1}{2}s^2$ term in the Lyapunov-like function with a term of the form $g(s)$ where $g'(s)s \geq 0$ and $g'(s)$ is known. In the multi-input case, these two approaches differ. Taking g to be a strongly convex function with minimum attained at $s = 0$ and a known gradient, the Lyapunov-like function

$$V = g(\mathbf{s}) - \inf_{\mathbf{s}} g(\mathbf{s}) + d_\psi(\mathbf{a} \,\|\, \hat{\mathbf{a}})$$

shows that the adaptation law

$$\dot{\hat{\mathbf{a}}} = -\left[\nabla^2\psi(\hat{\mathbf{a}})\right]^{-1}\mathbf{Y}^{\mathrm{T}}\nabla g(\mathbf{s})$$

is globally convergent. On the other hand, the Lyapunov-like function

$$V = d_\phi(0 \,\|\, \mathbf{s}) + d_\psi(\mathbf{a} \,\|\, \hat{\mathbf{a}})$$

shows that the distinct adaptation law

$$\dot{\hat{\mathbf{a}}} = -\left[\nabla^2\psi(\hat{\mathbf{a}})\right]^{-1}\mathbf{Y}^{\mathrm{T}}\left[\nabla^2\phi(\mathbf{s})\right]\mathbf{s}$$

is also globally convergent.

4 Adaptive Dynamics Prediction, Control, and Observer Design

In this section, we demonstrate how the new non-Euclidean adaptation laws of section 3.1 may be used for regularized dynamics prediction, regularized adaptive control, and regularized observer design.

4.1 Regularized Adaptive Dynamics Prediction. Similar to direct adaptive control, online parameter estimation may also be used within an observer-like framework for dynamics prediction. This enables, for instance, the design of provably stable online learning rules for the weights of a recurrent neural network in the dynamics approximation context (Alemi, Machens, Deneve, & Slotine, 2018; Gilra & Gerstner, 2017; Sussillo & Abbott, 2009). Consider a nonlinear system dynamics

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{c}(t),$$

where $\mathbf{x} \in \mathbb{R}^n$ is the system state, $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ is the system dynamics, and $\mathbf{c} : \mathbb{R}_+ \to \mathbb{R}^n$ is a system input. Define the observer-like system

$$\dot{\hat{\mathbf{x}}} = -k(\hat{\mathbf{x}} - \mathbf{x}) + \mathbf{Y}(\hat{\mathbf{x}})\hat{\mathbf{a}} + \mathbf{c}(t),$$

where $\mathbf{Y} : \mathbb{R}^n \to \mathbb{R}^{n\times p}$, $\hat{\mathbf{a}} \in \mathbb{R}^p$, and $k > 0$ is a scalar gain. Assume that there exists a fixed parameter vector $\mathbf{a}$ such that for all $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{Y}(\mathbf{x})\mathbf{a} = \mathbf{f}(\mathbf{x})$. By adding and subtracting $\mathbf{f}(\hat{\mathbf{x}}) = \mathbf{Y}(\hat{\mathbf{x}})\mathbf{a}$, the error $\mathbf{e} = \hat{\mathbf{x}} - \mathbf{x}$ has dynamics

$$\dot{\mathbf{e}} = -k\mathbf{e} + \mathbf{Y}(\hat{\mathbf{x}})\tilde{\mathbf{a}} + \mathbf{f}(\hat{\mathbf{x}}) - \mathbf{f}(\mathbf{x}).$$

Consider the parameter estimator

$$\dot{\hat{\mathbf{a}}} = -\gamma\left[\nabla^2\psi(\hat{\mathbf{a}})\right]^{-1}\mathbf{Y}^{\mathrm{T}}(\hat{\mathbf{x}})\boldsymbol{\Gamma}\mathbf{e}, \tag{4.1}$$

where $\gamma > 0$ is a constant learning rate, ψ is a strongly convex potential function, and $\boldsymbol{\Gamma} \in \mathbb{R}^{n\times n} > 0$ is a constant symmetric positive definite matrix. Now consider the Lyapunov-like function

$$V = \frac{1}{2}\mathbf{e}^{\mathrm{T}}\boldsymbol{\Gamma}\mathbf{e} + \frac{1}{\gamma}d_\psi(\mathbf{a} \,\|\, \hat{\mathbf{a}}),$$

which has time derivative

$$\begin{aligned}\dot{V} &= \mathbf{e}^{\mathrm{T}}\mathbf{\Gamma}\left(-k\mathbf{e} + \mathbf{Y}(\hat{\mathbf{x}})\tilde{\mathbf{a}} + \mathbf{f}(\hat{\mathbf{x}}) - \mathbf{f}(\mathbf{x})\right) - \tilde{\mathbf{a}}^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}(\hat{\mathbf{x}})\mathbf{\Gamma}\mathbf{e}, \\ &= \mathbf{e}^{\mathrm{T}}\mathbf{\Gamma}\left(-k\mathbf{e} + \mathbf{f}(\hat{\mathbf{x}}) - \mathbf{f}(\mathbf{x})\right), \\ &= \mathbf{e}^{\mathrm{T}}\left(\int_0^1 \left(\mathbf{\Gamma}\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x} + s\mathbf{e}) - k\mathbf{\Gamma}\right) ds\right)\mathbf{e}. \end{aligned} \tag{4.2}$$

Equation 4.2 shows that $e \to 0$ as long as $\mathbf{f}(\mathbf{x}) - k\mathbf{x}$ is contracting in the metric $\mathbf{\Gamma}$ (Lohmiller & Slotine, 1998; Slotine, 2003), that is, if

$$\left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}\right)^{\mathrm{T}}\mathbf{\Gamma} + \mathbf{\Gamma}\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \leq 2\left(k - \lambda\right)\mathbf{\Gamma}$$

uniformly over $\mathbf{x}$ for some contraction rate $\lambda > 0$. It is simple to check that the metric $\mathbf{\Gamma}$ may also be time dependent, $\mathbf{\Gamma} = \mathbf{\Gamma}(t)$. More generally, rather than the proportional term $-k\mathbf{e}$, any term of the form $\mathbf{g}(\hat{\mathbf{x}}) - \mathbf{g}(\mathbf{x})$ may be used in $\dot{\hat{\mathbf{x}}}$, leading to the condition

$$\left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} + \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}\right)^{\mathrm{T}}\mathbf{\Gamma} + \mathbf{\Gamma}\left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} + \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}\right) \leq -2\lambda\mathbf{\Gamma}$$

uniformly over $\mathbf{x}$ for some contraction rate $\lambda > 0$. The implicit regularization results of section 3.1 show that this framework provides a technique for provably regularizing learned predictive dynamics models without negatively affecting stability or convergence of the combined error and parameter estimation systems.

The above discussion demonstrates a separation theorem for adaptive dynamics prediction. If a dynamics predictor can be designed under the assumption that the true system dynamics is known (e.g., if bounds on $\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}$ are available), then the same dynamics predictor can be made adaptive by incorporating the skew-symmetric law, equation 4.1. Convergence properties then only depend on the nominal system with control feedback and are independent of the parameter estimator, as shown by the conditions for contraction.

Remark 12. In principle, these simple results could be made more general using the techniques developed in Lopez and Slotine (2021), or could be performed in a latent space computed via a nonlinear dimensionality-reduction technique such as an autoencoder (Champion, Lusch, Kutz, & Brunton, 2019) or more generally a hierarchical expansion (Chen, Paiton, & Olshausen, 2018). This could also extend to adaptive control, for example, in robot control applications where an adaptive controller could be designed in a latent space computed from raw pixels via a neural network.

4.2 Regularized Dynamics Prediction for Hamiltonian Systems. If the underlying system is known to have a specific structure, this structure may be leveraged in a principled way to adaptively compute models for dynamics prediction (Sanner & Slotine, 1995). For example, large classes of physical systems are described by Hamiltonian dynamics,

$$\begin{aligned}\mathcal{H} &= \mathcal{H}(\mathbf{p}, \mathbf{q}),\\ \dot{\mathbf{p}} &= -\nabla_{\mathbf{q}}\mathcal{H}(\mathbf{p}, \mathbf{q}),\\ \dot{\mathbf{q}} &= \nabla_{\mathbf{p}}\mathcal{H}(\mathbf{p}, \mathbf{q}),\end{aligned}$$

where $\mathcal{H}(\mathbf{p}, \mathbf{q})$ is the system Hamiltonian, $\mathbf{p}$ is the generalized momentum, and $\mathbf{q}$ is the generalized coordinate conjugate to $\mathbf{p}$. This structure was exploited in recent work by Chen, Zhang, Arjovsky, and Bottou (2020) via direct estimation of the system Hamiltonian with a deep feedforward network in combination with symplectic integration of the resulting dynamics. In a similar spirit, rather than parameterizing the system dynamics as in section 4.1, consider estimating the scalar Hamiltonian itself as a linear expansion in a set of known nonlinear basis functions $\{Y_k\}$,

$$\widehat{\mathcal{H}}(\hat{\mathbf{a}}, \mathbf{p}, \mathbf{q}) = \sum_k \hat{a}_k Y_k(\mathbf{p}, \mathbf{q}) = \mathbf{Y}(\mathbf{p}, \mathbf{q})\hat{\mathbf{a}},$$

where $\mathbf{Y}(\mathbf{p}, \mathbf{q}) \in \mathbb{R}^{1\times p}$ is a row vector of basis functions. Assume that there exists some true parameter vector $\mathbf{a}$ that exactly approximates the Hamiltonian globally, and consider the dynamics prediction model for $k_p > 0$, $k_q > 0$:

$$\dot{\hat{\mathbf{p}}} = -\left(\nabla_{\hat{\mathbf{q}}}\mathbf{Y}(\hat{\mathbf{p}}, \hat{\mathbf{q}})\right)\hat{\mathbf{a}} + k_p\left(\mathbf{p} - \hat{\mathbf{p}}\right), \tag{4.3}$$

$$\dot{\hat{\mathbf{q}}} = \left(\nabla_{\hat{\mathbf{p}}}\mathbf{Y}(\hat{\mathbf{p}}, \hat{\mathbf{q}})\right)\hat{\mathbf{a}} + k_q\left(\mathbf{q} - \hat{\mathbf{q}}\right). \tag{4.4}$$

The above predictor employs parameter sharing between both dynamics due to the direct estimation of the system Hamiltonian. The basis functions for the individual dynamics reflect the symplectic structure, as they are given by partial derivatives of the basis functions for the Hamiltonian.

After subtracting the true dynamics $\dot{\mathbf{p}}$ and $\dot{\mathbf{q}}$ from above, consider the decomposition of the error dynamics,

$$\begin{aligned}\dot{\tilde{\mathbf{p}}} = &-\left(\nabla_{\hat{\mathbf{q}}}\mathbf{Y}(\hat{\mathbf{p}}, \hat{\mathbf{q}})\right)\tilde{\mathbf{a}} - k_p\tilde{\mathbf{p}} - \left(\nabla_{\hat{\mathbf{q}}}\mathcal{H}(\hat{\mathbf{p}}, \hat{\mathbf{q}}) - \nabla_{\hat{\mathbf{q}}}\mathcal{H}(\mathbf{p}, \hat{\mathbf{q}})\right)\\ &-\left(\nabla_{\hat{\mathbf{q}}}\mathcal{H}(\mathbf{p}, \hat{\mathbf{q}}) - \nabla_{\mathbf{q}}\mathcal{H}(\mathbf{p}, \mathbf{q})\right),\\ \dot{\tilde{\mathbf{q}}} = &\left(\nabla_{\hat{\mathbf{p}}}\mathbf{Y}(\hat{\mathbf{p}}, \hat{\mathbf{q}})\right)\tilde{\mathbf{a}} - k_q\tilde{\mathbf{q}} + \left(\nabla_{\hat{\mathbf{p}}}\mathcal{H}(\hat{\mathbf{p}}, \hat{\mathbf{q}}) - \nabla_{\hat{\mathbf{p}}}\mathcal{H}(\hat{\mathbf{p}}, \mathbf{q})\right)\\ &+\left(\nabla_{\hat{\mathbf{p}}}\mathcal{H}(\hat{\mathbf{p}}, \mathbf{q}) - \nabla_{\mathbf{p}}\mathcal{H}(\mathbf{p}, \mathbf{q})\right),\end{aligned}$$

along with the adaptation law,

$$\dot{\hat{\mathbf{a}}} = \gamma \left[\nabla^2 \psi(\hat{\mathbf{a}})\right]^{-1} \left(\left(\nabla_{\hat{\mathbf{q}}} \mathbf{Y}(\hat{\mathbf{p}}, \hat{\mathbf{q}})\right)^{\mathrm{T}} \tilde{\mathbf{p}} - \left(\nabla_{\hat{\mathbf{p}}} \mathbf{Y}(\hat{\mathbf{p}}, \hat{\mathbf{q}})\right)^{\mathrm{T}} \tilde{\mathbf{q}} \right),$$

with $\gamma > 0$ a positive learning rate. The Lyapunov-like function

$$V = \frac{1}{2}\tilde{\mathbf{p}}^{\mathrm{T}}\tilde{\mathbf{p}} + \frac{1}{2}\tilde{\mathbf{q}}^{\mathrm{T}}\tilde{\mathbf{q}} + \frac{d_\psi(\mathbf{a} \,\|\, \hat{\mathbf{a}})}{\gamma}$$

has time derivative

$$\begin{aligned}
\dot{V} &= \tilde{\mathbf{p}}^{\mathrm{T}}\big[- \left(\nabla_{\hat{\mathbf{q}}} \mathbf{Y}(\hat{\mathbf{p}}, \hat{\mathbf{q}})\right) \tilde{\mathbf{a}} - k_p \tilde{\mathbf{p}} - \left(\nabla_{\hat{\mathbf{q}}} \mathcal{H}(\hat{\mathbf{p}}, \hat{\mathbf{q}}) - \nabla_{\hat{\mathbf{q}}} \mathcal{H}(\mathbf{p}, \hat{\mathbf{q}})\right) \\
&\quad - \left(\nabla_{\hat{\mathbf{q}}} \mathcal{H}(\mathbf{p}, \hat{\mathbf{q}}) - \nabla_{\mathbf{q}} \mathcal{H}(\mathbf{p}, \mathbf{q})\right)\big] \\
&\quad + \tilde{\mathbf{q}}^{\mathrm{T}}\big[\left(\nabla_{\hat{\mathbf{p}}} \mathbf{Y}(\hat{\mathbf{p}}, \hat{\mathbf{q}})\right) \tilde{\mathbf{a}} - k_q \tilde{\mathbf{q}} + \left(\nabla_{\hat{\mathbf{p}}} \mathcal{H}(\hat{\mathbf{p}}, \hat{\mathbf{q}}) - \nabla_{\hat{\mathbf{p}}} \mathcal{H}(\hat{\mathbf{p}}, \mathbf{q})\right) \\
&\quad + \left(\nabla_{\hat{\mathbf{p}}} \mathcal{H}(\hat{\mathbf{p}}, \mathbf{q}) - \nabla_{\mathbf{p}} \mathcal{H}(\mathbf{p}, \mathbf{q})\right)\big] \\
&\quad + \tilde{\mathbf{a}}^{\mathrm{T}} \left(\left(\nabla_{\hat{\mathbf{q}}} \mathbf{Y}(\hat{\mathbf{p}}, \hat{\mathbf{q}})\right)^{\mathrm{T}} \tilde{\mathbf{p}} - \left(\nabla_{\hat{\mathbf{p}}} \mathbf{Y}(\hat{\mathbf{p}}, \hat{\mathbf{q}})\right)^{\mathrm{T}} \tilde{\mathbf{q}} \right) \\
&= \tilde{\mathbf{p}}^{\mathrm{T}}\left[-k_p \tilde{\mathbf{p}} - \left(\nabla_{\hat{\mathbf{q}}} \mathcal{H}(\hat{\mathbf{p}}, \hat{\mathbf{q}}) - \nabla_{\hat{\mathbf{q}}} \mathcal{H}(\mathbf{p}, \hat{\mathbf{q}})\right) - \left(\nabla_{\hat{\mathbf{q}}} \mathcal{H}(\mathbf{p}, \hat{\mathbf{q}}) - \nabla_{\mathbf{q}} \mathcal{H}(\mathbf{p}, \mathbf{q})\right)\right] \\
&\quad + \tilde{\mathbf{q}}^{\mathrm{T}}\left[-k_q \tilde{\mathbf{q}} + \left(\nabla_{\hat{\mathbf{p}}} \mathcal{H}(\hat{\mathbf{p}}, \hat{\mathbf{q}}) - \nabla_{\hat{\mathbf{p}}} \mathcal{H}(\hat{\mathbf{p}}, \mathbf{q})\right) + \left(\nabla_{\hat{\mathbf{p}}} \mathcal{H}(\hat{\mathbf{p}}, \mathbf{q}) - \nabla_{\mathbf{p}} \mathcal{H}(\mathbf{p}, \mathbf{q})\right)\right] \\
&= \begin{pmatrix} \tilde{\mathbf{p}}^{\mathrm{T}} & \tilde{\mathbf{q}}^{\mathrm{T}} \end{pmatrix} \\
&\quad \times \begin{pmatrix} -k_p \mathbf{I} - \int_0^1 \nabla_{\mathbf{p}+s\tilde{\mathbf{p}}} \nabla_{\hat{\mathbf{q}}} \mathcal{H}(\mathbf{p} + s\tilde{\mathbf{p}}, \hat{\mathbf{q}}) ds & -\int_0^1 \nabla^2_{\mathbf{q}+s\tilde{\mathbf{q}}} \mathcal{H}(\mathbf{p}, \mathbf{q} + s\tilde{\mathbf{q}}) ds \\ \int_0^1 \nabla^2_{\mathbf{p}+s\tilde{\mathbf{p}}} \mathcal{H}(\mathbf{p} + s\tilde{\mathbf{p}}, \mathbf{q}) ds & -k_q \mathbf{I} + \int_0^1 \nabla_{\mathbf{q}+s\tilde{\mathbf{q}}} \nabla_{\hat{\mathbf{p}}} \mathcal{H}(\hat{\mathbf{p}}, \mathbf{q} + s\tilde{\mathbf{q}}) ds \end{pmatrix} \\
&\quad \times \begin{pmatrix} \tilde{\mathbf{p}} \\ \tilde{\mathbf{q}} \end{pmatrix}
\end{aligned}$$

A sufficient condition for convergence of $\tilde{\mathbf{p}} \to 0$ and $\tilde{\mathbf{q}} \to 0$ is uniform negative definiteness of the Jacobian matrix

$$\mathbf{J} = \begin{pmatrix} -k_p \mathbf{I} - \nabla_{\mathbf{p}} \nabla_{\mathbf{q}} \mathcal{H}(\mathbf{p}, \mathbf{q}) & -\nabla^2_{\mathbf{q}} \mathcal{H}(\mathbf{q}, \mathbf{p}) \\ \nabla^2_{\mathbf{p}} \mathcal{H}(\mathbf{p}, \mathbf{q}) & -k_q \mathbf{I} + \nabla_{\mathbf{q}} \nabla_{\mathbf{p}} \mathcal{H}(\mathbf{p}, \mathbf{q}) \end{pmatrix},$$

in $\mathbf{p}$ and $\mathbf{q}$, that is, contraction of the nominal $\dot{\hat{\mathbf{p}}}$ and $\dot{\hat{\mathbf{q}}}$ system in the Euclidean metric. Sufficient conditions for this are

$$k_p > -\frac{1}{2}\lambda_{\min}\left(\nabla_{\mathbf{p}}\nabla_{\mathbf{q}}\mathcal{H}(\mathbf{p},\mathbf{q}) + \nabla_{\mathbf{q}}\nabla_{\mathbf{p}}\mathcal{H}(\mathbf{p},\mathbf{q})\right),$$

$$k_q > \frac{1}{2}\lambda_{\max}\left(\nabla_{\mathbf{p}}\nabla_{\mathbf{q}}\mathcal{H}(\mathbf{p},\mathbf{q}) + \nabla_{\mathbf{q}}\nabla_{\mathbf{p}}\mathcal{H}(\mathbf{p},\mathbf{q})\right),$$

$$\lambda_p\lambda_q > \frac{1}{4}\lambda^2_{\max}\left[\nabla^2_{\mathbf{p}}\mathcal{H}(\mathbf{p},\mathbf{q}) - \nabla^2_{\mathbf{q}}\mathcal{H}(\mathbf{p},\mathbf{q})\right],$$

where λ_p and λ_q are the contraction rates of the $\hat{\mathbf{p}}$ and $\hat{\mathbf{q}}$ systems, respectively, given by the difference of the left- and right-hand sides of the first two inequalities above. More general conditions can be obtained by utilizing a nonidentity metric, that is, replacing the $\frac{1}{2}\tilde{\mathbf{p}}^{\mathrm{T}}\tilde{\mathbf{p}}$ and $\frac{1}{2}\tilde{\mathbf{q}}^{\mathrm{T}}\tilde{\mathbf{q}}$ terms in V by the Mahalanobis distances $\frac{1}{2}\tilde{\mathbf{p}}^{\mathrm{T}}\boldsymbol{\Gamma}_{\mathbf{p}}\tilde{\mathbf{p}}$ and $\frac{1}{2}\tilde{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\Gamma}_{\mathbf{q}}\tilde{\mathbf{q}}$ where $\boldsymbol{\Gamma}_{\mathbf{p}}$ and $\boldsymbol{\Gamma}_{\mathbf{q}}$ are symmetric positive-definite matrices. The adaptation law will need to be modified accordingly.

Rather than a general Hamiltonian $\mathcal{H} = \mathcal{H}(\mathbf{p},\mathbf{q})$, it is common to have a separable Hamiltonian structure,

$$\mathcal{H}(\mathbf{p},\mathbf{q}) = T(\mathbf{p}) + U(\mathbf{q}).$$

Above, $T(\cdot)$ is the kinetic energy and $U(\cdot)$ is the potential energy. Following an identical proof, the Jacobian matrix then reduces to

$$\mathbf{J} = \begin{pmatrix} -k_p\mathbf{I} & -\nabla^2_{\hat{\mathbf{q}}}U(\hat{\mathbf{q}}) \\ \nabla^2_{\hat{\mathbf{p}}}T(\hat{\mathbf{p}}) & -k_q\mathbf{I} \end{pmatrix},$$

so that the conditions for contraction in the Euclidean metric are simplified to

$$k_q k_p > \frac{1}{4}\lambda^2_{\max}\left(\nabla^2_{\hat{\mathbf{p}}}T(\hat{\mathbf{p}}) - \nabla^2_{\hat{\mathbf{q}}}U(\hat{\mathbf{q}})\right). \tag{4.5}$$

The results of section 3.1 show that the choice of ψ may be used to regularize the estimate of the Hamiltonian and, in turn, the dynamics. This may be used, for instance, for parsimonious Hamiltonian estimation through the combination of a rich set of physically motivated scalar basis functions and a sparse representation obtained via l_1 regularization, similar to Champion et al. (2019). Further results that exploit the structure of separable Hamiltonians through independent estimation of the kinetic and potential energies are presented in appendix B.

4.3 Regularized Adaptive Control for Lagrangian Systems. A similar methodology can be applied to parameterize a scalar Lagrangian rather than Hamiltonian, leading to a second-order differential equation with inertia matrix, centripetal and Coriolis forces, and potential energy parameterized by a shared set of weights. As we now show, generalizing the derivation of the Slotine and Li robot controller (Slotine & Li, 1987) to this setting allows for stable adaptive control of Lagrangian systems by direct estimation of the Lagrangian itself. Consider the Lagrangian

$$\mathcal{L} = \frac{1}{2}\dot{\mathbf{q}}^{\mathrm{T}}\mathbf{H}(\mathbf{q})\dot{\mathbf{q}} - U(\mathbf{q}),$$

with $\mathbf{H}(\mathbf{q})$ an unknown inertia matrix and $U(\mathbf{q})$ an unknown potential. Assume that the inertia matrix and scalar potential are given exactly by an expansion in physically motivated basis functions. That is, for a set of positive-definite matrices $\mathbf{M}^l > 0$ and scalar functions ϕ^l,

$$\mathbf{H}(\mathbf{q}) = \sum_l a_l^{(K)}\mathbf{M}^l(\mathbf{q}),$$
$$U(\mathbf{q}) = \sum_l a_l^{(P)}\phi^l(\mathbf{q}),$$

where superscript (K) and (P) denote kinetic and potential, respectively, and the vectors $\mathbf{a}^{(K)}$ and $\mathbf{a}^{(P)}$ are unknown. The Euler-Lagrange equations of motion $\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}} - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = \mathbf{u}$ with $\mathbf{u}$ a control input then give the dynamics

$$\sum_{lj} a_l^{(K)} M_{ij}^l(\mathbf{q})\ddot{q}_j + \sum_{lkj} a_l^{(K)}\dot{q}_k\dot{q}_j\left[\frac{\partial M_{ij}^l(\mathbf{q})}{\partial q_k} - \frac{1}{2}\frac{\partial M_{kj}^l(\mathbf{q})}{\partial q_i}\right] + \sum_l a_l^{(P)}\frac{\partial \phi^l(\mathbf{q})}{\partial q_i} = u_i.$$

Above, the second term,

$$\sum_{lkj} a_l^{(K)}\dot{q}_k\dot{q}_j\left[\frac{\partial M_{ij}^l(\mathbf{q})}{\partial q_k} - \frac{1}{2}\frac{\partial M_{kj}^l(\mathbf{q})}{\partial q_i}\right],$$

uniquely defines the centripetal and Coriolis forces (traditionally written as $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ with $\mathbf{C}$ the Coriolis matrix), but does not uniquely define the Coriolis matrix (Slotine & Li, 1991). Choosing

$$C_{ij}(\mathbf{q}, \dot{\mathbf{q}}) = \sum_{kl} a_l^{(K)}\frac{1}{2}\left[\frac{\partial M_{ij}^l(\mathbf{q})}{\partial q_k} - \left(\frac{\partial M_{kj}^l(\mathbf{q})}{\partial q_i} - \frac{\partial M_{ki}^l(\mathbf{q})}{\partial q_j}\right)\right]\dot{q}_k$$

preserves the Coriolis force $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ and ensures that $\dot{\mathbf{H}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is a skew-symmetric matrix. In matrix notation, the dynamics are then given by

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{u},$$

with the potential force $\mathbf{g}(\mathbf{q}) = \sum_l a_l^{(P)} \nabla_{\mathbf{q}} \phi^l(\mathbf{q})$. Defining $\mathbf{s}$ and $\dot{\mathbf{q}}_r$ as $\mathbf{s} = \left(\frac{d}{dt} + \lambda\right)\tilde{\mathbf{q}} = \dot{\mathbf{q}} - \dot{\mathbf{q}}_r$, these dynamics can be equivalently rewritten as

$$\mathbf{H}(\mathbf{q})\dot{\mathbf{s}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{s} = \mathbf{u} - (\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}}_r + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_r + \mathbf{g}(\mathbf{q})). \tag{4.6}$$

Observe that because the Lagrangian was linearly parameterized, the resulting dynamics are also linearly parameterized. Defining the known basis functions,

$$Y_{il}^{(P)} = \frac{\partial \phi^l(\mathbf{q})}{\partial q_i},$$

$$Y_{il}^{(K)} = \sum_{kj} \frac{1}{2}\left[\frac{\partial M_{ij}^l(\mathbf{q})}{\partial q_k} - \left(\frac{\partial M_{kj}^l(\mathbf{q})}{\partial q_i} - \frac{\partial M_{ki}^l(\mathbf{q})}{\partial q_j}\right)\right]\dot{q}_k \dot{q}_{r,j} + \sum_j M_{ij}^l \ddot{q}_{r,j},$$

we can write equation 4.6 as

$$\mathbf{H}(\mathbf{q})\dot{\mathbf{s}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{s} = \mathbf{u} - \mathbf{Y}^{(P)}\mathbf{a}^{(P)} - \mathbf{Y}^{(K)}\mathbf{a}^{(K)}.$$

For $\mathbf{K} > 0$ a positive-definite matrix and for parameter estimates $\hat{\mathbf{a}}^{(P)}$ and $\hat{\mathbf{a}}^{(K)}$, taking $\mathbf{u} = -\mathbf{K}\mathbf{s} + \mathbf{Y}^{(p)}\hat{\mathbf{a}}^{(P)} + \mathbf{Y}^{(K)}\hat{\mathbf{a}}^{(K)}$ leads to

$$\mathbf{H}(\mathbf{q})\dot{\mathbf{s}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{s} = -\mathbf{K}\mathbf{s} + \mathbf{Y}^{(P)}\tilde{\mathbf{a}}^{(P)} + \mathbf{Y}^{(K)}\tilde{\mathbf{a}}^{(K)}.$$

The proof in Slotine and Li (1987) can now be directly extended. For $\psi^{(K)}$, $\psi^{(P)}$ strictly convex functions and $\gamma_K > 0$, $\gamma_P > 0$ positive gains, the Lyapunov-like function

$$V = \frac{1}{2}\mathbf{s}^{\mathrm{T}}\mathbf{H}(\mathbf{q})\mathbf{s} + \frac{1}{\gamma_K} d_{\psi^{(K)}}\left(\mathbf{a}^{(K)} \,\|\, \hat{\mathbf{a}}^{(K)}\right) + \frac{1}{\gamma_P} d_{\psi^{(P)}}\left(\mathbf{a}^{(P)} \,\|\, \hat{\mathbf{a}}^{(P)}\right),$$

shows stability of the adaptation laws

$$\dot{\hat{\mathbf{a}}}^{(K)} = -\gamma_K \left(\nabla^2 \psi^{(K)}\left(\hat{\mathbf{a}}^{(K)}\right)\right)^{-1} \left[\mathbf{Y}^{(K)}\right]^{\mathrm{T}} \mathbf{s},$$

$$\dot{\hat{\mathbf{a}}}^{(P)} = -\gamma_P \left(\nabla^2 \psi^{(P)}\left(\hat{\mathbf{a}}^{(P)}\right)\right)^{-1} \left[\mathbf{Y}^{(P)}\right]^{\mathrm{T}} \mathbf{s},$$

after an application of Barbalat's lemma (lemma 1 in this article) and using skew-symmetry of $\dot{\mathbf{H}} - 2\mathbf{C}$ to eliminate $\frac{1}{2}\mathbf{s}^{\mathrm{T}}\dot{\mathbf{H}}\mathbf{s}$. In physical applications, dimensions or relative scaling of the components of $\hat{\mathbf{a}}^{(K)}$ and $\hat{\mathbf{a}}^{(P)}$ can be handled as described in remarks 6 and 11.

As in section 4.2, by using an l_1 approximation for ψ, this approach may find sparse, interpretable models of the kinetic and potential energies. Estimating the potential energy directly may in some cases lead to simpler parameterizations than estimating the resulting forces.

If more structure in the inertia matrix is known, for example, that it depends only on a few unknown parameters, it may still be approximated using the usual Slotine and Li controller. The external forces can then be estimated by directly estimating the corresponding potential that generates them.

4.4 Regularized Adaptive Observer Design. In many physical and engineering systems, only a low-dimensional output of the system $\mathbf{y}(\mathbf{x}) \in \mathbb{R}^m$ is available for measurement. Assuming that $\mathbf{y}(\mathbf{x}) = \mathbf{C}\mathbf{x}$ is a linear readout for some known matrix $\mathbf{C} \in \mathbb{R}^{m\times n}$, we now show that the tools of sections 4.2 and 4.3 can be used to design regularized adaptive observers for the full system state. Assume that the true system dynamics satisfies

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{c}(t) = \mathbf{Y}(\mathbf{y}(\mathbf{x}))\mathbf{a} + \mathbf{c}(t),$$

with $\mathbf{a} \in \mathbb{R}^p$ a vector of unknown parameters and where the known regressor matrix $\mathbf{Y} \in \mathbb{R}^{n\times p}$ only depends on the system output $\mathbf{y}(\mathbf{x})$. Consider the adaptive observer,

$$\dot{\hat{\mathbf{x}}} = \mathbf{Y}(\hat{\mathbf{y}})\hat{\mathbf{a}} + \mathbf{c}(t) + \mathbf{g}(\hat{\mathbf{y}}) - \mathbf{g}(\mathbf{y}),$$
$$\dot{\hat{\mathbf{a}}} = -\gamma \left(\nabla^2\psi(\hat{\mathbf{a}})\right)^{-1} \mathbf{Y}^{\mathrm{T}}(\hat{\mathbf{y}})\mathbf{C}^{\mathrm{T}}\boldsymbol{\Gamma}\tilde{\mathbf{y}},$$

with $\gamma > 0$ a positive learning rate, $\hat{\mathbf{y}} = \mathbf{y}(\hat{\mathbf{x}})$, ψ a strongly convex potential function, and $\boldsymbol{\Gamma}$ a positive-definite matrix. The Lyapunov-like function

$$V = \frac{1}{2}\tilde{\mathbf{y}}^{\mathrm{T}}\boldsymbol{\Gamma}\tilde{\mathbf{y}} + \frac{1}{\gamma}d_\psi(\mathbf{a} \,\|\, \hat{\mathbf{a}}),$$

has time derivative

$$\begin{aligned}
\dot{V} &= \tilde{\mathbf{y}}^{\mathrm{T}}\boldsymbol{\Gamma}\mathbf{C}\left(\mathbf{Y}(\hat{\mathbf{y}})\tilde{\mathbf{a}} + \left[\mathbf{Y}(\hat{\mathbf{y}}) - \mathbf{Y}(\mathbf{y})\right]\mathbf{a} + \mathbf{g}(\hat{\mathbf{y}}) - \mathbf{g}(\mathbf{y})\right) - \tilde{\mathbf{y}}^{\mathrm{T}}\boldsymbol{\Gamma}\mathbf{C}\mathbf{Y}(\hat{\mathbf{y}})\tilde{\mathbf{a}}, \\
&= \tilde{\mathbf{y}}^{\mathrm{T}}\boldsymbol{\Gamma}\mathbf{C}\left(\left[\mathbf{Y}(\hat{\mathbf{y}}) - \mathbf{Y}(\mathbf{y})\right]\mathbf{a} + \mathbf{g}(\hat{\mathbf{y}}) - \mathbf{g}(\mathbf{y})\right), \\
&= \tilde{\mathbf{y}}^{\mathrm{T}}\left(\boldsymbol{\Gamma}\mathbf{C}\int_0^1 \left(\frac{\partial \mathbf{Y}(\mathbf{y} + s\tilde{\mathbf{y}})\mathbf{a}}{\partial \mathbf{y}} + \frac{\partial \mathbf{g}(\mathbf{y} + s\tilde{\mathbf{y}})}{\partial \mathbf{y}}\right) ds\right)\tilde{\mathbf{y}},
\end{aligned}$$

which shows that a sufficient condition for convergence of $\tilde{\mathbf{y}} \to 0$ is

$$\mathbf{\Gamma C}\left(\frac{\partial \mathbf{Y(y)a}}{\partial \mathbf{y}} + \frac{\partial \mathbf{g(y)}}{\partial \mathbf{y}}\right) + \left(\frac{\partial \mathbf{Y(y)a}}{\partial \mathbf{y}} + \frac{\partial \mathbf{g(y)}}{\partial \mathbf{y}}\right)^{\mathrm{T}} \mathbf{C}^{\mathrm{T}}\mathbf{\Gamma} \leq -\lambda \mathbf{\Gamma}$$

uniformly in $\mathbf{y}$ for some contraction rate $\lambda > 0$. A natural choice of $\mathbf{g(y)}$ to satisfy this condition with $\mathbf{\Gamma} = \mathbf{I}$ is $\mathbf{g(y)} = -k\mathbf{C}^{\mathrm{T}}\mathbf{y}$ for some $k > 0$ if $\mathbf{CC}^{\mathrm{T}}$ is full rank. The requirement is equivalent to contraction of the unknown output dynamics,

$$\dot{\mathbf{y}} = \mathbf{CY(y)a} + \mathbf{Cg(y)} + \mathbf{Cc}(t),$$

in the metric $\mathbf{\Gamma}$. Under suitable observability assumptions on the system, convergence of $\hat{\mathbf{y}}$ to $\mathbf{y}$ ensures that $\hat{\mathbf{x}}$ converges to $\mathbf{x}$, and hence that the full system state can be observed (Luenberger, 1979).

As in section 4.1, this discussion demonstrates a separation theorem for adaptive observer design. If an observer can be designed for the true system with unknown parameters, then the same observer can be made adaptive by incorporating the adaptation law presented in this section. Convergence properties then depend only on the true system with feedback and are independent of the parameter estimator. The results of section 3.1 show that the choice of ψ can be used to regularize the observer model while maintaining provable reconstruction of the full system state.

4.5 Regularized Dynamics Prediction for Recurrent Neural Networks. Consider a recurrent neural network model,

$$\tau \dot{\mathbf{x}} = -\mathbf{x} + \boldsymbol{\sigma}(\mathbf{\Theta x}), \tag{4.7}$$

with $\mathbf{x} \in \mathbb{R}^n$ a vector of neuron firing rates, $\mathbf{\Theta} \in \mathbb{R}^{n \times n}$ the synaptic weights, $\boldsymbol{\sigma}(\mathbf{\Theta x})$ the postsynaptic potentials, and $\tau > 0$ a relaxation timescale. Let $\boldsymbol{\sigma}(\cdot)$ be an elementwise Lipschitz and monotonic activation function:

$$\begin{aligned} \boldsymbol{\sigma}(\mathbf{x})_i &= \sigma_i(x_i), \\ |\sigma_i(x) - \sigma_i(y)| &\leq L_i|x - y|, \\ (x - y)(\sigma_i(x) - \sigma_i(y)) &\geq 0. \end{aligned}$$

These requirements are satisfied by common activation functions such as the ReLU, softplus, tanh, and sigmoid. For ψ a strongly convex function on $n \times n$ matrices or vectors in $\mathbb{R}^{n^2}$ and $\gamma > 0$ a positive gain, consider the regularized adaptive dynamics predictor for equation 4.7,

$$\tau \dot{\hat{\mathbf{x}}} = -\hat{\mathbf{x}} + \boldsymbol{\sigma}\left(\widehat{\mathbf{\Theta}}\mathbf{x}\right) + k(\mathbf{x} - \hat{\mathbf{x}}), \tag{4.8}$$

$$\dot{\widehat{\mathbf{\Theta}}} = -\gamma\left(\nabla^2\psi\left(\widehat{\mathbf{\Theta}}\right)\right)^{-1}\left(\boldsymbol{\sigma}\left(\widehat{\mathbf{\Theta}}\mathbf{x}\right) - \boldsymbol{\sigma}(\mathbf{\Theta x})\right)\mathbf{x}^{\mathrm{T}}. \tag{4.9}$$

In equation 4.8 the true vector of firing rates $\mathbf{x}$ is used underneath the application of $\boldsymbol{\sigma}(\cdot)$ in the $\dot{\hat{\mathbf{x}}}$ dynamics. The update law, equation 4.9, can be seen as the vector-valued generalization of the algorithm considered in remark 4. The Lyapunov-like function

$$V = \frac{1}{\gamma} d_\psi(\boldsymbol{\Theta} \,\|\, \widehat{\boldsymbol{\Theta}}),$$

has time derivative

$$\begin{aligned}
\dot{V} &= -\sum_{ij} \widetilde{\Theta}_{ij} \left(\boldsymbol{\sigma}\left(\widehat{\boldsymbol{\Theta}}\mathbf{x}\right) - \boldsymbol{\sigma}\left(\boldsymbol{\Theta}\mathbf{x}\right)\right)_i x_j, \\
&= -\sum_{ij} \widetilde{\Theta}_{ij} \left(\sigma_i\left(\sum_k \widehat{\Theta}_{ik} x_k\right) - \sigma_i\left(\sum_k \Theta_{ik} x_k\right)\right) x_j, \\
&= -\sum_{i} \left(\sum_k \widetilde{\Theta}_{ik} x_k\right) \left(\sigma_i\left(\sum_k \widehat{\Theta}_{ik} x_k\right) - \sigma_i\left(\sum_k \Theta_{ik} x_k\right)\right), \\
&\leq -\sum_i \frac{1}{L_i} \left(\sigma_i\left(\sum_k \widehat{\Theta}_{ik} x_k\right) - \sigma_i\left(\sum_k \Theta_{ik} x_k\right)\right)^2, \\
&\leq -\frac{1}{\max_k L_k} \left\|\boldsymbol{\sigma}\left(\widehat{\boldsymbol{\Theta}}\mathbf{x}\right) - \boldsymbol{\sigma}\left(\boldsymbol{\Theta}\mathbf{x}\right)\right\|_2^2 \leq 0.
\end{aligned}$$

Integrating the above inequality shows that $\left[\boldsymbol{\sigma}\left(\widehat{\boldsymbol{\Theta}}\mathbf{x}\right) - \boldsymbol{\sigma}\left(\boldsymbol{\Theta}\mathbf{x}\right)\right]$ is an $\mathcal{L}_2$ signal and, hence, that each component $\left[\sigma_i\left(\widehat{\boldsymbol{\Theta}}\mathbf{x}\right) - \sigma_i\left(\boldsymbol{\Theta}\mathbf{x}\right)\right]$ is also an $\mathcal{L}_2$ signal. The error dynamics

$$\dot{\mathbf{e}} = -(k+1)\mathbf{e} + \boldsymbol{\sigma}\left(\widehat{\boldsymbol{\Theta}}\mathbf{x}\right) - \boldsymbol{\sigma}\left(\boldsymbol{\Theta}\mathbf{x}\right),$$

shows that each component e_i is a low-pass filter of each component of the function approximation error $\left[\sigma_i\left(\widehat{\boldsymbol{\Theta}}\mathbf{x}\right) - \sigma_i\left(\boldsymbol{\Theta}\mathbf{x}\right)\right]$. Applying lemma 2 shows that $\mathbf{e} \to 0$. This approach could be used, for example, for identifying regularized low-dimensional models in computational neuroscience. Our results are similar to those of Foster, Rakhlin, and Sarkar (2020), but handle a mirror descent or natural gradient extension valid in the continuous-time deterministic setting.

The adaptation law, equation 4.9, cannot be implemented directly through a PI form. However, it can be well approximated, for example, by the PI construction

$$\dot{\bar{\mathbf{x}}} = \lambda\left(\mathbf{x} - \bar{\mathbf{x}}\right),$$
$$\nabla\psi\left(\widehat{\boldsymbol{\Theta}}\right) = \gamma\left(\overline{\boldsymbol{\Theta}} - \mathbf{e}\bar{\mathbf{x}}^{\mathrm{T}}\right),$$
$$\dot{\overline{\boldsymbol{\Theta}}} = -\left(k+1\right)\mathbf{e}\bar{\mathbf{x}}^{\mathrm{T}} + \lambda\mathbf{e}\left(\mathbf{x} - \bar{\mathbf{x}}\right)^{\mathrm{T}},$$

for $\lambda > 0$ a positive gain ensuring $\bar{\mathbf{x}} \approx \mathbf{x}$.

5 Velocity Gradient Algorithms and the Bregman Lagrangian

In this section, we provide background material on the velocity gradient formalism (Andrievskii et al., 1988; Fradkov, 1980, 1986; Fradkov et al., 1999) and the Bregman Lagrangian (Betancourt et al., 2018; Wibisono et al., 2016; Wilson et al., 2016).

5.1 Velocity Gradient Algorithms. We now provide a brief introduction to a class of adaptive control methods known as velocity gradient algorithms (Andrievskii et al., 1988; Fradkov, 1980, 1986; Fradkov et al., 1999). In their most basic form, they are specified by a "local" goal functional $Q(\mathbf{x}, t) : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}$ we would like to drive to zero. The adaptation law is defined as

$$\dot{\hat{\mathbf{a}}} = -\mathbf{P}\nabla_{\hat{\mathbf{a}}}\dot{Q}(\mathbf{x}, \hat{\mathbf{a}}, t), \tag{5.1}$$

where $\mathbf{P} = \mathbf{P}^{\mathrm{T}} > 0$ is a positive definite matrix of learning rates of appropriate dimension and $\dot{Q}(\mathbf{x}, \hat{\mathbf{a}}, t) = \left(\nabla_{\mathbf{x}}Q(\mathbf{x}, t)\right)^{\mathrm{T}}\dot{\mathbf{x}} + \frac{\partial Q(\mathbf{x},t)}{\partial t}$. Intuitively, while the goal functional $Q(\mathbf{x}, t)$ may only depend on the control parameters $\hat{\mathbf{a}}$ indirectly through $\mathbf{x}$, its time derivative will depend explicitly on $\hat{\mathbf{a}}$ through $\dot{\mathbf{x}}$.[6] The adaptation law, equation 5.1, ensures that $\hat{\mathbf{a}}$ moves in a direction that instantaneously decreases $\dot{Q}(\mathbf{x}, \hat{\mathbf{a}}, t)$. Under the conditions specified by assumptions 4 to 6, this causes $\dot{Q}(\mathbf{x}, \hat{\mathbf{a}}, t)$ to be negative for long enough to accomplish the control goal (Fradkov et al., 1999).

Assumption 4. *$Q(\mathbf{x}, t)$ is nonnegative and radially unbounded, so that $Q(\mathbf{x}, t) \geq 0$ for all $\mathbf{x}$, t and $Q(\mathbf{x}, t) \to \infty$ when $\|\mathbf{x}\| \to \infty$. $Q(\mathbf{x}, t)$ is uniformly continuous in t whenever $\mathbf{x}$ is bounded.*

Assumption 5. *There exists an ideal set of control parameters $\mathbf{a}$ such that the origin of the system 2.1 is globally asymptotically stable when the control is evaluated at $\mathbf{a}$. Furthermore, $Q(\mathbf{x}, t)$ is a Lyapunov function for the system when the control is evaluated at $\mathbf{a}$. That is, there exists a strictly increasing function ρ such that $\rho(0) = 0$ with $\dot{Q}(\mathbf{x}, \mathbf{a}, t) \leq -\rho(Q)$.*

[6] It will also depend on $\mathbf{a}$, but we suppress this dependence for notational simplicity.

Assumption 6. *The time derivative of Q is convex in the control parameters $\hat{\mathbf{a}}$, that is,*

$$\dot{Q}(\mathbf{x}, \mathbf{a}_1, t) \geq \dot{Q}(\mathbf{x}, \mathbf{a}_2, t) + (\mathbf{a}_1 - \mathbf{a}_2)^{\mathrm{T}} \nabla_{\mathbf{a}_2} \dot{Q}(\mathbf{x}, \mathbf{a}_2, t), \tag{5.2}$$

is satisfied for all $\mathbf{a}_1$ and $\mathbf{a}_2$.

The properties of equation 5.1 are summarized in the following proposition (Fradkov et al., 1999).

Proposition 3. *Consider the local velocity gradient algorithm equation 5.1, under assumptions 4 to 6. Then all solutions $(\mathbf{x}(t), \hat{\mathbf{a}}(t))$ of equations 2.1 and 5.1 remain bounded, and*

$$\lim_{t \to \infty} Q(\mathbf{x}(t), t) = 0$$

for all $\mathbf{x}(0) \in \mathbb{R}^n$.

The proof follows by consideration of the Lyapunov-like function $V = Q + \frac{1}{2}\tilde{\mathbf{a}}^{\mathrm{T}}\mathbf{P}^{-1}\tilde{\mathbf{a}}$.

Remark 13. If $Q(\mathbf{x}, t)$ is chosen so that $\dot{Q}(\mathbf{x}, \hat{\mathbf{a}}, t)$ depends on $\hat{\mathbf{a}}$ only through $\hat{f}(\mathbf{x}, \hat{\mathbf{a}}, t)$ and $f(\mathbf{x}, \mathbf{a}, t)$ is linearly parameterized, then assumption 6 will immediately be satisfied by convexity of affine functions. Indeed, consider defining the goal functional $Q(\mathbf{x}, t) = \frac{1}{2}s(\mathbf{x}, t)^2$ for system 2.1 where s depends on t through $x_d(t)$. It is clear that this proposed goal functional satisfies assumptions 4 and 5 for bounded $x_d(t)$. Then $\dot{Q}(\mathbf{x}, \hat{\mathbf{a}}, t) = -\eta s(\mathbf{x}, t)^2 + s\tilde{f}(\mathbf{x}, \hat{\mathbf{a}}, \mathbf{a}, t)$, and equation 5.1 exactly recovers the Slotine and Li controller, equation 2.8.

Remark 14. An alternative perspective on velocity gradient algorithms can be found by using the expression $\dot{Q}(\mathbf{x}, \hat{\mathbf{a}}, t) = (\nabla_{\mathbf{x}} Q(\mathbf{x}, t))^{\mathrm{T}} \dot{\mathbf{x}} + \frac{\partial Q(\mathbf{x}, t)}{\partial t}$. Assume that $\dot{\mathbf{x}} = \mathbf{u} - \mathbf{Y}(\mathbf{x}, t)\mathbf{a}$, and set $\mathbf{u} = \mathbf{Y}(\mathbf{x}, t)\hat{\mathbf{a}} + \mathbf{u}_d$ where $\mathbf{u}_d$ ensures that $\mathbf{x}(t) \to \mathbf{x}_d(t)$ for $\hat{\mathbf{a}} = \mathbf{a}$. Then $\nabla_{\hat{\mathbf{a}}} \dot{Q}(\mathbf{x}, \hat{\mathbf{a}}, t) = \mathbf{Y}^{\mathrm{T}} \nabla_{\mathbf{x}} Q(\mathbf{x}, t)$. This shows that the adaptation law $\dot{\hat{\mathbf{a}}} = -\mathbf{P}\nabla_{\hat{\mathbf{a}}} \dot{Q}(\mathbf{x}, \hat{\mathbf{a}}, t) = -\mathbf{P}\mathbf{Y}(\mathbf{x}, t)^{\mathrm{T}} \nabla_{\mathbf{x}} Q(\mathbf{x}, t)$ transforms the gradient of $Q(\mathbf{x}, t)$ with respect to $\mathbf{x}$ by premultiplication by the regressor $\mathbf{Y}(\mathbf{x}, t)^{\mathrm{T}}$. This interpretation applies to the observers and dynamics predictors designed in section 4, as well as the adaptation law for contracting systems developed in Lopez and Slotine (2021). Conversely, this perspective shows that if a Lyapunov function $V(\mathbf{x}, t)$ is known for a nominal system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$, then the control input $\mathbf{u} = \mathbf{Y}(\mathbf{x}, t)\hat{\mathbf{a}}$ with adaptation law $\dot{\hat{\mathbf{a}}} = -\mathbf{P}\mathbf{Y}^{\mathrm{T}} \nabla_{\mathbf{x}} V(\mathbf{x}, t)$ will return the perturbed system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) + \mathbf{u} - \mathbf{Y}(\mathbf{x}, t)\mathbf{a}$ back to its nominal behavior.

Rather than a local functional, one may instead specify an integral goal functional of the form $Q(\mathbf{x}, \hat{\mathbf{a}}, t) = \int_0^t R(\mathbf{x}(t'), \hat{\mathbf{a}}(t'), t')dt'$. In this case, equation 5.1 takes the form

$$\dot{\hat{\mathbf{a}}} = -\mathbf{P}\nabla_{\hat{\mathbf{a}}} R(\mathbf{x}, \hat{\mathbf{a}}, t). \tag{5.3}$$

Equation 5.3 is a gradient flow algorithm on the loss function $R(\mathbf{x}, \hat{\mathbf{a}}, t)$. We now replace assumptions 4 and 5 by a slightly modified setting.

Assumption 7. *R is a nonnegative function and $R(\mathbf{x}(t), \hat{\mathbf{a}}(t), t)$ is uniformly continuous in t for bounded $\mathbf{x}$ and $\hat{\mathbf{a}}$. Furthermore, $\nabla_{\hat{\mathbf{a}}} R(\mathbf{x}, \hat{\mathbf{a}}, t)$ is locally bounded in $\mathbf{x}$ and $\hat{\mathbf{a}}$ uniformly in t.*

Assumption 8. *There exists an ideal set of controller parameters $\mathbf{a}$ and a scalar function μ such that $\int_0^\infty \mu(t')dt' < \infty$, $\lim_{t\to\infty} \mu(t) = 0$, and $R(\mathbf{x}(t), \mathbf{a}, t) \le \mu(t)$ for all t.*

The properties of algorithm 5.3 are summarized in the following proposition (Fradkov et al., 1999).

Proposition 4. *Consider the integral velocity gradient algorithm 5.3 where the goal functional Q satisfies assumptions 6 to 8. Then $Q(\mathbf{x}(t); t) \le \alpha$, where*

$$\alpha = \frac{1}{2}\tilde{\mathbf{a}}(0)^{\mathrm{T}}\mathbf{P}^{-1}\tilde{\mathbf{a}}(0) + \int_0^\infty \mu(t')dt',$$

and $\int R(\mathbf{x}(t'), \hat{\mathbf{a}}(t'), t')dt' < \infty$ over the maximal interval of existence of $\mathbf{x}$. Furthermore, $R(\mathbf{x}, \hat{\mathbf{a}}, t) \to 0$ for any bounded solution $\mathbf{x}(t)$.

The proof follows by consideration of the Lyapunov-like function $V = \int_0^t R(\mathbf{x}(t'), \hat{\mathbf{a}}(t'), t')dt' + \frac{1}{2}\tilde{\mathbf{a}}^{\mathrm{T}}\mathbf{P}^{-1}\tilde{\mathbf{a}} + \int_t^\infty \mu(t')dt'$.

Integral functionals allow the specification of a control goal that depends on all past data. $R(\mathbf{x}, \hat{\mathbf{a}}, t)$ is chosen so that it does not necessarily depend on the structure of the dynamics but depends explicitly on $\hat{\mathbf{a}}$. Local functionals, on the other hand, result in adaptation laws that *do* have an explicit dependence on the dynamics through the appearance of the term $\left(\frac{\partial Q}{\partial \mathbf{x}}\right)^{\mathrm{T}} \dot{\mathbf{x}}$ in $\dot{Q}(\mathbf{x}, \hat{\mathbf{a}}, t)$.

Integral functionals can be particularly useful if $R(\mathbf{x}, \hat{\mathbf{a}}, t) \to 0$ implies the desired control goal. In this work, we focus on the choice $R(\mathbf{x}, \hat{\mathbf{a}}, t) = \frac{1}{2}\tilde{f}(\mathbf{x}, \hat{\mathbf{a}}, \mathbf{a}, t)^2$, which will require a PI form as described in section 2 in the context of Tyukin's algorithm.[7] In particular, note that for this choice of R, the result of proposition 4 implies that $\tilde{f} \in \mathcal{L}_2$ over the maximal interval of

[7] Indeed, Tyukin's algorithm can be seen as an integral velocity gradient algorithm with the pseudogradient modification described in section 2.

existence of $\mathbf{x}$. For some error models, this is enough to ensure that $\mathbf{x} \in \mathcal{L}_\infty$, and hence that $\tilde{f}(\mathbf{x}, \hat{\mathbf{a}}, \mathbf{a}, t) \to 0$ and $\mathbf{x} \to \mathbf{x}_d$.[8]

Goal functionals can also be written as a sum of local and integral functionals with similar guarantees, and these approaches will lead to composite algorithms in the subsequent sections. (See Fradkov et al., 1999, chap. 3 for more detail.)

Remark 15. Following the developments of section 3, we can immediately prove analogous results for natural gradient or mirror descent–like velocity gradient algorithms. For local functionals, the adaptation law

$$\dot{\hat{\mathbf{a}}} = -\gamma \left(\nabla^2 \psi(\hat{\mathbf{a}})\right)^{-1} \nabla_{\hat{\mathbf{a}}} \dot{Q}(\mathbf{x}, \hat{\mathbf{a}}, t)$$

with $\gamma > 0$ a positive learning rate and ψ a strongly convex function will lead to the same conclusions as proposition 3 under the same conditions. The proof follows by consideration of the Lyapunov-like function,

$$V = Q(\mathbf{x}, t) + d_\psi(\mathbf{a} \,\|\, \hat{\mathbf{a}}).$$

Similarly, the Lyapunov-like function

$$V = \int_0^t R(\mathbf{x}(t'), \hat{\mathbf{a}}(t'), t')dt' + d_\psi(\mathbf{a} \,\|\, \hat{\mathbf{a}}) + \int_t^\infty \mu(t')dt'$$

shows that the same conclusions as in proposition 4 hold under the same conditions for the integral natural velocity gradient algorithm,

$$\dot{\hat{\mathbf{a}}} = -\gamma \left(\nabla^2 \psi(\hat{\mathbf{a}})\right)^{-1} \nabla_{\hat{\mathbf{a}}} R(\mathbf{x}, \hat{\mathbf{a}}, t).$$

In both cases, the choice of ψ offers a principled way to regularize velocity gradient algorithms.

5.2 The Bregman Lagrangian and Accelerated Optimization Algorithms. In Wibisono et al. (2016), the Bregman Lagrangian was shown to generate a suite of accelerated optimization algorithms in continuous time by appealing to the Euler Lagrange equations through the principle of least action. In its original form, the Bregman Lagrangian is given by

$$\mathcal{L}(\mathbf{x}, \dot{\mathbf{x}}, t) = e^{\overline{\alpha}+\overline{\gamma}} \left(d_\psi\left(\mathbf{x} + e^{-\overline{\alpha}}\dot{\mathbf{x}} \,\middle\|\, \mathbf{x}\right) - e^{\overline{\beta}} f(\mathbf{x}) \right). \tag{5.4}$$

[8] See, for example, lemma 2, which shows that our error model 2.4 has this property.

In equation 5.4, $f(\mathbf{x})$ is the loss function to be optimized, and $\psi(\mathbf{x})$ is a strongly convex function. We take $\psi(\cdot) = \frac{1}{2}\|\cdot\|_2^2$ in section 6 and consider extensions to arbitrary ψ in section 7. Allowing for arbitrary ψ extends the algorithms presented in section 6 to the natural gradient-like setting of section 2.3.

The quantities $\overline{\alpha}(t) : \mathbb{R}_+ \to \mathbb{R}$, $\overline{\beta}(t) : \mathbb{R}_+ \to \mathbb{R}$, and $\overline{\gamma}(t) : \mathbb{R}_+ \to \mathbb{R}$ in equation 5.4 are arbitrary time-dependent functions that will ultimately set the damping and learning rates in the second-order Euler Lagrange dynamics. To generate accelerated optimization algorithms, Wibisono, Wilson, and Jordan (2006) required two ideal scaling conditions: $\dot{\overline{\beta}} \leq e^{\overline{\alpha}}$ and $\dot{\overline{\gamma}} = e^{\overline{\alpha}}$. These conditions originate from the Euler Lagrange equations, where the second is used to eliminate an unwanted term, and a Lyapunov argument, where the first is used to ensure decrease of a chosen Lyapunov function.

Gaudio et al. (2019) recently utilized the Bregman Lagrangian to derive a momentum-like adaptive control algorithm. To do so, they defined $\overline{\alpha} = \log(\beta\mathcal{N})$, $\overline{\beta} = \log\left(\frac{\gamma}{\beta\mathcal{N}}\right)$, and $\overline{\gamma} = \int e^{\overline{\alpha}} dt$[9]. Here, $\gamma \geq 0$ and $\beta \geq 0$ are nonnegative scalar hyperparameters, and $\mathcal{N} = \mathcal{N}(t)$ is a signal chosen based on the system. With these definitions, choosing the Euclidean norm $\psi(\cdot) = \frac{1}{2}\|\cdot\|^2$, and modifying the Bregman Lagrangian presented in Gaudio et al. (2019) to the adaptive control framework defined in section 2, equation 5.4 becomes

$$\mathcal{L}\left(\hat{\mathbf{a}}, \dot{\hat{\mathbf{a}}}, t\right) = e^{\int_0^t \beta\mathcal{N}(t)dt} \frac{1}{\beta\mathcal{N}} \left(\frac{1}{2}\dot{\hat{\mathbf{a}}}^{\mathrm{T}}\dot{\hat{\mathbf{a}}} - \gamma\beta\mathcal{N}\frac{d}{dt}\left[\frac{1}{2}s^2\right]\right). \tag{5.5}$$

Comparing equations 5.4 and 5.5, it is clear that the loss function $f(\mathbf{x})$ in equation 5.4 has been replaced by $\frac{d}{dt}\frac{1}{2}s^2$ in equation 5.5. Following remark 13, this is precisely the $\dot{Q}$ velocity gradient functional that gives rise to the Slotine and Li controller. For equation 5.5, the Euler-Lagrange equations lead to the adaptation law,

$$\ddot{\hat{\mathbf{a}}} + \dot{\hat{\mathbf{a}}}\left(\beta\mathcal{N} - \frac{\dot{\mathcal{N}}}{\mathcal{N}}\right) = -\gamma\beta\mathcal{N}\mathbf{Y}^{\mathrm{T}}s. \tag{5.6}$$

Equation 5.6 may be understood as a modification of the Slotine and Li adaptive controller to incorporate momentum and time-dependent damping. This equation may also be rewritten as two first-order systems,

[9] Note that these conditions validate the second ideal scaling condition but not the first. As mentioned above, the first ideal scaling condition is required only by the choice of Lyapunov function in the original work, which was used to derive convergence rates for optimization algorithms (Wibisono et al., 2016). In this sense, it is not strictly required for adaptive control.

$$\dot{\hat{\mathbf{v}}} = -\gamma \mathbf{Y}^{\mathrm{T}} s, \tag{5.7}$$

$$\dot{\hat{\mathbf{a}}} = \beta \mathcal{N} (\hat{\mathbf{v}} - \hat{\mathbf{a}}), \tag{5.8}$$

which are useful for proving stability. The properties of equation 5.6 are summarized in the following proposition.

Proposition 5. *Consider the higher-order adaptation algorithm, equation 5.6, with* $\mathcal{N} = 1 + \mu \|\mathbf{Y}\|^2$ *and* $\mu > \frac{\gamma}{\eta\beta}$. *Then all trajectories* $(\mathbf{x}, \hat{\mathbf{v}}, \hat{\mathbf{a}})$ *remain bounded,* $s \in \mathcal{L}_\infty \cap \mathcal{L}_2$, $(\hat{\mathbf{a}} - \hat{\mathbf{v}}) \in \mathcal{L}_2$, $s \to 0$ *and* $\mathbf{x} \to \mathbf{x}_d$.

The proof follows by consideration of the Lyapunov-like function $V = \frac{1}{2}\left(s^2 + \frac{1}{\gamma}\|\tilde{\mathbf{v}}\|^2 + \frac{1}{\gamma}\|\hat{\mathbf{v}} - \hat{\mathbf{a}}\|^2\right)$.

Remark 16. The transformation to a system of two first-order equations may seem somewhat ad-hoc, but it follows immediately by consideration of the non-Euclidean Bregman Lagrangian, equation 5.4. Indeed, it is easy to check that $\hat{\mathbf{v}} = \hat{\mathbf{a}} + \frac{\dot{\hat{\mathbf{a}}}}{\beta\mathcal{N}}$, which is precisely the adaptive control equivalent of $\mathbf{x} + e^{-\overline{\alpha}}\dot{\mathbf{x}}$ in the first argument of $d_\psi(\cdot \,\|\, \cdot)$ in equation 5.4. The transformation is also readily apparent by use of the Bregman Hamiltonian

$$\mathcal{H}(\hat{\mathbf{a}}, \mathbf{p}) = \frac{1}{2}\beta\mathcal{N}e^{-\overline{\gamma}}\|\mathbf{p}\|^2 + \gamma e^{\overline{\gamma}}\left[\frac{d}{dt}\frac{1}{2}s^2\right], \tag{5.9}$$

which, via Hamilton's equations, leads to

$$\dot{\mathbf{p}} = -\frac{\partial \mathcal{H}}{\partial \hat{\mathbf{a}}} = -\gamma e^{\overline{\gamma}}\mathbf{Y}^{\mathrm{T}} s,$$

$$\dot{\hat{\mathbf{a}}} = \frac{\partial \mathcal{H}}{\partial \mathbf{p}} = \beta\mathcal{N}e^{-\overline{\gamma}}\mathbf{p}.$$

Defining $\hat{\mathbf{v}} = e^{-\overline{\gamma}}\mathbf{p} + \hat{\mathbf{a}}$ immediately leads to equations 5.7 and 5.8. This line of reasoning was recently investigated further by Gaudio, Annaswamy, Bolender, Lavretsky, and Gibson (2021). As is typical in classical mechanics, the Bregman Hamiltonian may be obtained from a Legendre transform of the Bregman Lagrangian. The Hamiltonian dynamics may be useful for discrete-time algorithm development through application of symplectic discretization techniques (Betancourt et al., 2018; França, Sulam, Robinson, & Vidal, 2019; Shi, Du, Su, & Jordan, 2019).

Remark 17. It is well known, for example from a passivity interpretation of the Lyapunov-like analysis (see, e.g., Slotine & Li, 1991), that the pure integrator in the standard Slotine and Li adaptation law, equation 2.8, can be replaced by any linear positive real transfer function containing a pure

integrator. The higher-order algorithms presented in this work are distinct from this approach, as most clearly seen by the state-dependent damping term in equation 5.6.

Remark 18. In Wibisono et al. (2016), the suggested Lyapunov function in the Euclidean setting is $V = \|\mathbf{x} + e^{-\overline{\alpha}}\dot{\mathbf{x}} - \mathbf{x}_*\|^2 + e^{\overline{\beta}} f(\mathbf{x})$, where $\mathbf{x}_*$ is the global optimum and $f(\mathbf{x})$ is the loss function. Noting that $\hat{\mathbf{v}}$ is the equivalent of $\mathbf{x} + e^{-\overline{\alpha}}\dot{\mathbf{x}}$ in the adaptive control context (see remark 16), we see that the Lyapunov-like function used to prove stability of the adaptive law, equation 5.6, is similar to that used to prove convergence in the optimization context. The loss function term $f(\mathbf{x})$ is replaced by $\frac{1}{2}s^2$, and it is necessary to add the term $\frac{1}{\gamma}\|\hat{\mathbf{v}} - \hat{\mathbf{a}}\|^2$.

6 Adaptation Laws with Momentum

In this section, we develop several new adaptation laws for both linearly and nonlinearly parameterized systems. We begin by noting that the Bregman Lagrangian generates velocity gradient algorithms with momentum. We prove some general conditions under which these momentum algorithms will achieve tracking. By analogy with integral velocity gradient functionals, we then derive a proportional-integral scheme to implement a first-order composite adaptation law (Slotine & Li, 1991) driven directly by the function approximation error rather than its filtered version. We subsequently fuse the generating functional for the composite law with the Bregman Lagrangian to construct a composite algorithm with momentum.

We then employ a connection between recent developments in isotonic regression—the GLMTron of Kakade et al. (2011), along with extensions due to Goel and Klivans (2017) and Goel et al. (2018)—and Tyukin's algorithm, equation 2.12, to derive momentum algorithms for nonlinearly parameterized systems. These momentum algorithms can be seen as the adaptive control equivalent of the GLMTron with momentum.

We follow this development by discussing a new form of high-order algorithm inspired by the elastic averaging stochastic gradient descent (EASGD) algorithm (Boffi & Slotine, 2020; Zhang, Choromanska, & LeCun, 2014). We subsequently demonstrate the capability of using time-varying learning rates with our presented algorithms (Slotine & Li, 1991).

6.1 Velocity Gradient Algorithms with Momentum. As noted in section 5.2, the Bregman Lagrangian (equation 5.5) that generates the higher-order algorithm in equation 5.6 contains the local velocity gradient functional $Q(\mathbf{x}, t) = \frac{1}{2}s(\mathbf{x}, t)^2$ that gives rise to the Slotine and Li controller (equation 2.8). Based on this observation, we define local and integral higher-order velocity gradient algorithms via the Euclidean Bregman Lagrangian. We begin with the local functional

$$\mathcal{L}\left(\hat{\mathbf{a}}, \dot{\hat{\mathbf{a}}}, t\right) = e^{\int_0^t \beta\mathcal{N}(t)dt} \frac{1}{\beta\mathcal{N}(t)} \left(\frac{1}{2}\dot{\hat{\mathbf{a}}}^{\mathrm{T}}\dot{\hat{\mathbf{a}}} - \gamma\beta\mathcal{N}(t)\frac{d}{dt}Q(\mathbf{x}, t)\right),$$

which generates the higher-order law

$$\ddot{\hat{\mathbf{a}}} + \dot{\hat{\mathbf{a}}}\left(\beta\mathcal{N} - \frac{\dot{\mathcal{N}}}{\mathcal{N}}\right) = -\gamma\beta\mathcal{N}\nabla_{\hat{\mathbf{a}}}\dot{Q}(\mathbf{x}, \hat{\mathbf{a}}, t). \tag{6.1}$$

Algorithm 6.1 can be rewritten as two first-order systems:

$$\dot{\hat{\mathbf{v}}} = -\gamma\nabla_{\hat{\mathbf{a}}}\dot{Q}(\mathbf{x}, \hat{\mathbf{a}}, t), \tag{6.2}$$

$$\dot{\hat{\mathbf{a}}} = \beta\mathcal{N}\left(\hat{\mathbf{v}} - \hat{\mathbf{a}}\right). \tag{6.3}$$

To achieve the control goal, we require the following technical assumption in addition to assumptions 4 and 6. This assumption replaces assumption 5 for first-order velocity gradient algorithms.

Assumption 9. *There exists a time-dependent signal $N(t)$ and nonnegative scalar values $\beta \geq 0$, $\mu \geq 0$ such that the time-derivative of the goal functional evaluated at the true parameters, $\dot{Q}(\mathbf{x}, \mathbf{a}, t)$, satisfies the following inequality:*

$$\dot{Q}(\mathbf{x}, \mathbf{a}, t) - \frac{\beta\mu}{\gamma}N(t)\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2 + 2\left(\hat{\mathbf{a}} - \hat{\mathbf{v}}\right)^{\mathrm{T}}\nabla_{\hat{\mathbf{a}}}\dot{Q}(\mathbf{x}, \hat{\mathbf{a}}, t) \leq -\rho(Q). \tag{6.4}$$

In equation 6.4, $\rho(\cdot)$ is positive definite, continuous in Q, and satisfies $\rho(0) = 0$.

Assumption 9 is a formal statement that we may "complete the square" on the left-hand side of 6.4. For example, for $\nabla_{\hat{\mathbf{a}}}\dot{Q}(\mathbf{x}, \hat{\mathbf{a}}, t) = \mathbf{Y}^{\mathrm{T}}s$ and for $\dot{Q}(\mathbf{x}, \mathbf{a}, t) = -\eta s^2$, we may choose $N = \|\mathbf{Y}^{\mathrm{T}}\|^2$.

With assumption 9 in hand, we can state the following proposition.

Proposition 6. *Consider algorithm 6.1 or its equivalent form, 6.2 and 6.3, and assume Q satisfies assumptions 4, 6, and 9. Then, all solutions $(\mathbf{x}(t), \hat{\mathbf{v}}(t), \hat{\mathbf{a}}(t))$ remain bounded, $(\hat{\mathbf{a}} - \hat{\mathbf{v}}) \in \mathcal{L}_2$, and $\lim_{t\to\infty} Q(\mathbf{x}(t); t) = 0$.*

Proof. Consider the Lyapunov-like function,

$$V = Q(\mathbf{x}, t) + \frac{1}{2\gamma}\tilde{\mathbf{v}}^{\mathrm{T}}\tilde{\mathbf{v}} + \frac{1}{2\gamma}\left(\hat{\mathbf{a}} - \hat{\mathbf{v}}\right)^{\mathrm{T}}\left(\hat{\mathbf{a}} - \hat{\mathbf{v}}\right). \tag{6.5}$$

Equation 6.5 implies that with $\mathcal{N}(t) = 1 + \mu N(t)$,

$$\dot{V} = \dot{Q}(\mathbf{x}, \hat{\mathbf{a}}, t) - \tilde{\mathbf{a}}^{\mathrm{T}}\nabla_{\hat{\mathbf{a}}}\dot{Q}(\mathbf{x}, \hat{\mathbf{a}}, t) - \frac{\beta}{\gamma}\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2 - \frac{\beta\mu}{\gamma}N(t)\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2$$
$$+ 2\left(\hat{\mathbf{a}} - \hat{\mathbf{v}}\right)^{\mathrm{T}}\nabla_{\hat{\mathbf{a}}}\dot{Q}(\mathbf{x}, \hat{\mathbf{a}}, t),$$

$$\leq \dot{Q}(\mathbf{x}, \mathbf{a}, t) - \frac{\beta}{\gamma}\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2 - \frac{\beta\mu}{\gamma}N(t)\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2 + 2\,(\hat{\mathbf{a}} - \hat{\mathbf{v}})^{\mathrm{T}}\,\nabla_{\hat{\mathbf{a}}}\dot{Q}(\mathbf{x}, \hat{\mathbf{a}}, t),$$

$$\leq -\rho(Q) - \frac{\beta}{\gamma}\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2. \tag{6.6}$$

By radial unboundedness of $Q(\mathbf{x}, t)$ in $\mathbf{x}$, equations 6.5 and 6.6 show that $\mathbf{x}$ remains bounded. Similarly, radial unboundedness of V in $\tilde{\mathbf{v}}$ and $\hat{\mathbf{a}} - \hat{\mathbf{v}}$ show that $\hat{\mathbf{v}}$ and $\hat{\mathbf{a}}$ remain bounded. Integrating equation 6.6 shows that $\frac{\beta}{\gamma}\int_0^\infty \|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2 dt \leq V(0) - V(\infty) < \infty$, so that $(\hat{\mathbf{a}} - \hat{\mathbf{v}}) \in \mathcal{L}_2$. An identical argument shows that $\int_0^\infty \rho(Q)dt < \infty$. Now, because $\mathbf{x}$ and $\hat{\mathbf{a}}$ are bounded and because $\tilde{f}(\mathbf{x}, \hat{\mathbf{a}}, \mathbf{a}, t)$ is locally bounded in $\mathbf{x}$ and $\hat{\mathbf{a}}$ uniformly in t by assumption, writing $\mathbf{x}(t) - \mathbf{x}(s) = \int_s^t (\mathbf{f}(\mathbf{x}(t'), \mathbf{a}, t') + \mathbf{u}(\hat{\mathbf{a}}(t'), t'))\,dt'$ shows that $\mathbf{x}(t)$ is uniformly continuous in t. Because $Q(\mathbf{x}, t)$ is uniformly continuous in t when $\mathbf{x}$ is bounded, because Q is bounded, and because ρ is continuous in Q, we conclude ρ is uniformly continuous in t and $\lim_{t\to\infty}\rho(t) = \lim_{t\to\infty}\rho(Q(\mathbf{x}(t), t)) = 0$ by Barbalat's lemma (lemma 1). This shows that $\lim_{t\to\infty} Q(\mathbf{x}(t), t) = 0$. □

By taking $Q = \frac{1}{2}s^2$ in proposition 6, we immediately recover proposition 5.

We now consider the integral functional,

$$\mathcal{L}\left(\hat{\mathbf{a}}, \dot{\hat{\mathbf{a}}}, t\right) = e^{\int_0^t \beta\mathcal{N}(t)dt}\frac{1}{\beta\mathcal{N}(t)}\left(\frac{1}{2}\dot{\hat{\mathbf{a}}}^{\mathrm{T}}\dot{\hat{\mathbf{a}}} - \gamma\beta\mathcal{N}(t)\frac{d}{dt}\int_0^t R(\mathbf{x}(t'), \hat{\mathbf{a}}(t'), t')dt'\right),$$

which generates the higher-order law

$$\ddot{\hat{\mathbf{a}}} + \dot{\hat{\mathbf{a}}}\left(\beta\mathcal{N} - \frac{\dot{\mathcal{N}}}{\mathcal{N}}\right) = -\gamma\beta\mathcal{N}\nabla_{\hat{\mathbf{a}}}R(\mathbf{x}, \hat{\mathbf{a}}, t). \tag{6.7}$$

We again rewrite equation 6.7 as two first-order systems

$$\dot{\hat{\mathbf{v}}} = -\gamma\nabla_{\hat{\mathbf{a}}}R(\mathbf{x}, \hat{\mathbf{a}}, t), \tag{6.8}$$

$$\dot{\hat{\mathbf{a}}} = \beta\mathcal{N}\,(\hat{\mathbf{v}} - \hat{\mathbf{a}})\,, \tag{6.9}$$

and now require a modified version of assumption 9.

Assumption 10. *$R(\mathbf{x}, \hat{\mathbf{a}}, t) \geq 0$ for all $\mathbf{x}$, $\hat{\mathbf{a}}$, and t, and is uniformly continuous in t for bounded $\mathbf{x}$ and $\hat{\mathbf{a}}$. $\nabla_{\hat{\mathbf{a}}}R(\mathbf{x}, \hat{\mathbf{a}}, t)$ is locally bounded in $\mathbf{x}$ and $\hat{\mathbf{a}}$ uniformly in t. Furthermore, there exists a time-dependent signal $N(t)$ and nonnegative scalar values $\beta \geq 0$, $\mu \geq 0$ such that*

$$R(\mathbf{x}, \mathbf{a}, t) - R(\mathbf{x}, \hat{\mathbf{a}}, t) - \frac{\beta\mu}{\gamma}N(t)\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2 + 2\,(\hat{\mathbf{a}} - \hat{\mathbf{v}})^{\mathrm{T}}\,\nabla_{\hat{\mathbf{a}}}R(\mathbf{x}, \hat{\mathbf{a}}, t)$$
$$\leq -kR(\mathbf{x}, \hat{\mathbf{a}}, t)$$

for some constant $k > 0$.

Similar to assumption 9, assumption 10 is a formal requirement that we may "complete the square." Consider the case when $R(\mathbf{x}, \hat{\mathbf{a}}, t) = \frac{1}{2}\tilde{f}(\mathbf{x}, \hat{\mathbf{a}}, \mathbf{a}, t)^2$. Then $R(\mathbf{x}, \mathbf{a}, t) = 0$, $\nabla_{\hat{\mathbf{a}}} R(\mathbf{x}, \hat{\mathbf{a}}, t) = \tilde{f}(\mathbf{x}, \hat{\mathbf{a}}, \mathbf{a}, t)\nabla_{\hat{\mathbf{a}}}\hat{f}(\mathbf{x}, \hat{\mathbf{a}}, t)$, and we may choose $N(t) = \|\nabla_{\hat{\mathbf{a}}}\hat{f}(\mathbf{x}, \hat{\mathbf{a}}, t)\|^2$.

With assumption 10, we can state the following proposition.

Proposition 7. *Consider algorithm 6.7 along with assumptions 6 and 10. Let T_x denote the maximal interval of existence of $\mathbf{x}(t)$. Then $\hat{\mathbf{v}}$ and $\hat{\mathbf{a}}$ remain bounded for $t \in [0, T_x]$, $(\hat{\mathbf{a}} - \hat{\mathbf{v}}) \in \mathcal{L}_2$ over this interval, and $\int_0^{T_x} R(\mathbf{x}(t'), \hat{\mathbf{a}}(t'), t')dt' < \infty$. Furthermore, for any bounded solution $\mathbf{x}$, these conclusions hold for all t and $R(\mathbf{x}(t), \hat{\mathbf{a}}(t), t) \to 0$.*

Proof. Consider the Lyapunov-like function,

$$V = \frac{1}{2\gamma}\tilde{\mathbf{v}}^{\mathrm{T}}\tilde{\mathbf{v}} + \frac{1}{2\gamma}(\hat{\mathbf{a}} - \hat{\mathbf{v}})^{\mathrm{T}}(\hat{\mathbf{a}} - \hat{\mathbf{v}}). \tag{6.10}$$

Equation 6.10 implies that with $\mathcal{N}(t) = 1 + \mu N(t)$,

$$\begin{aligned}
\dot{V} &= -\tilde{\mathbf{a}}^{\mathrm{T}}\nabla_{\hat{\mathbf{a}}} R(\mathbf{x}, \hat{\mathbf{a}}, t) - \frac{\beta}{\gamma}\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2 - \frac{\beta\mu}{\gamma}N(t)\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2 \\
&\quad + 2(\hat{\mathbf{a}} - \hat{\mathbf{v}})^{\mathrm{T}}\nabla_{\hat{\mathbf{a}}} R(\mathbf{x}, \hat{\mathbf{a}}, t), \\
&\leq R(\mathbf{x}, \mathbf{a}, t) - R(\mathbf{x}, \hat{\mathbf{a}}, t) - \frac{\beta}{\gamma}\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2 - \frac{\beta\mu}{\gamma}N(t)\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2 \\
&\quad + 2(\hat{\mathbf{a}} - \hat{\mathbf{v}})^{\mathrm{T}}\nabla_{\hat{\mathbf{a}}} R(\mathbf{x}, \hat{\mathbf{a}}, t), \\
&\leq -kR(\mathbf{x}, \hat{\mathbf{a}}, t) - \frac{\beta}{\gamma}\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2.
\end{aligned} \tag{6.11}$$

Equations 6.10 and 6.11 show boundedness of $\hat{\mathbf{v}}$ and $\hat{\mathbf{a}}$ over $[0, T_x]$. Furthermore, integrating equation 6.11 shows that $\int_0^{T_x}\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2 dt' < \infty$ and $\int_0^{T_x} R(\mathbf{x}(t'), \hat{\mathbf{a}}(t'), t')dt' < \infty$. For any bounded solution $\mathbf{x}$, these integrals may be extended to infinity, and we conclude that $(\hat{\mathbf{a}} - \hat{\mathbf{v}}) \in \mathcal{L}_2$, $\hat{\mathbf{a}} \in \mathcal{L}_\infty$, and $\hat{\mathbf{v}} \in \mathcal{L}_\infty$. Writing $\mathbf{x}(t) - \mathbf{x}(s)$ in integral form as in the proof of proposition 6 shows that $\mathbf{x}(t)$ is uniformly continuous in t, and in light of the local boundedness assumption on $\nabla_{\hat{\mathbf{a}}} R$, the same procedure can be applied to $\hat{\mathbf{v}}$ and $\hat{\mathbf{a}}$. Because $R(\mathbf{x}(t), \hat{\mathbf{a}}(t), t)$ is uniformly continuous in t for bounded $\mathbf{x}$ and $\hat{\mathbf{a}}$, and because $\mathbf{x}(t)$ and $\hat{\mathbf{a}}(t)$ are both uniformly continuous in t, we conclude that $R(\mathbf{x}(t), \hat{\mathbf{a}}(t), t)$ is uniformly continuous in t and $R \to 0$ by Barbalat's lemma (lemma 1). □

As mentioned in section 5.1, we will be particularly interested in proposition 7 when $R = \frac{1}{2}\tilde{f}^2$, which will generate composite adaptation algorithms and algorithms applicable to nonlinearly parameterized systems. Proposition 7 then shows that $\tilde{f} \in \mathcal{L}_2$ over the interval of existence of $\mathbf{x}(t)$. As shown

by lemma 18, with our error model this is enough to show that $\mathbf{x}(t)$ always remains bounded and hence $\tilde{f} \to 0$.

Remark 19. Classically, Lyapunov-like functions used in adaptive control consist of a sum of tracking and parameter estimation error terms, with $\dot{\hat{\mathbf{a}}}$ chosen to cancel a term of unknown sign. Several Lyapunov functions in this work consist only of parameter estimation error terms, such as equation 6.10. From a mathematical point of view, all that matters is that $\dot{V}$ is negative semidefinite and contains signals related to the tracking error. Integrating $\dot{V}$ allows the application of tools from functional analysis to ensure that the control goal is accomplished. The lack of tracking error term in V is the origin of the additional complication that $\mathbf{x}(t)$ must be shown to be bounded even after it is known that $\dot{V} \leq 0$.

6.2 First- and Second-Order Composite Adaptation Laws. Here we consider the linearly parameterized setting $f(\mathbf{x}, \mathbf{a}, t) = \mathbf{Y}(\mathbf{x}, t)\mathbf{a}$, and derive new first- and second-order composite adaptation laws. Composite adaptation laws are driven by two sources of error: the tracking error itself, as summarized by s in the Slotine and Li controller, and a prediction error. The prediction error term is generally obtained from an algebraic relation constructed by filtering the dynamics (Slotine & Li, 1991). We present a composite algorithm that does not require any explicit filtering of the dynamics but is instead driven simultaneously by s and $\tilde{f}$.

A starting point for our first proposed algorithm is to consider a hybrid local and integral velocity gradient functional,

$$Q(\mathbf{x}, t) = \frac{\gamma}{2} s(\mathbf{x}, t)^2 + \frac{\kappa}{2} \int_0^t \tilde{f}^2(\mathbf{x}(t'), \hat{\mathbf{a}}(t'), \mathbf{a}, t') dt', \tag{6.12}$$

where $\kappa > 0$ and $\gamma > 0$ are positive learning rates weighting the contributions of each term. As discussed in section 5.1, the first term leads to the Slotine and Li controller. The second can be clearly seen to satisfy assumptions 7 and 8 with $\mu(t) = 0$. It also satisfies assumption 6, as $\tilde{f}^2$ is a quadratic function of $\tilde{\mathbf{a}}$ for linear $\tilde{f}$. Following the velocity gradient formalism, the resulting adaptation law is given by

$$\dot{\hat{\mathbf{a}}} = -\mathbf{P}\mathbf{Y}^{\mathrm{T}} (\gamma s + \kappa \mathbf{Y}\tilde{\mathbf{a}}), \tag{6.13}$$

which is a composite adaptation law simultaneously driven by s and the instantaneous function approximation error $\mathbf{Y}\tilde{\mathbf{a}} = \tilde{f}$. Equation 6.13 depends on the function approximation error $\tilde{f}$, which is not measured and hence cannot be used directly in an adaptation law. Nevertheless, it can be obtained through a proportional-integral form for $\hat{\mathbf{a}}$ in an identical manner to

section 2.2. To do so, we define

$$\boldsymbol{\xi}(\mathbf{x},t) = -\kappa \mathbf{P} s(\mathbf{x},t)\mathbf{Y}(\mathbf{x},t)^{\mathrm{T}}, \tag{6.14}$$

$$\boldsymbol{\rho}(\mathbf{x},t) = \kappa \mathbf{P} \int_{x_n(t_0)}^{x_n(t)} s(\mathbf{x},t) \frac{\partial \mathbf{Y}(\mathbf{x},t)^{\mathrm{T}}}{\partial x_n} dx_n, \tag{6.15}$$

$$\hat{\mathbf{a}} = \bar{\mathbf{a}} + \boldsymbol{\xi}(\mathbf{x},t) + \boldsymbol{\rho}(\mathbf{x},t), \tag{6.16}$$

$$\begin{aligned}\dot{\bar{\mathbf{a}}} = &-(\kappa\eta+\gamma)s\mathbf{P}\mathbf{Y}^{\mathrm{T}} + \kappa s \sum_{i=1}^{n-1} \mathbf{P}\frac{\partial \mathbf{Y}}{\partial x_i}\dot{x}_i - \sum_{i=1}^{n-1}\left(\frac{\partial \boldsymbol{\rho}}{\partial x_i}\right)^{\mathrm{T}}\dot{x}_i, \\ &-\left(\frac{\partial \boldsymbol{\rho}}{\partial \mathbf{x}_d}\right)^{\mathrm{T}}\dot{\mathbf{x}}_d - \frac{\partial \boldsymbol{\xi}}{\partial t} - \frac{\partial \boldsymbol{\rho}}{\partial t}.\end{aligned} \tag{6.17}$$

Computing $\dot{\hat{\mathbf{a}}}$ demonstrates that equation 6.13 is obtained through only the known signals contained in equations 6.14 to 6.17 despite its dependence on $\mathbf{Y}\tilde{\mathbf{a}}$. A few remarks concerning algorithms 6.13 to 6.17 are in order.

Remark 20. The $\mathbf{Y}\tilde{\mathbf{a}}$ term may also be obtained by following the I&I formalism (Astolfi & Ortega, 2003; Liu et al., 2010). To our knowledge, this discussion is the first that demonstrates the possibility of using a PI law in combination with a standard Lyapunov-stability motivated adaptation law to obtain a composite law.

Remark 21. More error signals may be used for additional terms in the adaptation law. For example, a prediction error obtained by filtering the dynamics may also be employed, leading to a three-term composite algorithm.

Remark 22. Much like the standard composite law obtained by filtering the dynamics, rearranging equation 6.13 shows that $\dot{\tilde{\mathbf{a}}} + \mathbf{P}\mathbf{Y}^{\mathrm{T}}\mathbf{Y}\tilde{\mathbf{a}} = -\mathbf{P}\mathbf{Y}^{\mathrm{T}}s$, so that the additional term can be seen to add a damping term that smooths adaptation (Slotine & Li, 1991).

Remark 23. As mentioned in section 2.1, for clarity of presentation we have restricted our discussion to the nth-order system 2.1. In general, the PI form 6.16 leads to undesired unknown terms contained in $\left(\frac{\partial \boldsymbol{\xi}(\mathbf{x},\mathbf{x}_d)}{\partial \mathbf{x}}\right)^{\mathrm{T}}\dot{\mathbf{x}}$ in addition to the desired unknown term. In this case, the desired unknown term is $-\kappa\mathbf{P}\mathbf{Y}^{\mathrm{T}}\mathbf{Y}\tilde{\mathbf{a}}$, while the undesired unknown term is $-\kappa\mathbf{P}\frac{\partial \mathbf{Y}}{\partial x_n}\dot{x}_n s$. Indeed, the purpose of introducing the additional proportional term $\boldsymbol{\rho}(\mathbf{x},\mathbf{x}_d)$ in equation 6.14 is to cancel this undesired unknown term. In general, cancellation of the undesired terms can be obtained by choosing $\boldsymbol{\rho}$ to solve a PDE, and solutions to this PDE will only exist if the undesired term is the gradient of an auxiliary function. $\boldsymbol{\rho}$ is then chosen to be exactly this auxiliary function. In some cases, the PDE can be avoided, such as through dynamic scaling

techniques (Karagiannis, Sassano, & Astolfi, 2009) or the similar embedding technique of Tyukin (2011).

The properties of the adaptive law, equation 6.13, may be summarized with the following proposition.

Proposition 8. *Consider the adaptation algorithm 6.13 with a linearly parameterized unknown,* $f(\mathbf{x}, \mathbf{a}, t) = \mathbf{Y}(\mathbf{x}, t)\mathbf{a}$. *Then all trajectories* $(\mathbf{x}, \hat{\mathbf{a}})$ *remain bounded,* $s \in \mathcal{L}_2 \cap L_\infty$, $\tilde{f} \in \mathcal{L}_2$, $s \to 0$, *and* $\mathbf{x} \to \mathbf{x}_d$.

The proof is given in section A.1.

Following the velocity gradient with momentum approach of section 6.1, we now obtain a higher-order composite algorithm and give a PI implementation. We again consider a hybrid local and integral velocity gradient functional, so that equation 5.4 takes the form

$$\mathcal{L}\left(\hat{\mathbf{a}}, \dot{\hat{\mathbf{a}}}, t\right) = e^{\int_0^t \beta\mathcal{N}(t)dt} \frac{1}{\beta\mathcal{N}(t)} \left(\frac{1}{2} \dot{\hat{\mathbf{a}}}^{\mathrm{T}} \dot{\hat{\mathbf{a}}} - \beta\mathcal{N}(t) \frac{d}{dt} \times \left[\frac{\gamma}{2} s^2 + \frac{\kappa}{2} \int_0^t \tilde{f}^2(\mathbf{x}(t'), \hat{\mathbf{a}}(t'), \mathbf{a}, t') dt' \right] \right) \tag{6.18}$$

where $\gamma > 0$ and $\kappa > 0$ are positive constants weighting the two error terms. The Euler-Lagrange equations then lead to the higher-order composite system:

$$\ddot{\hat{\mathbf{a}}} + \left(\beta\mathcal{N} - \frac{\dot{\mathcal{N}}}{\mathcal{N}} \right) \dot{\hat{\mathbf{a}}} = -\beta\mathcal{N}\mathbf{Y}^{\mathrm{T}} \left(\gamma s + \kappa \mathbf{Y}\tilde{\mathbf{a}} \right). \tag{6.19}$$

As in section 5.2, equation 6.19 may be implemented as two first-order systems:

$$\dot{\hat{\mathbf{v}}} = -\mathbf{Y}^{\mathrm{T}} \left(\gamma s + \kappa \mathbf{Y}\tilde{\mathbf{a}} \right), \tag{6.20}$$

$$\dot{\hat{\mathbf{a}}} = \beta\mathcal{N} \left(\hat{\mathbf{v}} - \hat{\mathbf{a}} \right). \tag{6.21}$$

In an implementation, equation 6.20 is obtained through the PI form $\hat{\mathbf{v}} = \overline{\mathbf{v}} + \boldsymbol{\xi}(\mathbf{x}, t) + \boldsymbol{\rho}(\mathbf{x}, t)$ with $\boldsymbol{\xi}$, $\boldsymbol{\rho}$, and $\dot{\overline{\mathbf{v}}}$ given by equations 6.14, 6.15, and 6.17, respectively, with $\mathbf{P} = \mathbf{I}$. The properties of the higher-order composite adaptation law, equation 6.19, are stated in the following proposition.

Proposition 9. *Consider the higher-order composite adaptation algorithm, equation 6.19, for a linearly parameterized unknown,* $f(\mathbf{x}, \mathbf{a}, t) = \mathbf{Y}(\mathbf{x}, t)\mathbf{a}$. *Set* $\mathcal{N} = 1 + \mu\|\mathbf{Y}\|^2$ *and* $\mu > \frac{\gamma}{\beta}\left(\frac{1}{\eta} + \frac{\kappa}{\gamma}\right)$. *Then all trajectories* $(\mathbf{x}, \hat{\mathbf{v}}, \hat{\mathbf{a}})$ *remain bounded,* $\|\hat{\mathbf{v}} - \hat{\mathbf{a}}\| \in \mathcal{L}_2$, $s \in \mathcal{L}_\infty \cap L_2$, $\tilde{f} \in \mathcal{L}_\infty \cap L_2$, $s \to 0$, *and* $\mathbf{x} \to \mathbf{x}_d$.

The proof is given in section A.2.

Remark 24. By following the proof, the signal $\mathcal{N}$ may be chosen alternatively to be matrix-valued as $\mathbf{N} = \mathbf{I} + \mu \mathbf{Y}^{\mathrm{T}}\mathbf{Y}$.

Remark 25. The $\mathbf{Y}\tilde{\mathbf{a}}$ term may be used in isolation, by considering the Lyapunov function $V = \frac{1}{2}\|\tilde{\mathbf{v}}\|^2 + \frac{1}{2}\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2$.

Remark 26. A gain matrix $\mathbf{P} = \mathbf{P}^{\mathrm{T}} > 0$ of appropriate dimension may be placed in front of $\mathbf{Y}^{\mathrm{T}}$ in $\dot{\hat{\mathbf{v}}}$. The quadratic parameter estimation error terms in the Lyapunov function should then be replaced by the weighted terms $\frac{1}{2}\tilde{\mathbf{v}}^{\mathrm{T}}\mathbf{P}^{-1}\tilde{\mathbf{v}} + \frac{1}{2}\left(\hat{\mathbf{a}} - \hat{\mathbf{v}}\right)^{\mathrm{T}}\mathbf{P}^{-1}\left(\hat{\mathbf{a}} - \hat{\mathbf{v}}\right)$, and bounds on μ will be given in terms of $\|\mathbf{P}\|$.

6.3 A Momentum Algorithm for Nonlinearly Parameterized Adaptive Control. We now use the development in section 6.2 to present a new momentum algorithm applicable when the unknown parameters appear nonlinearly in the dynamics. We begin with an analogy to statistics.

Generalized linear model (GLM) regression is an extension of linear regression where the data are assumed to be generated by a function of the form $f(\mathbf{x}) = u(\mathbf{w}^{\mathrm{T}}\mathbf{x})$ for a known "link function" u and unknown parameters $\mathbf{w}$. The first computationally and statistically efficient algorithm for this problem, the GLM-Tron of Kakade et al. (2011), assumes that u is Lipschitz and monotonic, much like assumption 2.

The GLM-Tron algorithm was recently extended to the setting of kernel methods and was subsequently used to provably learn two hidden-layer neural networks by Goel and Klivans (2017); this extension is known as the Alphatron. In the kernel GLM setting handled by the Alphatron, the function to be approximated is assumed to be of the form $f(\mathbf{x}) = u\left(\sum_{i=1}^{m} w_i \mathcal{K}\left(\mathbf{x}, \mathbf{x}_i\right)\right)$, where $\mathcal{K}$ is the kernel function for a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$. $\mathcal{K}$ is thus given by the RKHS inner product of a feature map $\boldsymbol{\phi}$, $\mathcal{K}(\mathbf{x}, \mathbf{y}) = \langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{y}) \rangle_{\mathcal{H}}$.

The Alphatron initializes all weights to zero and, given a batch of labeled training data $(\mathbf{x}_i, f(\mathbf{x}_i))_{i=1}^m$, updates them with a learning rate $\lambda > 0$ according to the iteration

$$\hat{w}_i^{t+1} = \hat{w}_i^t - \frac{\lambda}{m}\left(\hat{f}(\hat{\mathbf{w}}^t, \mathbf{x}_i) - f(\mathbf{x}_i)\right). \tag{6.22}$$

We now demonstrate an equivalence between Tyukin's adaptation law equation 2.12, and the Alphatron weight update, equation 6.22 in the following proposition.

Proposition 10. *The adaptation law, equation 2.12, is an application of the Alphatron algorithm, equation 6.22, to adaptive control.*

The proof is given in section A.3.

Proposition 10 shows a convergence of techniques in nonlinearly parameterized adaptive control and nonconvex learning. This correspondence suggests the momentum-like variant of equation 2.12,

$$\ddot{\hat{\mathbf{a}}} + \left(\beta\mathcal{N} - \frac{\dot{\mathcal{N}}}{\mathcal{N}}\right)\dot{\hat{\mathbf{a}}} = -\gamma\beta\mathcal{N}\tilde{f}(\mathbf{x}, \hat{\mathbf{a}}, \mathbf{a}, t)\boldsymbol{\alpha}(\mathbf{x}, t), \tag{6.23}$$

which, as before, admits an equivalent representation in terms of two first-order systems,

$$\dot{\hat{\mathbf{v}}} = -\gamma\,\tilde{f}(\mathbf{x}, \hat{\mathbf{a}}, \mathbf{a}, t)\boldsymbol{\alpha}(\mathbf{x}, t), \tag{6.24}$$

$$\dot{\hat{\mathbf{a}}} = \beta\mathcal{N}\,(\hat{\mathbf{v}} - \hat{\mathbf{a}})\,. \tag{6.25}$$

Equation 6.23 may be implemented through equations 6.24 and 6.25 via the PI form, equations 2.13 to 2.16, applied to the $\hat{\mathbf{v}}$ variable.

Equation 6.23 may be obtained via the Bregman Lagrangian, equation 6.18, for velocity gradient laws with momentum by choosing only the integral term. It is then necessary to modify the resulting Euler-Lagrange equations by setting f_m' to ± 1 based on the monotonicity of $\tilde{f}$ as described in section 2.2. The following proposition summarizes the properties of equations 6.24 and 6.25.

Proposition 11. *Consider the algorithm 6.23 or its equivalent form, 6.24 and 6.25 under assumption 2 with* $\mathcal{N} = 1 + \mu\|\boldsymbol{\alpha}(\mathbf{x}, t)\|^2$ *and* $\mu > \frac{\gamma D_1}{\beta}$. *Then all trajectories* $(\mathbf{x}, \hat{\mathbf{a}}, \hat{\mathbf{v}})$ *remain bounded,* $\tilde{f} \in L_2$, $(\hat{\mathbf{a}} - \hat{\mathbf{v}}) \in \mathcal{L}_2$, $s \in \mathcal{L}_2 \cap \mathcal{L}_\infty$, $s \to 0$ *and* $\mathbf{x} \to \mathbf{x}_d$.

The proof is given in section A.4.

Remark 27. As noted in remark 24, by following the proof of proposition 11, one may also take $\mathcal{N}$ to be matrix-valued as $\mathbf{N} = 1 + \mu\boldsymbol{\alpha}(\mathbf{x}, t)\boldsymbol{\alpha}(\mathbf{x}, t)^{\mathrm{T}}$.

Remark 28. As in remark 26, a gain matrix $\mathbf{P} = \mathbf{P}^{\mathrm{T}} > 0$ of appropriate dimension may be placed in front of $\boldsymbol{\alpha}(\mathbf{x}, t)$ in $\dot{\hat{\mathbf{v}}}$.

Predominantly inspired by deep learning, there has recently been strong interest in nonconvex models that are nevertheless amenable to gradient-based or gradient-inspired optimization. The development in this section suggests that machine learning models that can be provably optimized using gradient techniques represent a promising class of nonlinear parameterizations for adaptive control development.

6.4 The Elastic Modification. We now consider a modification to the previously discussed adaptive control laws inspired by the elastic averaging SGD (EASGD) algorithm (Boffi & Slotine, 2020; Zhang et al., 2014). EASGD is an algorithm intended for distributed training of deep neural networks across p graphics processing units (GPUs). Each GPU is used to train a local copy of the deep network model, and each local copy maintains

its own set of parameters $\hat{\mathbf{a}}^{(i)}$. These parameters are updated according to the iteration

$$\hat{\mathbf{a}}_{t+1}^{(i)} = \hat{\mathbf{a}}_t^{(i)} - \lambda \mathbf{g}_t^{(i)} + \lambda k \left(\overline{\mathbf{a}}_t - \hat{\mathbf{a}}_t^{(i)} \right), \tag{6.26}$$

$$\overline{\mathbf{a}}_{t+1} = \overline{\mathbf{a}}_t + \lambda k \left(\frac{1}{p} \sum_{i=1}^{p} \hat{\mathbf{a}}^{(i)} - \overline{\mathbf{a}}_t \right), \tag{6.27}$$

where λ is the learning rate, $\mathbf{g}_t^{(i)}$ is the stochastic gradient approximation computed by the ith agent at time step t, k is the coupling strength, and $\overline{\mathbf{a}}$ is the center variable. Equation 6.27 takes the form of a low-pass filter of the instantaneous average of the set of local parameters.

Boffi and Slotine (2020) observed that in the nondistributed ($p = 1$) case, equations 6.26 and 6.27 do not reduce to standard stochastic gradient descent, and that application of EASGD in this setting has different generalization properties from those of standard SGD when used to train deep neural networks. In a similar spirit, by construction of suitable Lyapunov functions, we now show that adding a center-like variable to the adaptive laws considered in previous sections maintains their stability. This immediately gives rise to a new class of higher-order adaptive control algorithms. Interestingly, these algorithms do not seem to admit an equivalent representation in terms of a single second-, third-, or fourth-order system for $\hat{\mathbf{a}}$, but must be written as a system of first-order equations.

Remark 29. The algorithms considered in this subsection immediately extend to the case of cloud-based adaptation for networked robotic systems (Wensing & Slotine, 2018), where the center variable is allowed to have its own dynamics as in equation 6.27 rather than simply representing the instantaneous spatial average of the distributed parameters.

We first apply the elastic modification to the Slotine and Li adaptive controller, equation 2.8, for linearly parameterized unknown dynamics $\tilde{f} = \mathbf{Y}\tilde{\mathbf{a}}$. These results extend trivially to the nonfiltered composite algorithm of section 6.2. To this end, we define the adaptation law

$$\dot{\hat{\mathbf{a}}} = -\mathbf{P}\mathbf{Y}^{\mathrm{T}} s + k\,(\overline{\mathbf{a}} - \hat{\mathbf{a}})\,, \tag{6.28}$$

$$\dot{\overline{\mathbf{a}}} = k\,(\hat{\mathbf{a}} - \overline{\mathbf{a}})\,, \tag{6.29}$$

whose basic stability properties are summarized in the following proposition.

Proposition 12. *Consider the adaptation law, equations 6.28 and 6.29. Then all trajectories* $(\mathbf{x}, \hat{\mathbf{a}}, \overline{\mathbf{a}})$ *remain bounded,* $s \in \mathcal{L}_2 \cap \mathcal{L}_\infty$, $(\hat{\mathbf{a}} - \overline{\mathbf{a}}) \in \mathcal{L}_2$, $s \to 0$ *and* $\mathbf{x} \to \mathbf{x}_d$.

The proof is given in section A.5.

We now apply the elastic modification to algorithm 2.12 for nonlinearly parameterized unknown dynamics satisfying assumption 2. As in equations 6.28 and 6.29, we define

$$\dot{\hat{\mathbf{a}}} = -\tilde{f}\mathbf{P}\boldsymbol{\alpha} + k\,(\overline{\mathbf{a}} - \hat{\mathbf{a}})\,, \tag{6.30}$$

$$\dot{\overline{\mathbf{a}}} = k\,(\hat{\mathbf{a}} - \overline{\mathbf{a}})\,. \tag{6.31}$$

Proposition 13. *Consider the adaptation law, equations 6.30 and 6.31. Then all trajectories* $(\mathbf{x}, \hat{\mathbf{a}}, \overline{\mathbf{a}})$ *remain bounded,* $\tilde{f} \in \mathcal{L}_2 \cap \mathcal{L}_\infty$, $(\hat{\mathbf{a}} - \overline{\mathbf{a}}) \in \mathcal{L}_2$, $s \in \mathcal{L}_\infty \cap \mathcal{L}_2$, $s \to 0$ *and* $\mathbf{x} \to \mathbf{x}_d$.

The proof is given in section A.6.

We now consider the higher-order algorithms presented in sections 6.2 and 6.3. In the higher-order setting, there are three clear possibilities for the elastic modification: coupling to a center variable for the $\hat{\mathbf{a}}$ variable, coupling to a center variable for the $\hat{\mathbf{v}}$ variable, or coupling to center variables in both $\hat{\mathbf{a}}$ and $\hat{\mathbf{v}}$. We prove stability for all three possibilities only in the nonlinearly parameterized setting described by assumption 2. The results extend naturally to the higher-order composite algorithm for linearly parameterized systems presented in section 6.2. We begin with the first possibility,

$$\dot{\hat{\mathbf{v}}} = -\gamma\,\tilde{f}\boldsymbol{\alpha}, \tag{6.32}$$

$$\dot{\hat{\mathbf{a}}} = \beta\mathcal{N}\,(\hat{\mathbf{v}} - \hat{\mathbf{a}}) + k\beta\mathcal{N}\,(\overline{\mathbf{a}} - \hat{\mathbf{a}})\,, \tag{6.33}$$

$$\dot{\overline{\mathbf{a}}} = k\beta\mathcal{N}\,(\hat{\mathbf{a}} - \overline{\mathbf{a}})\,. \tag{6.34}$$

The basic stability properties of the algorithm, equations 6.32 to 6.34, are summarized in the following proposition.

Proposition 14. *Consider the algorithm, equations 6.32 to 6.34, under assumption 2. Set* $\frac{1}{3} \le k < 1$, $\mathcal{N} = 1 + \mu\|\boldsymbol{\alpha}(\mathbf{x}, t)\|^2$, *and* $\mu > \frac{2D_1\gamma}{\beta(1-k)}$. *Then all trajectories* $(\mathbf{x}, \hat{\mathbf{a}}, \hat{\mathbf{v}}, \overline{\mathbf{a}})$ *remain bounded,* $\tilde{f} \in \mathcal{L}_2 \cap \mathcal{L}_\infty$, $s \in \mathcal{L}_2 \cap \mathcal{L}_\infty$, $(\hat{\mathbf{a}} - \hat{\mathbf{v}}) \in \mathcal{L}_2$, $(\hat{\mathbf{a}} - \overline{\mathbf{a}}) \in \mathcal{L}_2$, $s \to 0$ *and* $\mathbf{x} \to \mathbf{x}_d$.

The proof is given in section A.7. We now consider the second possibility of adding a center variable in the $\hat{\mathbf{v}}$ variable,

$$\dot{\hat{\mathbf{v}}} = -\gamma\,\tilde{f}\boldsymbol{\alpha} + \rho\,(\overline{\mathbf{v}} - \hat{\mathbf{v}})\,, \tag{6.35}$$

$$\dot{\overline{\mathbf{v}}} = \rho\,(\hat{\mathbf{v}} - \overline{\mathbf{v}})\,, \tag{6.36}$$

$$\dot{\hat{\mathbf{a}}} = \beta\mathcal{N}\,(\hat{\mathbf{v}} - \hat{\mathbf{a}})\,. \tag{6.37}$$

The basic stability properties of equations 6.35 to 6.37 are summarized in the following proposition.

Proposition 15. *Consider the algorithm, equations 6.35 to 6.37, under assumption 2. Set* $\rho < 2\beta$, $\mathcal{N} = 1 + \mu\|\boldsymbol{\alpha}(\mathbf{x}, t)\|^2$, *and* $\mu > \frac{\gamma D_1}{\beta}$. *Then all trajectories*

$(\mathbf{x}, \hat{\mathbf{a}}, \hat{\mathbf{v}}, \overline{\mathbf{v}})$ *remain bounded,* $\tilde{f} \in \mathcal{L}_2 \cap \mathcal{L}_\infty$, $s \in \mathcal{L}_2 \cap \mathcal{L}_\infty$, $(\hat{\mathbf{v}} - \overline{\mathbf{v}}) \in \mathcal{L}_2$, $(\hat{\mathbf{v}} - \hat{\mathbf{a}}) \in \mathcal{L}_2$, $s \to 0$ *and* $\mathbf{x} \to \mathbf{x}_d$.

The proof is given in section A.8. Finally, we consider adding coupling to center variables in both $\hat{\mathbf{a}}$ and $\hat{\mathbf{v}}$,

$$\dot{\hat{\mathbf{v}}} = -\gamma \tilde{f}\boldsymbol{\alpha} + \rho\,(\overline{\mathbf{v}} - \hat{\mathbf{v}})\,, \tag{6.38}$$

$$\dot{\overline{\mathbf{v}}} = \rho\,(\hat{\mathbf{v}} - \overline{\mathbf{v}})\,, \tag{6.39}$$

$$\dot{\hat{\mathbf{a}}} = \beta\mathcal{N}\,(\hat{\mathbf{v}} - \hat{\mathbf{a}}) + k\beta\mathcal{N}\,(\overline{\mathbf{a}} - \hat{\mathbf{a}})\,, \tag{6.40}$$

$$\dot{\overline{\mathbf{a}}} = k\beta\mathcal{N}\,(\hat{\mathbf{a}} - \overline{\mathbf{a}})\,. \tag{6.41}$$

The basic stability properties of equations 6.38 to 6.41 are summarized in the following proposition.

Proposition 16. *Consider the algorithm, equations 6.38 to 6.41, under assumption 2. Set* $\rho < \beta(1-k)$, $\frac{1}{3} \leq k < 1$, $\mathcal{N} = 1 + \mu\|\boldsymbol{\alpha}(\mathbf{x}, t)\|^2$, *and* $\mu > \frac{2\gamma D_1}{\beta(1-k)}$. *Then all trajectories* $(\mathbf{x}, \hat{\mathbf{v}}, \overline{\mathbf{v}}, \hat{\mathbf{a}}, \overline{\mathbf{a}})$ *remain bounded,* $\tilde{f} \in \mathcal{L}_2 \cap \mathcal{L}_\infty$, $s \in \mathcal{L}_2 \cap \mathcal{L}_\infty$, $(\hat{\mathbf{v}} - \overline{\mathbf{v}}) \in \mathcal{L}_2$, $(\hat{\mathbf{a}} - \overline{\mathbf{a}}) \in \mathcal{L}_2$, $(\hat{\mathbf{v}} - \hat{\mathbf{a}}) \in \mathcal{L}_2$, $s \to 0$, *and* $\mathbf{x} \to \mathbf{x}_d$.

The proof is given in section A.9.

We have thus shown that all Euclidean adaptive control algorithms presented in this article,[10] as well as the classic algorithm of Slotine and Li, can be modified to include feedback coupling to a low-pass filtered version of the adaptation variables. It is well known that iterate averaging for stochastic optimization algorithms such as stochastic gradient descent can improve convergence rates via variance reduction (Polyak & Juditsky, 1992). The elastic modification is similar in spirit but employs feedback rather than series coupling. This suggests that adding the elastic term may improve robustness of adaptation algorithms, and we leave a theoretical investigation of this conjecture for future work.

6.5 Exponential Forgetting Least Squares and Bounded Gain Forgetting. We now demonstrate how to apply the techniques of exponential forgetting and bounded gain forgetting least squares (Slotine & Li, 1991) to the adaptation algorithms we have developed. These techniques are useful for estimation of time-varying parameters, as they rapidly discard previous information used for parameter estimation. Exponential forgetting least squares is described by a time-dependent learning rate matrix $\mathbf{P}(t)$, which, in the linearly parameterized case $\tilde{f} = \mathbf{Y}\tilde{\mathbf{a}}$, takes the form

[10] Similar results apply for the natural algorithms with additional technical details by replacing quadratic terms in the Lyapunov functions with Bregman divergences.

$$\dot{\mathbf{P}} = \begin{cases} \lambda \mathbf{P} - \mathbf{P}\mathbf{Y}^{\mathrm{T}}\mathbf{Y}\mathbf{P} & \text{if } \|\mathbf{P}\| \leq P_0 \\ 0 & \text{else} \end{cases} \tag{6.42}$$

where $\lambda > 0$ is a constant forgetting factor, P_0 is a maximum bound on the norm, and $\|\mathbf{P}\|$ is a matrix norm such as the opeator norm. Equation 6.42 implies for the inverse matrix,

$$\frac{d}{dt}\mathbf{P}^{-1} = \begin{cases} -\lambda \mathbf{P}^{-1} + \mathbf{Y}^{\mathrm{T}}\mathbf{Y} & \text{if } \|\mathbf{P}\| \leq P_0 \\ 0 & \text{else} \end{cases} \tag{6.43}$$

In the nonlinearly parameterized case described by assumption 2, we will replace $\mathbf{Y}^{\mathrm{T}}$ in equations 6.42 and 6.43 by $\boldsymbol{\alpha}(\mathbf{x}, t)$. In the bounded gain forgetting technique, λ is a time-dependent function,

$$\lambda(t) = \lambda_0 \left(1 - \frac{\|\mathbf{P}\|}{P_0}\right), \tag{6.44}$$

where $\lambda_0 > 0$ sets the forgetting factor when the norm of $\mathbf{P}$ is small. It can be shown that this choice of $\lambda(t)$ ensures that $\|\mathbf{P}\| \leq P_0$, and thus we may drop the case statement in equations 6.42 and 6.43 (Slotine & Li, 1991). The choice of $\lambda(t)$ in bounded gain forgetting and the case statement used in equations 6.42 and 6.43 are both employed to prevent unboundedness of the learning rate matrix.

We focus on algorithms without the elastic modification of section 6.4; extension to the elastic modification is simple. We also focus on the bounded gain forgetting technique: proofs for the exponential forgetting least squares technique are identical, with the addition of an appropriate case statement in the time derivative of the Lyapunov function. For simplicity, we include only the time-dependent gain $\mathbf{P}(t)$ and set the scalar gains $\kappa = \gamma = 1$ where applicable.

We begin with the first-order non-filtered composite, equation 6.13, with $\mathbf{P}$ given by equation 6.42. In this case, the composite algorithm may be implemented via the PI form equations 6.14 to 6.17, where now $\mathbf{P} = \mathbf{P}(t)$.

Proposition 17. *Consider the adaptation algorithm, equation 6.13, with* $\mathbf{P}(t)$ *given by equation 6.42,* $\lambda(t)$ *given by equation 6.44, and* $\kappa = \gamma = 1$. *Then all trajectories* $(\mathbf{x}, \hat{\mathbf{a}})$ *remain bounded,* $\tilde{f} \in \mathcal{L}_2 \cap \mathcal{L}_\infty$, $s \in \mathcal{L}_2 \cap \mathcal{L}_\infty$, $s \to 0$, *and* $\mathbf{x} \to \mathbf{x}_d$.

The proof is given in section A.10.

We can state a similar result for the higher-order nonfiltered composite with time-dependent $\mathbf{P}(t)$ given by equation 6.42,

$$\ddot{\hat{\mathbf{a}}} + \left(\beta \mathcal{N} - \frac{\dot{\mathcal{N}}}{\mathcal{N}} - \dot{\mathbf{P}}\mathbf{P}^{-1}\right)\dot{\hat{\mathbf{a}}} = -\beta \mathcal{N}\mathbf{P}(t)\mathbf{Y}^{\mathrm{T}}\left(s + \tilde{f}\right), \tag{6.45}$$

which admits a representation as two first-order equations,

$$\dot{\hat{\mathbf{v}}} = -\mathbf{P}(t)\mathbf{Y}^{\mathrm{T}}\left(s + \tilde{f}\right), \tag{6.46}$$

$$\dot{\hat{\mathbf{a}}} = \beta\mathcal{N}\mathbf{P}(t)\left(\hat{\mathbf{v}} - \hat{\mathbf{a}}\right). \tag{6.47}$$

Equation 6.46 can be implemented via the PI form $\hat{\mathbf{v}} = \overline{\mathbf{v}} + \boldsymbol{\xi}(\mathbf{x}, t) + \boldsymbol{\rho}(\mathbf{x}, t)$ where $\boldsymbol{\xi}$, $\boldsymbol{\rho}$, and $\dot{\overline{\mathbf{v}}}$ are given by equations 6.14, 6.15, and 6.17, respectively, with $\gamma = \kappa = 1$.

Proposition 18. *Consider the adaptation algorithm, equation 6.45, with* $\mathbf{P}(t)$ *given by equation 6.42,* $\lambda(t)$ *given by equation 6.44,* $\mathcal{N}(t) = 1 + \mu\|\mathbf{Y}\|^2$ *and* $\mu > \frac{3\eta+2}{2\eta\beta}$. *Then all trajectories* $(\mathbf{x}, \hat{\mathbf{v}}, \hat{\mathbf{a}})$ *remain bounded,* $\tilde{f} \in \mathcal{L}_2 \cap \mathcal{L}_\infty$, $s \in \mathcal{L}_2 \cap \mathcal{L}_\infty$, $s \to 0$ *and* $\mathbf{x} \to \mathbf{x}_d$.

The proof is given in section A.11.

Remark 30. Because $\mathbf{P}(t)$ is uniformly bounded in t, it is not necessary to include $\mathbf{P}(t)$ in equation 6.47; by a slight modification of the proof, it is easy to show that the modified higher-order law,

$$\ddot{\hat{\mathbf{a}}} + \left(\beta\mathcal{N} - \frac{\dot{\mathcal{N}}}{\mathcal{N}}\right)\dot{\hat{\mathbf{a}}} = -\beta\mathcal{N}\mathbf{P}(t)\mathbf{Y}^{\mathrm{T}}\left(s + \tilde{f}\right),$$

is also a stable adaptive law with a suitable choice of gains.

We now consider Tyukin's first-order algorithm for nonlinearly parameterized systems, equation 2.12, with $\mathbf{P} = \mathbf{P}(t)$ given by equation 6.42. To do so, we require an additional assumption:

Assumption 11. *For the same function* $\boldsymbol{\alpha}(\mathbf{x}, t)$ *as in assumption 2, there exists a constant* D_2 *such that*

$$|\tilde{f}(\mathbf{x}, \hat{\mathbf{a}}, \mathbf{a}, t)| \geq D_2|\boldsymbol{\alpha}(\mathbf{x}, t)^{\mathrm{T}}\tilde{\mathbf{a}}|. \tag{6.48}$$

Together with assumption 2, Assumption 6.48 states that $\tilde{f}$ lies between two linear functions. Given that the update, equation 6.42, is derived based on recursive linear least squares considerations, it is unsurprising that such an assumption is required in the nonlinearly parameterized setting. We are now in a position to state the following proposition.

Proposition 19. *Consider the adaptation algorithm equation 2.12 with* $\mathbf{P}(t)$ *given by equation 6.42 and* $\lambda(t)$ *given by equation 6.44 for* $\tilde{f}$ *satisfying assumptions 2 and 11. Further assume that* $D_1 < 2D_2^2$ *or that* $D_2 > \frac{1}{2}$. *Then, all trajectories* $(\mathbf{x}, \hat{\mathbf{a}})$ *remain bounded,* $\tilde{f} \in \mathcal{L}_2$, $s \in \mathcal{L}_2 \cap \mathcal{L}_\infty$, $s \to 0$ *and* $\mathbf{x} \to \mathbf{x}_d$.

The proof is given in section A.12.

Last, we consider the momentum algorithm for nonlinearly parameterized systems,

$$\ddot{\hat{\mathbf{a}}} + \left(\beta\mathcal{N} - \frac{\dot{\mathcal{N}}}{\mathcal{N}} - \dot{\mathbf{P}}\mathbf{P}^{-1}\right)\dot{\hat{\mathbf{a}}} = -\beta\mathcal{N}\tilde{f}\mathbf{P}(t)\boldsymbol{\alpha}(\mathbf{x}, t), \tag{6.49}$$

which admits a representation as two first-order equations,

$$\dot{\hat{\mathbf{v}}} = -\tilde{f}\mathbf{P}(t)\boldsymbol{\alpha}(\mathbf{x}, t), \tag{6.50}$$

$$\dot{\hat{\mathbf{a}}} = \beta\mathcal{N}\mathbf{P}(t)\left(\hat{\mathbf{v}} - \hat{\mathbf{a}}\right). \tag{6.51}$$

Proposition 20. *Consider the adaptation algorithm, equation 6.49, with* $\mathbf{P}(t)$ *given by 6.42,* $\lambda(t)$ *given by (6.44),* $\mathcal{N} = 1 + \mu\|\boldsymbol{\alpha}\|^2$*, and* $\mu > \frac{4D_2 - 2 + (2D_1+1)^2}{\beta(4D_2-1)}$*. Suppose* $\tilde{f}$ *satisfies assumptions 2 and 11. Further assume that* $D_2 > \frac{1}{2}$*. Then all trajectories* $(\mathbf{x}, \hat{\mathbf{v}}, \hat{\mathbf{a}})$ *remain bounded,* $\tilde{f} \in \mathcal{L}_2$*,* $s \in \mathcal{L}_2 \cap \mathcal{L}_\infty$*,* $s \to 0$ *and* $\mathbf{x} \to \mathbf{x}_d$*.*

The proof is given in section A.13.

Remark 31. As in remark 30, because $\mathbf{P}(t)$ is uniformly bounded in t, it is not necessary to include $\mathbf{P}(t)$ in equation 6.51. It is simple to show by modification of the proof that

$$\ddot{\hat{\mathbf{a}}} + \left(\beta\mathcal{N} - \frac{\dot{\mathcal{N}}}{\mathcal{N}}\right)\dot{\hat{\mathbf{a}}} = -\beta\mathcal{N}\tilde{f}\mathbf{P}(t)\boldsymbol{\alpha}(\mathbf{x}, t)$$

is also a stable adaptive law with a suitable choice of gains.

7 Natural Momentum Algorithms

The Bregman Lagrangian allows for the introduction of non-Euclidean metrics. In section 5.2, we took the potential function ψ to be the Euclidean norm, $\psi(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|^2$. We now show that taking ψ to be an arbitrary strictly convex function leads to a more general class of algorithms that can be seen as the higher-order variants of those discussed in sections 2.3 and 3. With the same definitions of $\bar{\alpha}$, $\bar{\gamma}$, and $\bar{\beta}$ as in section 5.2, but now taking the general Bregman divergence $d_\psi(\cdot \,\|\, \cdot)$, the Bregman Lagrangian, equation 5.5 takes the form

$$\mathcal{L} = e^{\int_0^t \beta\mathcal{N}(t)dt}\left(\beta\mathcal{N}d_\psi\left(\hat{\mathbf{a}} + \frac{\dot{\hat{\mathbf{a}}}}{\beta\mathcal{N}} \,\middle\|\, \hat{\mathbf{a}}\right) - \gamma\frac{d}{dt}\left[\frac{1}{2}s^2\right]\right). \tag{7.1}$$

The Euler-Lagrange equations for equation 7.1 lead to the natural adaptation law with momentum

$$\ddot{\hat{\mathbf{a}}} + \left(\beta\mathcal{N} - \frac{\dot{\mathcal{N}}}{\mathcal{N}}\right)\dot{\hat{\mathbf{a}}} + \gamma\beta\mathcal{N}\left(\nabla^2\psi\left(\hat{\mathbf{a}} + \frac{\dot{\hat{\mathbf{a}}}}{\beta\mathcal{N}}\right)\right)^{-1}\mathbf{Y}^{\mathrm{T}}s = 0. \tag{7.2}$$

Above, the Euclidean adaptive law has been modified so that $\mathbf{Y}^{\mathrm{T}}s$ is now premultiplied by the inverse Hessian of ψ evaluated at $\hat{\mathbf{a}} + \frac{\dot{\hat{\mathbf{a}}}}{\beta\mathcal{N}}$. As discussed in section 5.2, this quantity is precisely $\hat{\mathbf{v}}$. The resulting adaptation law can thus be written in the equivalent form:

$$\dot{\hat{\mathbf{v}}} = -\gamma\left(\nabla^2\psi\left(\hat{\mathbf{v}}\right)\right)^{-1}\mathbf{Y}^{\mathrm{T}}s, \tag{7.3}$$

$$\dot{\hat{\mathbf{a}}} = \beta\mathcal{N}\left(\hat{\mathbf{v}} - \hat{\mathbf{a}}\right). \tag{7.4}$$

Equations 7.3 and 7.4 demonstrate that using the Bregman divergence in the Bregman Lagrangian leads to momentum variants of the natural algorithms of section 2.3. Taking the $\beta \to \infty$ limit immediately recovers the first-order laws discussed in section 2.3. The stability of the above laws for strongly convex ψ is stated in the following proposition.

Proposition 21. *Consider the higher-order "natural" adaptation law equation 7.2. Assume that ψ is l-strongly convex so that $\nabla^2\psi(\cdot) \geq l\ \mathbf{I}$ globally. Take $\mathcal{N} = 1 + \mu\|\mathbf{Y}\|^2$ and $\mu > \frac{\gamma\left(1+l^{-1}\right)^2}{4\beta\eta}$. Then all trajectories $(\mathbf{x}, \hat{\mathbf{v}}, \hat{\mathbf{a}})$ remain bounded, $s \in \mathcal{L}_2 \cap \mathcal{L}_\infty$, $s \to 0$, and $\mathbf{x} \to \mathbf{x}_d$.*

The proof is given in section A.14. A second, related variant is given by

$$\dot{\hat{\mathbf{v}}} = -\gamma\left(\nabla^2\psi(\hat{\mathbf{v}})\right)^{-1}\mathbf{Y}^{\mathrm{T}}s, \tag{7.5}$$

$$\dot{\hat{\mathbf{a}}} = \beta\mathcal{N}\left(\nabla^2\psi(\hat{\mathbf{a}})\right)^{-1}\left(\nabla\psi(\hat{\mathbf{v}}) - \nabla\psi(\hat{\mathbf{a}})\right). \tag{7.6}$$

Algorithm 7.5 and 7.6 is equivalent to algorithm 5.6 entirely in the mirrored domain. Indeed, it may be rewritten as

$$\frac{d}{dt}\nabla\psi(\hat{\mathbf{v}}) = -\gamma\mathbf{Y}^{\mathrm{T}}s,$$

$$\frac{d}{dt}\nabla\psi(\hat{\mathbf{a}}) = \beta\mathcal{N}\left(\nabla\psi(\hat{\mathbf{v}}) - \nabla\psi(\hat{\mathbf{a}})\right),$$

which shows that $\nabla\psi(\hat{\mathbf{a}})$ obtains the same values over time as $\hat{\mathbf{a}}$ computed via algorithm 5.6. The stability of this adaptive law (shown in proposition 22) implies that the parameters obtained by the momentum algorithm, equation 5.6, may be transformed via the inverse of the gradient of an

l-strongly convex and L-smooth function, and the resulting transformed parameters will still ensure stability and tracking for the closed-loop system.

A modification of equation 7.6 that is driven by $\hat{\mathbf{v}}$ rather than $\nabla\psi(\hat{\mathbf{v}})$ is given by

$$\dot{\hat{\mathbf{v}}} = -\gamma \left(\nabla^2\psi(\hat{\mathbf{v}})\right)^{-1} \mathbf{Y}^{\mathrm{T}} s, \tag{7.7}$$

$$\dot{\hat{\mathbf{a}}} = \beta\mathcal{N} \left(\nabla^2\psi(\hat{\mathbf{a}})\right)^{-1} (\hat{\mathbf{v}} - \hat{\mathbf{a}}). \tag{7.8}$$

The properties of these two possible adaptation laws are given in the following proposition.

Proposition 22. *Consider the adaptation algorithm, equations 7.5 and 7.6, or the adaptation algorithm, equations 7.7 and 7.8. Assume that ψ is l-strongly convex and L-smooth, so that $l\mathbf{I} \leq \nabla^2\psi(\cdot) \leq L\mathbf{I}$. Take $\mathcal{N} = 1 + \mu\|\mathbf{Y}\|^2$, and choose $\mu > \frac{\gamma(l+\gamma L)^2}{4\beta\eta l^3}$ in the former case and $\mu > \frac{\gamma(l+\gamma L)^2}{4\beta\eta l^2}$ in the latter. Then all trajectories $(\mathbf{x}, \hat{\mathbf{v}}, \hat{\mathbf{a}})$ remain bounded, $s \in \mathcal{L}_2 \cap \mathcal{L}_\infty$, $s \to 0$ and $\mathbf{x} \to \mathbf{x}_d$.*

The proof is presented in section A.15.

Remark 32. For efficient implementation of the proposed natural momentum algorithms, as well as for their first-order equivalents, ψ should be chosen so that $\left[\nabla^2\psi(\cdot)\right]^{-1}$ is efficiently computable and ideally sparse. Alternatively, if the inverse function of the gradient $\left(\nabla\psi^{-1}\right)(\cdot)$ is efficiently computable, $\nabla\psi(\hat{\mathbf{a}})$ or $\nabla\psi(\hat{\mathbf{v}})$ may be updated directly and subsequently inverted to arrive at the parameter values. Discretization of these algorithms is a subtle issue, and discretization of the $\dot{\hat{\mathbf{a}}}$ and $\dot{\hat{\mathbf{v}}}$ dynamics directly results in a natural gradient-like update (Amari, 1998), while discretization of the $\frac{d}{dt}\nabla\psi(\hat{\mathbf{a}})$ and $\frac{d}{dt}\nabla\psi(\hat{\mathbf{v}})$ dynamics leads to a mirror descent-like update (Beck & Teboulle, 2003; Nemirovski & Yudin, 1983); these discrete-time algorithms have the same continuous-time limit (Krichene, Bayen, & Bartlett, 2015).

The above natural adaptation laws with momentum may be generalized to composite algorithms, as well as to algorithms for nonlinearly parameterized adaptive control, by replacing Euclidean norms by Bregman divergences where appropriate in the proofs of the corresponding Euclidean algorithms (see, e.g., the proofs of propositions 21 and 22). Rather than derive this for each algorithm, we now show how the general results on velocity gradient algorithms with momentum (see propositions 6 and 7) can be extended to the non-Euclidean setting. We start with the case of a local functional, which requires the modification of assumption 9 to an equivalent non-Euclidean version.

Assumption 12. *There exists a time-dependent signal $N(t)$ and nonnegative scalar values $\beta \geq 0$, $\mu \geq 0$ such that the time derivative of the goal functional*

evaluated at the true parameters, $\dot{Q}(\mathbf{x}, \mathbf{a}, t)$, satisfies the following inequality:

$$\dot{Q}(\mathbf{x}, \mathbf{a}, t) - \frac{\beta\mu}{\gamma} N(t) \|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2 + (\hat{\mathbf{a}} - \hat{\mathbf{v}})^{\mathrm{T}} \left(\mathbf{I} + \left[\nabla^2 \psi(\hat{\mathbf{v}}) \right]^{-1} \right) \nabla_{\hat{\mathbf{a}}} \dot{Q}(\mathbf{x}, \hat{\mathbf{a}}, t) \leq -\rho(Q). \tag{7.9}$$

In equation 7.9, $\rho(\cdot)$ is positive definite, continuous in Q, and satisfies $\rho(0) = 0$.

With assumption 12 in hand, we can state the following non-Euclidean equivalent of proposition 6. We focus on the variant equation 7.2, as the other possibilities are similar.

Proposition 23. *Consider the algorithm*

$$\ddot{\hat{\mathbf{a}}} + \left(\beta\mathcal{N} - \frac{\dot{\mathcal{N}}}{\mathcal{N}} \right) \dot{\hat{\mathbf{a}}} + \gamma\beta\mathcal{N} \left[\nabla^2 \psi \left(\hat{\mathbf{a}} + \frac{\dot{\hat{\mathbf{a}}}}{\beta\mathcal{N}} \right) \right]^{-1} \nabla_{\hat{\mathbf{a}}} \dot{Q}(\mathbf{x}, \hat{\mathbf{a}}, t) = 0$$

or its equivalent first-order form,

$$\begin{aligned} \dot{\hat{\mathbf{v}}} &= -\gamma \left[\nabla^2 \psi(\hat{\mathbf{v}}) \right]^{-1} \nabla_{\hat{\mathbf{a}}} \dot{Q}(\mathbf{x}, \hat{\mathbf{a}}, t), \\ \dot{\hat{\mathbf{a}}} &= \beta\mathcal{N} (\hat{\mathbf{v}} - \hat{\mathbf{a}}), \end{aligned}$$

and assume Q satisfies assumptions 4, 6, and 12. Then all solutions $(\mathbf{x}(t), \hat{\mathbf{v}}(t), \hat{\mathbf{a}}(t))$ remain bounded, $(\hat{\mathbf{a}} - \hat{\mathbf{v}}) \in \mathcal{L}_2$, and $\lim_{t\to\infty} Q(\mathbf{x}(t); t) = 0$.

Proof. Consider the Lyapunov-like function,

$$V = Q(\mathbf{x}, t) + \frac{1}{\gamma} d_\psi (\mathbf{a} \,\|\, \hat{\mathbf{v}}) + \frac{1}{2\gamma} (\hat{\mathbf{a}} - \hat{\mathbf{v}})^{\mathrm{T}} (\hat{\mathbf{a}} - \hat{\mathbf{v}}). \tag{7.10}$$

Equation 7.10 implies that, with $\mathcal{N}(t) = 1 + \mu N(t)$,

$$\begin{aligned} \dot{V} &= \dot{Q}(\mathbf{x}, \hat{\mathbf{a}}, t) - \tilde{\mathbf{a}}^{\mathrm{T}} \nabla_{\hat{\mathbf{a}}} \dot{Q}(\mathbf{x}, \hat{\mathbf{a}}, t) - \frac{\beta}{\gamma} \|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2 - \frac{\beta\mu}{\gamma} N(t) \|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2 \\ &\quad + (\hat{\mathbf{a}} - \hat{\mathbf{v}})^{\mathrm{T}} \left(\mathbf{I} + \left[\nabla^2 \psi(\hat{\mathbf{v}}) \right]^{-1} \right) \nabla_{\hat{\mathbf{a}}} \dot{Q}(\mathbf{x}, \hat{\mathbf{a}}, t), \\ &\leq \dot{Q}(\mathbf{x}, \mathbf{a}, t) - \frac{\beta}{\gamma} \|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2 - \frac{\beta\mu}{\gamma} N(t) \|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2 \\ &\quad + (\hat{\mathbf{a}} - \hat{\mathbf{v}})^{\mathrm{T}} \left(\mathbf{I} + \left[\nabla^2 \psi(\hat{\mathbf{v}}) \right]^{-1} \right) \nabla_{\hat{\mathbf{a}}} \dot{Q}(\mathbf{x}, \hat{\mathbf{a}}, t), \\ &\leq -\rho(Q) - \frac{\beta}{\gamma} \|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2. \end{aligned} \tag{7.11}$$

The first line to the second follows by convexity of $\dot{Q}(\mathbf{x}, \hat{\mathbf{a}}, t)$ in its second argument, while the second line to the third follows by assumption 12. The remainder of the proof is identical to Proposition 6. □

For the integral variant, we require a non-Euclidean version of assumption 10.

Assumption 13. *$R(\mathbf{x}, \hat{\mathbf{a}}, t) \geq 0$ for all $\mathbf{x}$, $\hat{\mathbf{a}}$, and t, and is uniformly continuous in t for bounded $\mathbf{x}$ and $\hat{\mathbf{a}}$. $\nabla_{\hat{\mathbf{a}}} R(\mathbf{x}, \hat{\mathbf{a}}, t)$ is locally bounded in $\mathbf{x}$ and $\hat{\mathbf{a}}$ uniformly in t. Furthermore, there exists a time-dependent signal $N(t)$ and nonnegative scalar values $\beta \geq 0$, $\mu \geq 0$ such that*

$$R(\mathbf{x}, \mathbf{a}, t) - R(\mathbf{x}, \hat{\mathbf{a}}, t) - \frac{\beta\mu}{\gamma} N(t) \|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2 + (\hat{\mathbf{a}} - \hat{\mathbf{v}})^{\mathrm{T}} \left(\mathbf{I} + \left[\nabla^2 \psi \left(\hat{\mathbf{v}} \right) \right]^{-1} \right) \nabla_{\hat{\mathbf{a}}} R(\mathbf{x}, \hat{\mathbf{a}}, t) \leq -kR(\mathbf{x}, \hat{\mathbf{a}}, t)$$

for some constant $k > 0$.

With assumption 13, we can state the following proposition.

Proposition 24. *Consider the algorithm*

$$\ddot{\hat{\mathbf{a}}} + \left(\beta\mathcal{N} - \frac{\dot{\mathcal{N}}}{\mathcal{N}} \right) \dot{\hat{\mathbf{a}}} + \gamma\beta\mathcal{N} \left[\nabla^2 \psi \left(\hat{\mathbf{a}} + \frac{\dot{\hat{\mathbf{a}}}}{\beta\mathcal{N}} \right) \right]^{-1} \nabla_{\hat{\mathbf{a}}} R(\mathbf{x}, \hat{\mathbf{a}}, t) = 0,$$

or its equivalent first-order form,

$$\dot{\hat{\mathbf{v}}} = -\gamma \left[\nabla^2 \psi(\hat{\mathbf{v}}) \right]^{-1} \nabla_{\hat{\mathbf{a}}} R(\mathbf{x}, \hat{\mathbf{a}}, t),$$
$$\dot{\hat{\mathbf{a}}} = \beta\mathcal{N} \left(\hat{\mathbf{v}} - \hat{\mathbf{a}} \right),$$

along with assumptions 6 and 13. Let T_x denote the maximal interval of existence of $\mathbf{x}(t)$. Then $\hat{\mathbf{v}}$ and $\hat{\mathbf{a}}$ remain bounded for $t \in [0, T_x)$, $(\hat{\mathbf{a}} - \hat{\mathbf{v}}) \in \mathcal{L}_2$ over this interval, and $\int_0^{T_x} R(\mathbf{x}(t'), \hat{\mathbf{a}}(t'), t')dt' < \infty$. Furthermore, for any bounded solution $\mathbf{x}$, these conclusions hold for all t and $R(\mathbf{x}(t), \hat{\mathbf{a}}(t), t) \to 0$.

Proof. Consider the Lyapunov-like function,

$$V = \frac{1}{\gamma} d_\psi (\mathbf{a} \,\|\, \hat{\mathbf{v}}) + \frac{1}{2\gamma} (\hat{\mathbf{a}} - \hat{\mathbf{v}})^{\mathrm{T}} (\hat{\mathbf{a}} - \hat{\mathbf{v}}). \tag{7.12}$$

Equation 7.12 implies that, with $\mathcal{N}(t) = 1 + \mu N(t)$,

$$\dot{V} = -\tilde{\mathbf{a}}^{\mathrm{T}} \nabla_{\hat{\mathbf{a}}} R(\mathbf{x}, \hat{\mathbf{a}}, t) - \frac{\beta}{\gamma} \|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2 - \frac{\beta\mu}{\gamma} N(t) \|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2 + (\hat{\mathbf{a}} - \hat{\mathbf{v}})^{\mathrm{T}} \left(\mathbf{I} + \left[\nabla^2 \psi(\hat{\mathbf{v}}) \right]^{-1} \right) \nabla_{\hat{\mathbf{a}}} R(\mathbf{x}, \hat{\mathbf{a}}, t),$$

$$\leq R(\mathbf{x}, \mathbf{a}, t) - R(\mathbf{x}, \hat{\mathbf{a}}, t) - \frac{\beta}{\gamma}\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2 - \frac{\beta\mu}{\gamma}N(t)\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2$$
$$+ (\hat{\mathbf{a}} - \hat{\mathbf{v}})^{\mathrm{T}}\left(\mathbf{I} + \left[\nabla^2\psi(\hat{\mathbf{v}})\right]^{-1}\right)\nabla_{\hat{\mathbf{a}}}R(\mathbf{x}, \hat{\mathbf{a}}, t),$$
$$\leq -kR(\mathbf{x}, \hat{\mathbf{a}}, t) - \frac{\beta}{\gamma}\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2. \tag{7.13}$$

The first line to the second follows by convexity of $R(\mathbf{x}, \hat{\mathbf{a}}, t)$ in its second argument, while the second to the third follows by assumption 13. The remainder of the proof is identical to proposition 7. □

The general methodology captured by the proofs of propositions 23 and 24, in combination with the results of section 6.3, may be exploited to derive non-Euclidean variants of our nonfiltered composite algorithm and our momentum algorithm for nonlinearly parameterized adaptive control. Note that the strong convexity and smoothness requirements of propositions 21 and 22, in combination with a suitable choice of $N(t)$, are one way to satisfy the requirements of assumptions 12 and 13.

Remark 33. Our implicit regularization results in section 3 also extend to the higher-order setting captured by algorithm 7.2. The assumption that $\hat{\mathbf{a}} \to \hat{\mathbf{a}}_\infty$ implies $\dot{\hat{\mathbf{a}}} \to 0$. As noted in section 5.2, $\hat{\mathbf{v}} = \hat{\mathbf{a}} + \frac{\dot{\hat{\mathbf{a}}}}{\beta N}$, and we thus conclude that under this assumption $\hat{\mathbf{v}} \to \hat{\mathbf{a}}_\infty$. Because $\dot{\hat{\mathbf{v}}}$ in equation 7.3 is identical to algorithm 3.2, the result follows. A formal statement of this fact is provided in section B.1.

8 Simulations

In this section, we perform several numerical experiments demonstrating the validity of our theory and consider a number of applications of our non-Euclidean adaptive laws.

8.1 Convergence and Implicit Regularization of a Momentum Algorithm for Nonlinearly Parameterized Systems. We first empirically verify the global convergence and implicit regularization of our momentum algorithm for nonlinearly parameterized systems, equation 6.23. In particular, we consider a second-order system,

$$\dot{x}_1 = x_2,$$
$$\dot{x}_2 = u - f(\mathbf{x}, \mathbf{a}, t),$$

with an unknown system dynamics of the form

$$f(\mathbf{x}, \mathbf{a}, t) = \sigma\left(\tanh\left(\mathbf{V}\mathbf{x}\right)^{\mathrm{T}}\mathbf{a}\right). \tag{8.1}$$

Equation 8.1 represents a three-layer neural network with input layer $\mathbf{x}$, hidden-layer weights $\mathbf{V}$, hidden-layer nonlinearity $\tanh(\cdot)$, hidden-layer weights $\mathbf{a}$, and output nonlinearity $\sigma(x) = e^{\cdot 1x}$. The system model, equation 8.1, can clearly be seen to satisfy assumption 2 with $\boldsymbol{\alpha}(\mathbf{x}) = \tanh(\mathbf{Vx})$.[11] The PI form of algorithm 6.23 is given by

$$\nabla\psi(\hat{\mathbf{v}}) = \overline{v} + \xi(\mathbf{x}, t) + \boldsymbol{\rho}(\mathbf{x}, t), \tag{8.2}$$

$$\xi(\mathbf{x}, t) = -\gamma s(\mathbf{x}, t)\tanh(\mathbf{Vx}), \tag{8.3}$$

$$\begin{aligned}\rho(\mathbf{x}, t) = \gamma\big[&\tanh(\mathbf{Vx})\, x_2 - \log(\cosh(\mathbf{Vx})) \oslash \mathbf{V}_2 \\ &+ (\lambda\tilde{x} - x_{2,d}(t))\tanh(\mathbf{Vx})\big],\end{aligned} \tag{8.4}$$

$$\begin{aligned}\dot{\overline{v}} = \gamma\,&(\dot{x}_{2,d}(t) - \lambda\,(x_2 - x_{2,d}(t)) - \eta s)\tanh(\mathbf{Vx}) \\ &+ \gamma x_2 \tanh(\mathbf{Vx}) \circ \mathbf{V}_1 \oslash \mathbf{V}_2,\end{aligned} \tag{8.5}$$

$$\dot{\hat{\mathbf{a}}} = \beta\left(1 + \mu\|\tanh(\mathbf{Vx})\|^2\right)(\hat{\mathbf{v}} - \hat{\mathbf{a}}), \tag{8.6}$$

where $\circ$ and $\oslash$ denote elementwise multiplication and division, respectively, where $\mathbf{V}_i$ is the ith column of $\mathbf{V}$ and $\hat{\mathbf{v}}$ is obtained from $\nabla\psi(\hat{\mathbf{v}})$ by inverting $\nabla\psi$. For the squared p norm $\psi(\cdot) = \frac{1}{2}\|\cdot\|_p^2$, the inverse function can be analytically computed as

$$\left(\nabla\psi^{-1}\right)(\mathbf{y}) = \|\mathbf{y}\|_q^{2-q}|\mathbf{y}|^{q-1}\text{sign}(\mathbf{y}), \tag{8.7}$$

where $\frac{1}{q} + \frac{1}{p} = 1$, $|\cdot|$ denotes elementwise absolute value and $\text{sign}(\cdot)$ denotes elementwise sign (Gentile, 2003). We consider the l_1, l_2, l_4, l_6, and l_{10} norms for ψ. To approximate the l_1 norm, equation 8.7 is used with $p = 1.1$. All other p norms can be used directly.

In all simulations we take $\lambda = .5$ in the definition of s (see equation 2.4) and $\eta = .5$ in the control input (see equation 2.5). For the adaptation hyperparameters, we choose $\gamma = 1.5$ for the l_2, l_4, and l_6 norms. We take $\gamma = 50$ for the l_1 norm and $\gamma = .5$ for the l_{10} norm.[12] In all cases, $\beta = 1$ and $\mu = \frac{3\gamma}{2\eta\beta}$. We set $\dim(\mathbf{a}) = \dim(\hat{\mathbf{a}}) = 500$ and randomly initialize $\hat{\mathbf{a}}$ and $\hat{\mathbf{v}}$ around zero from a normal distribution with standard deviation 10^{-3}. The true parameter vector $\mathbf{a}$ is drawn from a normal distribution with mean zero and standard deviation 7.5. The matrix $\mathbf{V}$ is set to have normally distributed

[11] While the exponential is not globally Lipschitz continuous, it is locally.

[12] These values of γ were chosen to ensure good control performance without excessively high control inputs or fast parameter adaptation. In particular, adaptation occurs very slowly with l_1 regularization, as small parameters are quickly eliminated to promote sparsity. A high adaptation gain was needed to ensure adaptation on a similar timescale to the other norms.

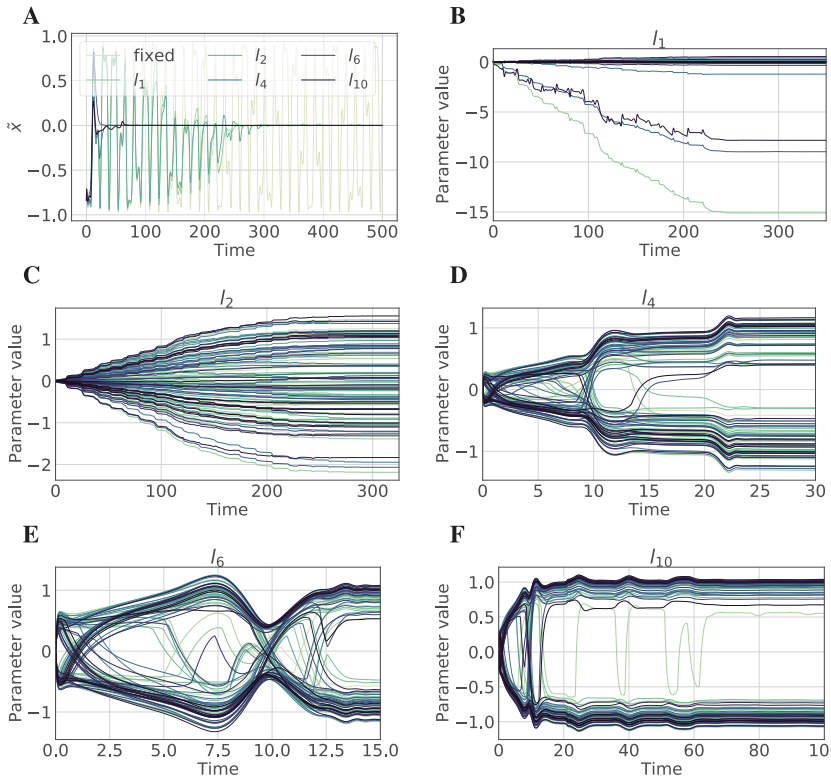

Figure 1: Tracking error and parameter trajectories. (A) Trajectory tracking error. All algorithms result in convergence $\mathbf{x} \to \mathbf{x}_d$, though transient performance differs between the algorithms. (B–F) Parameter trajectories for 100/500 of the total parameters. Each algorithm results in remarkably different parameter trajectories and final values $\hat{\mathbf{a}}^{\psi}_{\infty}$.

elements with standard deviation $\frac{1}{\sqrt{\dim \hat{\mathbf{a}}}}$. The state vector is initialized such that $\mathbf{x}(0) = \mathbf{x}_d(0)$. The desired trajectory is taken to be

$$x_d(t) = \sin\left(\frac{\sqrt{2}\pi}{12}t + \cos\left(\frac{\sqrt{3}\pi}{12}t\right)\right).$$

The tracking error for each choice of ψ along with a baseline comparison to fixed $\hat{\mathbf{a}}(t) = \hat{\mathbf{a}}(0)$ is shown in Figure 1A. Figures 1B–1F show trajectories for 100 out of the 500 parameters. The timescale on each axis is set to show the trajectories approximately until the parameters converge for the given algorithm. Each case results in remarkably different dynamics and resulting

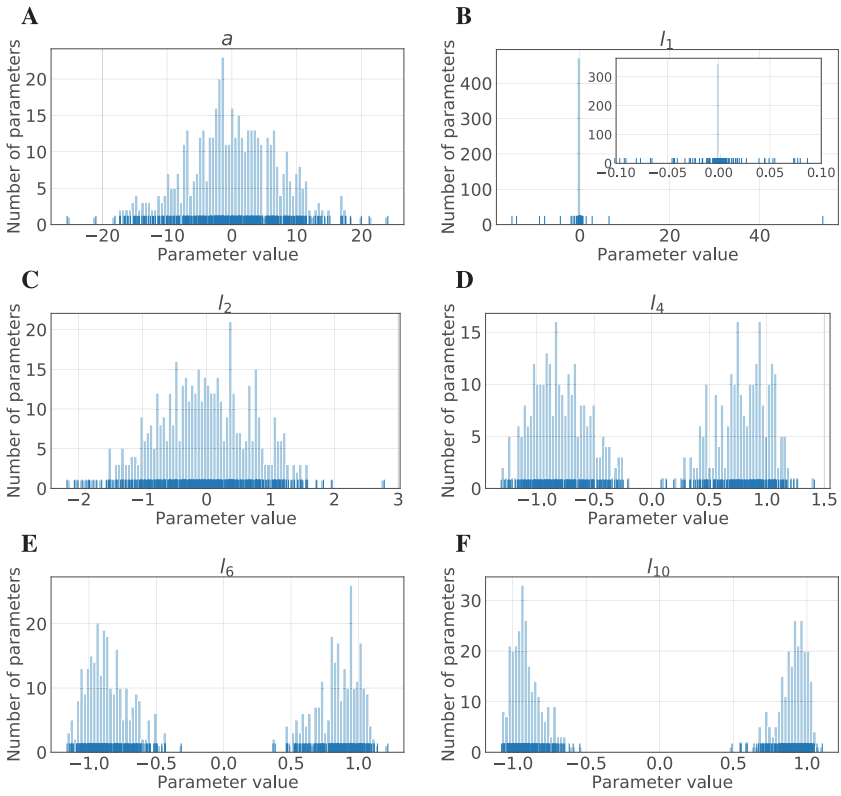

Figure 2: Parameter histograms. (A) True parameters $\mathbf{a}$. (B) Parameter vector found by the algorithm with $\psi(\cdot) = \frac{1}{2}\|\cdot\|_{1.1}^2$. The resulting solution is extremely sparse and has a few parameters with large magnitude, indicative of implicit l_1 regularization. (C) Parameter vector found by the standard Euclidean algorithm with $\psi(\cdot) = \frac{1}{2}\|\cdot\|_2^2$. The resulting parameter vector looks approximately gaussian distributed, indicating l_2 regularization. (C–F) Parameter vectors found by $\psi(\cdot) = \frac{1}{2}\|\cdot\|_p^2$ with $p = 4$, 6, and 10, respectively. The transition clearly indicates a trend toward l_∞-norm regularization, with two bimodal peaks forming around ± 1. The l_∞ norm of the parameter vector decreases with increasing p.

converged parameter vectors $\hat{\mathbf{a}}_\infty^\psi$. The tracking performance is good for each algorithm.

Further insight can be gained into the structure of the parameter vector $\hat{\mathbf{a}}_\infty^\psi$ found by each adaptation algorithm by consideration of the histograms (rug plots shown on x-axis) for $\hat{\mathbf{a}}$ at the end of the simulation in Figures 2A to 2F. Figure 2A shows the true parameter vector. The choice of $\psi(\cdot) = \frac{1}{2}\|\cdot\|_{1.1}^2$ in Figure 2B leads to a sparse solution with most of the weight placed on a

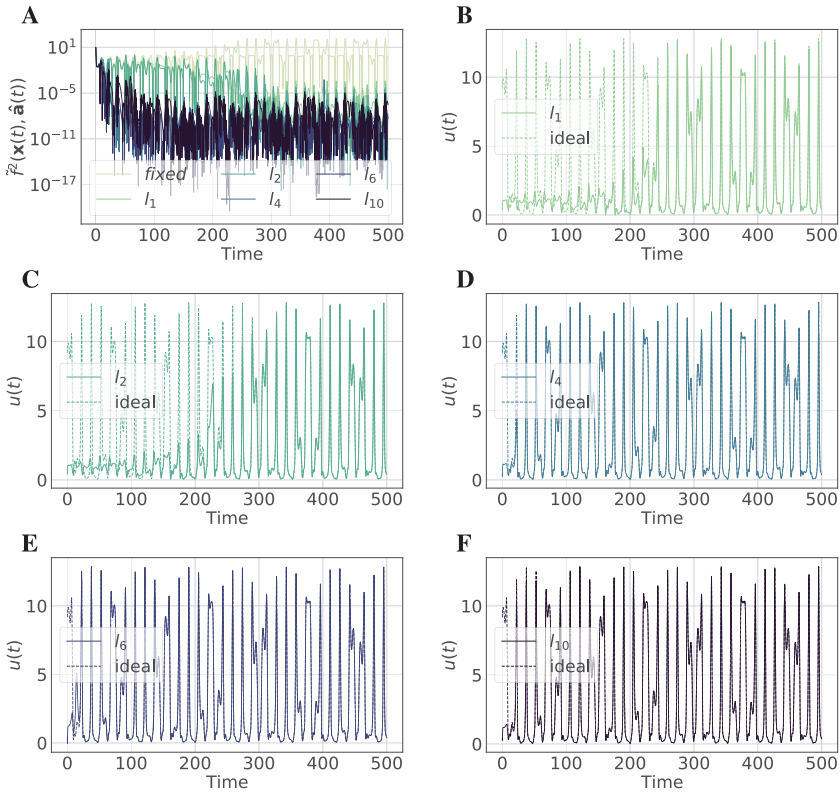

Figure 3: Function approximation error and control inputs. (A) The function approximation error $\tilde{f}^2(\mathbf{x}(t), \hat{\mathbf{a}}(t))$. All algorithms drive the error to zero. (B–F) Comparison of the control input $u^\psi(t)$ to the "ideal" control $u(t) = \ddot{x}_d(t) + f(\mathbf{x}_d(t), \mathbf{a})$. All algorithms converge to the ideal control, though at a different rate. The control magnitude is kept to a reasonable level in every case.

few parameters. This is consistent with l_1 regularized solutions found by the LASSO algorithm (Tibshirani, 1996). The inset displays a closer view around zero. The choice of $\psi(\cdot) = \frac{1}{2}\|\cdot\|_2^2$ in Figure 2C (Euclidean adaptation law) leads to a parameter vector $\hat{\mathbf{a}}_\infty^{\frac{1}{2}\|\cdot\|_2^2} \neq \mathbf{a}$ that is roughly gaussian distributed. This distribution highlights the implicit l_2 regularization of standard adaptation laws. The progression from $\psi(\cdot) = \frac{1}{2}\|\cdot\|_4^2$ to $\psi(\cdot) = \frac{1}{2}\|\cdot\|_{10}^2$ displays a trend toward approximate l_∞-norm regularization: the distribution of parameters is pushed to be bimodal and peaked around ± 1, and the l_∞ norm of $\hat{\mathbf{a}}_\infty$ decreases as p is increased.

Figure 3A shows the function approximation error $\tilde{f}^2(\mathbf{x}(t), \hat{\mathbf{a}}(t), \mathbf{a})$ for each algorithm along with a reference value for fixed $\hat{\mathbf{a}}(t) = \hat{\mathbf{a}}(0)$. Each

algorithm, as expected by our theory and seen by the low tracking error in Figure 1A, pushes $\tilde{f}^2$ to zero despite the different forms of regularization imposed on the parameter vectors. Figures 3B to 3F show the control input as a function of time along with the unique "ideal" control law $u(t) = \ddot{x}_d + f(\mathbf{x}_d(t), \mathbf{a}(t))$ valid when $\mathbf{x}(0) = \mathbf{x}_d(0)$. All control inputs can be seen to converge to the ideal law, though the rate of convergence depends on the choice of algorithm. The control input is of reasonable magnitude throughout adaptation for each algorithm.

8.2 Learning to Control with Primitives. We now demonstrate that the mirror descent-like laws of section 3 can be used to learn convex combinations of control primitives. Our approach is analogous to the use of multiplicative weight updates in machine learning and respects the natural l_1 geometry over the probability simplex.

As a model problem for this setting, we consider the second-order system

$$\dot{x}_1 = x_2,$$
$$\dot{x}_2 = u - \tanh(\mathbf{V}\mathbf{x})^{\mathrm{T}}\,\mathbf{a},$$

with $\mathbf{a} \in \mathbb{R}^p$ a fixed vector of unknown parameters and $\mathbf{V} \in \mathbb{R}^{p\times 2}$ a random matrix with $V_{ij} \sim \mathcal{N}\left(0, \frac{1}{p^2}\right)$. To define our control primitives, we consider a distribution over tasks specified by random desired trajectories

$$x_d^{(i)}(t) = M\sin(A_i t + B_i\cos(C_i t)) + D_i,$$

with $D_i = 2i(-1)^i \times M$, $A_i \sim \mathrm{Unif}(0, 5\pi)$, $B_i \sim \mathrm{Unif}(0, 3)$, and $C_i \sim \mathrm{Unif}(0, 5\pi)$. The shift D_i ensures that the desired trajectories occupy nonoverlapping regions of state space. We then learn primitives $\{u_i\}_{i=1}^N$ to track $\left\{x_d^{(i)}(t)\right\}_{i=1}^N$ where each u_i is given by equation 2.5 with parameter estimates $\hat{\mathbf{a}}^{(i)}$. The parameter estimates are found via the Slotine and Li adaptation law,

$$\dot{\hat{\mathbf{a}}}^{(i)} = -\gamma\tanh(\mathbf{V}\mathbf{x})^{\mathrm{T}} s,$$

which is allowed to run until the parameter estimates converge. We set $p = 15$, $N = 300$, $M = 0.1$, $\gamma = 5$, and $\eta = \lambda = 0.5$. Each vector of parameter estimates $\hat{\mathbf{a}}^{(i)}$ is initialized randomly, $\hat{\mathbf{a}}^{(i)}(0) \sim \mathcal{N}\left(0, \sigma_{\hat{\mathbf{a}}}^2\right)$ with $\sigma_{\hat{\mathbf{a}}} = 10^{-3}$. The state is initialized randomly for each task, $\mathbf{x}_0 \sim \mathcal{N}\left(0, \sigma_{\mathbf{x}}^2\right)$ with $\sigma_{\mathbf{x}} = 5$. The true parameters $\mathbf{a}$ are drawn randomly, $\mathbf{a} \sim \mathcal{N}\left(0, \sigma_{\mathbf{a}}^2\right)$ with $\sigma_{\mathbf{a}} = 2$.

With control primitives u_i capable of tracking trajectories $x_d^{(i)}$ in hand, we consider tracking a desired trajectory $x_d(t)$ given piecewise by the

previously drawn random trajectories. Concretely, we fix a time horizon T and a number of tasks k, and set

$$x_d(t) = x_d^{(i_l)}(t) \text{ if } t_{l-1} \leq t < t_l,$$

with $l = 1, \ldots, k$, i_l drawn uniformly from $i = 1, \ldots, N$, $t_0 = 0$, and $t_l = \frac{lT}{k}$. To leverage the learned control primitives, we use the input

$$u = \sum_{i=1}^{N} \hat{\beta}_i u_i = \mathbf{u}\hat{\boldsymbol{\beta}}.$$

Above, $\mathbf{u} \in \mathbb{R}^{1\times N}$ is a row vector with components u_i. We require that $\hat{\beta}_i > 0$ for all i and that $\sum_{i=1}^{N} \hat{\beta}_i = 1$. In our experiments, we fix $T = 1000$ and set $k = 5$.

It is well known in the online convex optimization community that mirror descent with respect to the entropy $\psi\left(\hat{\boldsymbol{\beta}}\right) = \sum_i \hat{\beta}_i \log \hat{\beta}_i$ can improve the dimension dependence of convergence rates in comparison to projected gradient descent when optimizing over the simplex (Hazan, 2016). Here we demonstrate that the same phenomenon appears in adaptive control. We consider two adaptation laws,

$$\dot{\hat{\boldsymbol{\beta}}} = -\gamma \mathbf{u}^T s,$$
$$\frac{d}{dt}\nabla\psi\left(\hat{\boldsymbol{\beta}}\right) = -\gamma \mathbf{u}^T s,$$

with projection of $\hat{\boldsymbol{\beta}}$ onto the simplex.[13] In both cases, we initialize $\hat{\beta}_i = \frac{1}{N}$ and set $\gamma = .25$.

Our results are shown in Figure 4. In Figure 4A, we show convergence of s for both adaptive laws. s jumps every 200 units of time as the task changes discretely. While both converge, adaptation with respect to the entropy converges significantly faster and minimizes s to lower values. This effect is more prominently displayed in Figure 4B, which shows the convergence of s on a logarithmic scale. Figures 4C and 4D show the parameter trajectories for the mirror descent and Euclidean laws, respectively. The mirror descent law displays trajectories in which fewer parameters stray from zero. Those that do stray an order of magnitude farther from zero than the Euclidean

[13] We use the `lsoda` integrator in `scipy.integrate.ode`. For the Euclidean adaptation law, we zero any components if they become negative and divide by the one norm of the parameter vector. For the non-Euclidean adaptation law, we integrate the mirror descent-like dynamics directly and update $\nabla\psi\left(\hat{\boldsymbol{\beta}}\right)$ via `lsoda`. After each time step, we compute $\hat{\boldsymbol{\beta}}$ by inverting the gradient of ψ, which ensures that each component is positive. We then project by dividing by the one norm of the parameter vector.

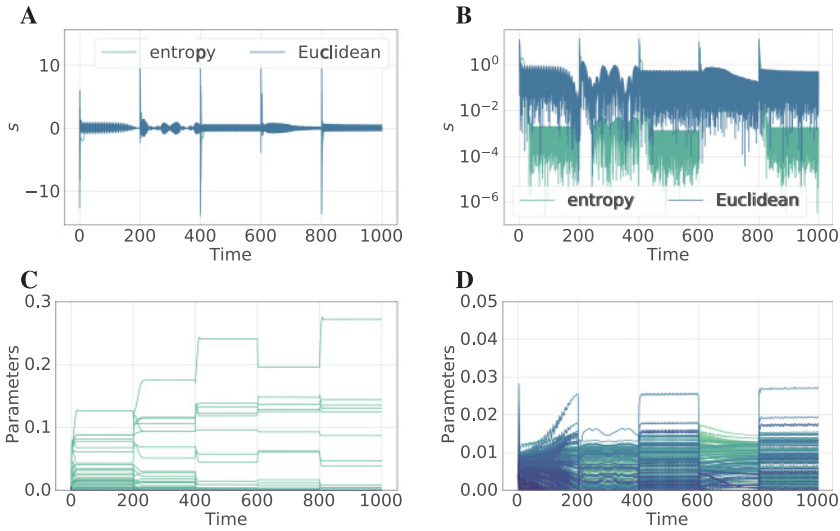

Figure 4: Learning to control with primitives. (A) s on a linear scale for the Euclidean and mirror descent–like adaptation laws. Mirror descent converges faster for all tasks. (B) s on a logarithmic scale for the Euclidean and mirror descent–like adaptation laws. Mirror descent converges faster and minimizes s further for all tasks. (C, D) Parameter trajectories for the mirror descent and Euclidean adaptation laws. Mirror descent leads to smoother trajectories, with fewer parameters straying from 0.

law. The discrete changes of the desired trajectory are more visible in the parameter trajectories for the mirror-descent law.

8.3 Dynamics Prediction for Hamiltonian Systems. We now experimentally demonstrate the predictions of the theoretical calculations performed in section 4.2. Similar to Chen et al. (2020), consider the Hamiltonian for three point masses interacting in $d = 2$ dimensions via Newtonian gravitation (in units such that the gravitational constant $G = 1$),

$$\mathcal{H} = \frac{1}{2m_1}\|\mathbf{p}_1\|^2 + \frac{1}{2m_2}\|\mathbf{p}_2\|^2 + \frac{1}{2m_3}\|\mathbf{p}_3\|^2 - \frac{m_1 m_2}{\|\mathbf{q}_1 - \mathbf{q}_2\|} - \frac{m_1 m_3}{\|\mathbf{q}_1 - \mathbf{q}_3\|} - \frac{m_2 m_3}{\|\mathbf{q}_2 - \mathbf{q}_3\|}, \tag{8.8}$$

with m_i the mass of body i, $\mathbf{p}_i$ the momentum of body i, and $\mathbf{q}_i$ the position of body i. Denote by $\mathbf{q}$ the vector $\left(\mathbf{q}_1^{\mathrm{T}}, \mathbf{q}_2^{\mathrm{T}}, \mathbf{q}_3^{\mathrm{T}}\right)^{\mathrm{T}}$ with similar notation for $\mathbf{p}$.

It is well known that physical systems can often be described by a few common mechanisms (see, e.g., Feynman, Leighton, & Sands, 1977, sec.

12.7). As such, we consider estimating the Hamiltonian, equation 8.8, directly with a physically motivated overparameterized basis,

$$\hat{\mathcal{H}}(\hat{\mathbf{a}}) = \mathbf{Y}(\mathbf{q}, \mathbf{p})\hat{\mathbf{a}},$$

to form the dynamics predictor equations 4.3 and 4.4. We define $\mathbf{Y}(\mathbf{q}, \mathbf{p})$ to be a row vector of basis functions consisting of quadratics and quartics in $\mathbf{p}_i$ and $\mathbf{q}_i$, as well as $1/r_{ij}$, $1/r_{ij}^2$, and $1/r_{ij}^3$ potentials with $r_{ij} = \|\mathbf{q}_i - \mathbf{q}_j\|$ for $i \neq j$, comprising 21 total basis functions. These choices represent standard expressions for kinetic energy, spring potentials, central force potentials, and higher-order terms; any basis functions can be chosen motivated by knowledge of the physical system at hand.

We set $k = 5$, $\gamma = 3.5$, and choose $\psi(\cdot) = \frac{1}{2}\|\cdot\|_{1.05}^2$ to identify basis functions relevant to the observed trajectory. We fix $m_i = 1$ for all i and initialize $\mathbf{q}$ and $\mathbf{p}$ to lock the system in an oscillatory mode. Past $t = 10$, we set $k = \gamma = 0$ and run the predictor open-loop, as well as perform shrinkage and set all coefficients with magnitude below 10^{-3} formally equal to zero, leaving 13 remaining terms.

Results are shown in Figure 5. Figure 5A displays convergence of $\tilde{\mathbf{x}}$ to zero with adaptation (solid) and demonstrates that adaptation is necessary for convergence (dashed). When switching to the open-loop predictor past $t = 10$, the system without adaptation sustains large errors, while the learned predictor maintains good performance. The inset displays a slow drift of the predictor trajectory $\hat{\mathbf{x}}(t)$ from the true trajectory $\mathbf{x}(t)$ when run open-loop. Figure 5B displays the state trajectory $\mathbf{x}$ (dotted), convergence of $\hat{\mathbf{x}}$ with adaptation (solid) to $\mathbf{x}$, and the incorrect behavior of $\hat{\mathbf{x}}$ without adaptation (dashed). The open-loop unlearned predictor tends to a fixed point, while the open-loop learned predictor maintains the correct oscillatory behavior. Figures 5C and 5D show parameter trajectories and asymptotically converged parameters, respectively. Together, the two panes demonstrate that the implicit bias of the algorithm ensures convergence to a sparse estimate of the system Hamiltonian.

8.4 Sparse Identification of Chemical Reaction Networks. We now demonstrate an example of regularized adaptive dynamics prediction for an unknown chemical reaction network. Consider a set of chemical reactions with N distinct chemical species. Under the continuum hypothesis and the well-mixed assumption, mass-action kinetics dictates that the system dynamics can be described exactly in a monomial basis (Liu, Slotine, & Barabási, 2013),

$$\nu_j(\mathbf{x}) = k_j \Pi_{i=1}^N x_i^{a_{ji}},$$
$$\dot{\mathbf{x}} = \mathbf{\Gamma}\boldsymbol{\nu}(\mathbf{x}),$$

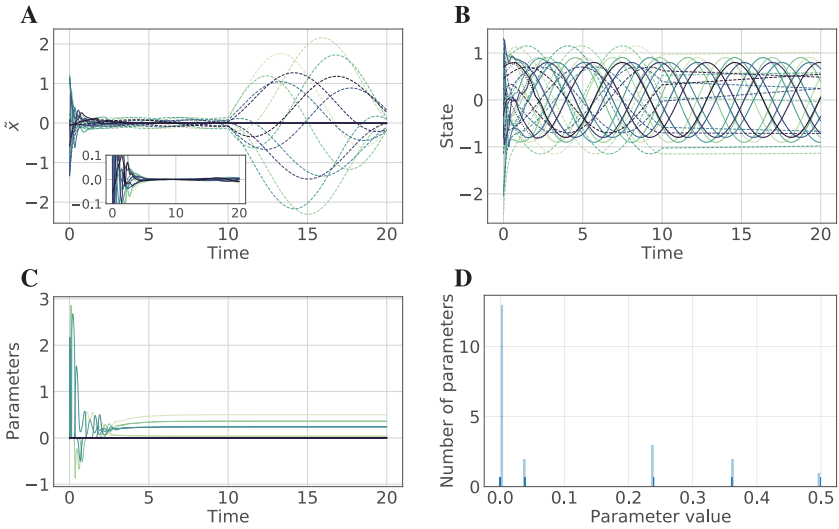

Figure 5: Three-body system. (A) Observer error $\tilde{\mathbf{x}} = \hat{\mathbf{x}} - \mathbf{x}$ for the adaptive dynamics predictor, equations 4.3 and 4.4, with adaptation (solid) and without adaptation (dashed). Inset shows the asymptotic behavior of the open-loop predictor after learning. (B) Convergence of $\hat{\mathbf{x}}$ with adaptation (solid) to $\mathbf{x}$ (dotted) for the adaptive dynamics predictor. $\hat{\mathbf{x}}$ without adaptation (dashed) does not converge to the true behavior. When run open-loop, the learned predictor maintains the correct oscillatory behavior, while the unlearned open-loop predictor incorrectly tends to a fixed point. (C) Parameter trajectories for the adaptive dynamics predictor. Many parameters stay at or near zero, as predicted by proposition 1. (D) Histogram of final parameter values learned by the adaptive dynamics predictor.

where x_i is the concentration of chemical species i, $\mathbf{\Gamma}$ is the stoichiometric matrix, and the a_{ji} are stoichiometric coefficients. Under the assumption that the full state of the network is measured, consider the adaptive dynamics predictor,

$$\dot{\hat{\mathbf{x}}} = \hat{\mathbf{\Gamma}}\hat{\boldsymbol{\nu}}(\hat{\mathbf{x}}) + k\,(\mathbf{x} - \hat{\mathbf{x}})\,, \tag{8.9}$$

$$\frac{d}{dt}\nabla\psi\left(\hat{\mathbf{\Gamma}}\right) = -\gamma(\hat{\mathbf{x}} - \mathbf{x})\hat{\boldsymbol{\nu}}(\hat{\mathbf{x}})^{\mathrm{T}}, \tag{8.10}$$

with $\gamma > 0$ a positive learning rate, $k > 0$ an observer gain, ψ a strongly convex function, $\hat{\mathbf{\Gamma}}$ an estimate of the stoichiometric matrix, and $\hat{\boldsymbol{\nu}}(\hat{\mathbf{x}})$ a vector of monomial basis functions representing available knowledge of the system. Here we consider a four-species chemical reaction network (see Liu et al.,

2013, supplementary information),

$$\begin{aligned}
\dot{x}_1 &= -k_1 x_1 x_2, \\
\dot{x}_2 &= -k_1 x_1 x_2 - k_2 x_2 x_3^2, \\
\dot{x}_3 &= -k_1 x_1 x_2 - 2k_2 x_2 x_3^2, \\
\dot{x}_4 &= k_2 x_2 x_3^2,
\end{aligned}$$

with corresponding adaptive dynamics predictor equations 8.9 and 8.10. We set $\hat{\boldsymbol{\nu}}$ to be a vector of all monomials up to degree 3 comprising a total of 140 candidate basis functions, and we set $\psi(\hat{\boldsymbol{\Gamma}}) = \frac{1}{2}\left\|\mathrm{vec}\left(\hat{\boldsymbol{\Gamma}}\right)\right\|_{1.01}^2$ to identify a sparse, parsimonious model consistent with the data. Searching over sparse models ensures that our learned predictor selects only a few relevant terms in the approximate system dynamics. We fix $k = 1.5$ and $\gamma = 0.25$ for $t < 10$. As in section 8.3, past $t = 10$, we set $k = \gamma = 0$ and run the predictor open-loop. We also perform shrinkage and set all coefficients with magnitude below 10^{-3} formally equal to zero, leaving 19 remaining parameters.

Results are shown in Figure 6. In Figure 6A, we show convergence of the observer error to zero with adaptation (solid) and divergence away from zero without adaptation (dashed), demonstrating that adaptation is necessary for effective prediction. The inset displays a closer look at the asymptotic behavior of the open-loop dynamics predictor after shrinkage, which shows that the fixed point of the system is correctly learned. Figure 6B shows convergence of $\hat{\mathbf{x}}$ (solid) to $\mathbf{x}$ (dashed). Figure 6C displays parameter trajectories as a function of time. Many parameters stay at or near zero as predicted by proposition 1. The inset displays a finer-grained view around zero of the parameter trajectories. Figure 6D shows a histogram of the final parameter values learned by the adaptive dynamics predictor, demonstrating that only a few relevant terms are identified.

9 Conclusion and Future Directions

It is somewhat unusual in nonlinear control to have a choice between a large variety of algorithms that can all be proven to globally converge. Nevertheless, in this article, we have presented a suite of new globally convergent adaptive control algorithms. The algorithms combine the velocity gradient methodology (Fradkov, 1980; Fradkov et al., 1999) with the Bregman Lagrangian (Betancourt et al., 2018; Wibisono et al., 2016) to systematically generate velocity gradient algorithms with momentum. Based on analogies between isotonic regression (Goel & Klivans, 2017; Goel et al., 2018; Kakade et al., 2011) and algorithms for nonlinearly parameterized adaptive control (Tyukin, 2011; Tyukin et al., 2007), we extended our higher-order velocity gradient algorithms to the nonlinearly parameterized setting of generalized

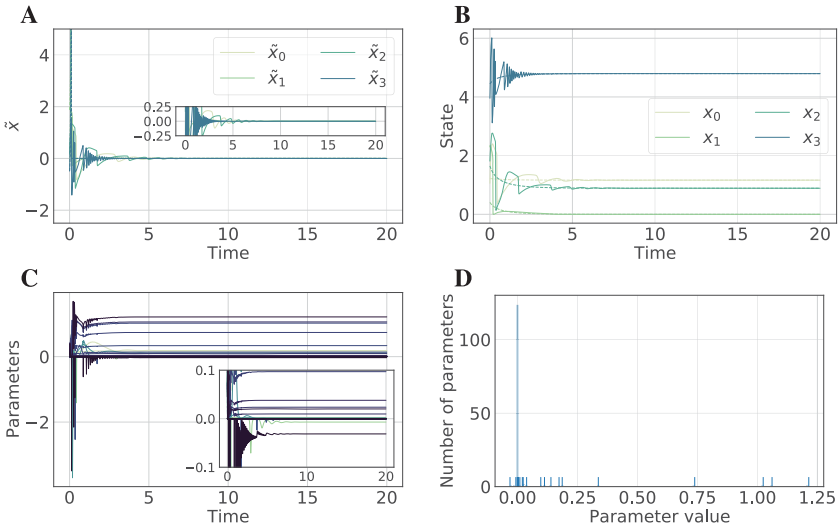

Figure 6: Chemical reaction network. (A) Observer error $\tilde{\mathbf{x}} = \hat{\mathbf{x}} - \mathbf{x}$ for the adaptive dynamics predictor, equations 8.9 and 8.10, with adaptation (solid) and without adaptation (dashed). The predictor without adaptation diverges immediately. Past $t = 10$, shrinkage is performed, all coefficients with magnitude below 10^{-3} are set to zero, and the predictor is run open-loop with $k = \gamma = 0$. (B) Convergence of $\hat{\mathbf{x}}$ to $\mathbf{x}$ for the adaptive dynamics predictor. The predictor accurately learns the fixed point of the system and stays stationary when run open-loop. (C) Parameter trajectories for the adaptive dynamics predictor. Many parameters stay at or near zero, as predicted by proposition 1. (D) Histogram of final parameter values learned by the adaptive dynamics predictor. Only a few relevant terms are identified.

linear models. Using a similar parallel to distributed stochastic gradient descent algorithms (Boffi & Slotine, 2020; Zhang et al., 2014), we developed a stable modification of all of our algorithms. We subsequently fused our developments with time-dependent learning rates based on the bounded gain forgetting formalism (Slotine & Li, 1991).

By consideration of the non-Euclidean Bregman Lagrangian, we derived natural gradient (Amari, 1998) and mirror descent (Beck & Teboulle, 2003; Nemirovski & Yudin, 1983)–like algorithms with momentum. Taking the infinite friction limit of these algorithms recovers a recent algorithm for adaptive robot control (Lee et al., 2018) that respects physical Riemannian constraints on the parameters throughout adaptation. By extending recent results on the implicit bias of optimization algorithms in machine learning (Azizan & Hassibi, 2019; Azizan et al., 2019) to the continuous-time

setting, we proved that these mirror descent-like algorithms in the first-order, second-order, and nonlinearly parameterized settings impose implicit regularization on the parameter vectors found by adaptive control.

Throughout the article, for simplicity of exposition, we focused on the nth order system, equation 2.1. As discussed in remark 2, our results extend to more general systems that have an error model similar to equation 2.6, in the sense that the proof technique summarized by lemma 2 is roughly preserved. The nth order system structure makes the employed proportional-integral forms simple, as they can be written down explicitly as in equations 6.14 to 6.17. As summarized in remark 23, a PDE needs to be solved in the general case, and solutions to this PDE may not exist. Solution of the PDE can be avoided by the dynamic scaling technique of Karagiannis et al. (2009) or a similar embedding technique of Tyukin (2011).

A significant outstanding question is whether there is an empirical advantage to using our proposed momentum algorithms. In optimization, accelerated algorithms generated by the Bregman Lagrangian provide faster convergence when properly discretized, and it is thus likely that a careful discretization is necessary to obtain optimal performance of our momentum algorithms. However, we are not aware of any available convergence rates in adaptive control, and it would be necessary to prove such rates to understand analytically if there is an advantage. Similar higher-order algorithms have appeared in the literature for linear systems of relative degree greater than one (Fradkov et al., 1999; Morse, 1992), where first-order algorithms cannot control the system. Here we have focused on feedback linearizable systems, and perhaps there exist classes of nonlinear systems that cannot be adaptively controlled with a first-order algorithm but can with a momentum algorithm. We leave the investigation of these interesting and important questions to future work.

Appendix A: Omitted Proofs and Required Results

Barbalat's lemma is a classical technique in adaptive control theory, which is used in conjuction with a Lyapunov-like analysis to prove convergence of a given signal.

Lemma 1. *(Barbalat's Lemma (Slotine & Li, 1991)). Assume that* $\lim_{t\to\infty} \int_0^t |x(\tau)|d\tau < \infty$. *If* $x(t)$ *is uniformly continuous, then* $\lim_{t\to\infty} x(t) = 0$.

Note that a sufficient condition for uniform continuity of $x(t)$ is for $\dot{x}(t)$ to be bounded. Hence, for any signal $x(t) \in \mathcal{L}_2 \cap \mathcal{L}_\infty$ with $\dot{x}(t) \in \mathcal{L}_\infty$, we can apply lemma 1 to the signal $x^2(t)$ and conclude that $x(t) \to 0$.

Lemma 2. *Assume that* $\int_0^t \tilde{f}^2(\mathbf{x}(t'), \hat{\mathbf{a}}(t'), \mathbf{a}, t')dt' < \infty$ *where* $[0, T)$ *is the maximal interval of existence of* $\mathbf{x}(t)$. *Further assume that* $\hat{\mathbf{a}}(t)$ *is bounded over* $[0, T)$,

that both bounds are independent of T, and that $\tilde{f}$ is locally bounded in $\mathbf{x}$ and $\hat{\mathbf{a}}$ uniformly in t. Then $\hat{\mathbf{a}} \in \mathcal{L}_\infty$, $\tilde{f} \in \mathcal{L}_2$, $s \in \mathcal{L}_2 \cap \mathcal{L}_\infty$, $s \to 0$ and $\mathbf{x} \to \mathbf{x}_d$.

Proof. By equation 2.6, we can write explicitly

$$s(t) = \int_0^t e^{-\eta(t-\tau)} \tilde{f}(\mathbf{x}(\tau), \hat{\mathbf{a}}(\tau), \mathbf{a}, \tau) d\tau. \tag{A.1}$$

By the Cauchy-Schwarz inequality,

$$\begin{aligned} s^2(T) &\le \left(\int_0^T e^{-2\eta(T-\tau)} d\tau \right) \left(\int_0^T \tilde{f}^2(\tau) d\tau \right) \\ &\le \frac{1}{2\eta} \left(\int_0^T \tilde{f}^2(\tau) d\tau \right) \left(1 - e^{-2\eta T}\right) \\ &\le \frac{1}{2\eta} \left(\int_0^T \tilde{f}^2(\tau) d\tau \right) \end{aligned}$$

so that $\sup_{t\in[0,T)} |s(t)| < \infty$. Observe that this bound is independent of T. It immediately follows that $\sup_{t\in[0,T)} \|\mathbf{x}(t)\| < \infty$ and that this bound is independent of T. This observation contradicts that $[0, T)$ is the maximal interval of existence of $\mathbf{x}(t)$ for any T, and thus $\mathbf{x}(t)$ must exist for all t. This shows that $\mathbf{x} \in \mathcal{L}_\infty$, $s \in \mathcal{L}_\infty$, and that the bounds on $\tilde{f}$ and $\hat{\mathbf{a}}$ can be extended for all t. From this we conclude $\tilde{f} \in \mathcal{L}_2$ and $\hat{\mathbf{a}} \in \mathcal{L}_\infty$. Similarly, Parseval's theorem applied to the low-pass filter, equation A.1, shows that $s \in \mathcal{L}_2$. Because $\mathbf{x} \in \mathcal{L}_\infty$ and $\hat{\mathbf{a}} \in \mathcal{L}_\infty$, and because $\tilde{f}$ is locally bounded in $\mathbf{x}$ and $\hat{\mathbf{a}}$ uniformly in t, $\tilde{f} \in \mathcal{L}_\infty$. By equation 2.6, $\dot{s} \in \mathcal{L}_\infty$, and hence by Barbalat's lemma (Lemma 1), $s \to 0$. By definition of s, we then conclude that $\mathbf{x} \to \mathbf{x}_d$. □

A.1 Proof of Proposition 8.

Proof. Consider the Lyapunov-like function,

$$V = \frac{1}{2} s^2 + \frac{1}{2\gamma} \tilde{\mathbf{a}}^{\mathrm{T}} \mathbf{P}^{-1} \tilde{\mathbf{a}},$$

which has time derivative

$$\dot{V} = -\eta s^2 - \frac{\kappa}{\gamma} \tilde{f}^2.$$

This immediately shows $s \in \mathcal{L}_\infty$ and $\hat{\mathbf{a}} \in \mathcal{L}_\infty$. Because $s \in \mathcal{L}_\infty$, $\mathbf{x} \in \mathcal{L}_\infty$ by definition of the sliding variable (Slotine & Li, 1991). Integrating $\dot{V}$ shows that $s \in \mathcal{L}_2$ and $\tilde{f} \in \mathcal{L}_2$. The result follows by application of lemma 2 or directly by Barbalat's lemma (lemma 1). □

A.2 Proof of Proposition 9.

Proof. Consider the Lyapunov function,

$$V = \frac{1}{2}s^2 + \frac{1}{2\gamma}\left(\|\tilde{\mathbf{v}}\|^2 + \|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2\right),$$

which has time derivative

$$\begin{aligned}
\dot{V} &= -\eta s^2 + s\tilde{f} + \frac{1}{\gamma}\big[\tilde{\mathbf{v}}^{\mathrm{T}}\left(-\kappa\tilde{f} - \gamma s\right)\mathbf{Y}^{\mathrm{T}} + (\hat{\mathbf{a}} - \hat{\mathbf{v}})^{\mathrm{T}}\left(\beta\mathcal{N}(\hat{\mathbf{v}} - \hat{\mathbf{a}}) + \gamma s\mathbf{Y}^{\mathrm{T}}\right. \\
&\quad \left. + \kappa\tilde{f}\mathbf{Y}^{\mathrm{T}}\right)\big] \\
&= -\eta s^2 - \frac{\kappa}{\gamma}\tilde{f}^2 - \frac{\beta}{\gamma}\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2 - \frac{\beta\mu}{\gamma}\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|\|\mathbf{Y}\|^2 + 2s(\hat{\mathbf{a}} - \hat{\mathbf{v}})^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}} \\
&\quad + 2\frac{\kappa}{\gamma}\tilde{f}(\hat{\mathbf{a}} - \hat{\mathbf{v}})^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}} \\
&\leq -\eta s^2 - \frac{\kappa}{\gamma}\tilde{f}^2 - \frac{\beta}{\gamma}\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2 - \frac{\beta\mu}{\gamma}\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|\|\mathbf{Y}\|^2 + 2|s|\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|\|\mathbf{Y}\| \\
&\quad + 2\frac{\kappa}{\gamma}|\tilde{f}|\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|\|\mathbf{Y}\| \\
&\leq -\epsilon_1\eta s^2 - \epsilon_2\frac{\kappa}{\gamma}\tilde{f}^2 - \left(\sqrt{(1-\epsilon_1)\eta}|s| - \frac{1}{\sqrt{(1-\epsilon_1)\eta}}\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|\|\mathbf{Y}\|\right)^2 \\
&\quad - \left(\sqrt{\frac{(1-\epsilon_2)\kappa}{\gamma}}|\tilde{f}| - \frac{\kappa}{\gamma}\sqrt{\frac{\gamma}{(1-\epsilon_2)\kappa}}\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|\|\mathbf{Y}\|\right)^2 - \frac{\beta}{\gamma}\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2,
\end{aligned}$$

where $0 < \epsilon_1 < 1$ and $0 < \epsilon_2 < 1$ are arbitrary and where we have taken $\mu = \frac{\gamma}{\beta}\left(\frac{1}{(1-\epsilon_1)\eta} + \frac{\kappa}{(1-\epsilon_2)\gamma}\right)$. Because ϵ_1 and ϵ_2 are arbitrary, this shows that $\dot{V}$ is negative semidefinite for $\mu > \frac{\gamma}{\beta}\left(\frac{1}{\eta} + \frac{\kappa}{\gamma}\right)$. Hence, $\hat{\mathbf{v}} \in \mathcal{L}_\infty$, $\hat{\mathbf{a}} \in \mathcal{L}_\infty$, and $s \in \mathcal{L}_\infty$. Because $s \in \mathcal{L}_\infty$, we automatically have $\mathbf{x} \in \mathcal{L}_\infty$, which shows that $\dot{s} \in \mathcal{L}_\infty$ by local boundedness of $\tilde{f}$ in $\mathbf{x}$ and $\hat{\mathbf{a}}$ uniformly in t. Integrating $\dot{V}$ shows that $s \in \mathcal{L}_2$ and hence by Barbalat's lemma (lemma 1) $s \to 0$ and $\mathbf{x} \to \mathbf{x}_d$. □

A.3 Proof of Proposition 10.

Proof. Defining the vector $\hat{\mathbf{v}}^t = \sum_{i=1}^m \hat{w}_i^t \boldsymbol{\phi}(\mathbf{x}_i)$, equation 6.22 implies the iteration on $\hat{\mathbf{v}}$,

$$\hat{\mathbf{v}}^{t+1} = \hat{\mathbf{v}}^t - \frac{\lambda}{m}\sum_{i=1}^m \left(\hat{f}(\hat{\mathbf{w}}^t, \mathbf{x}_i) - f(\mathbf{x}_i)\right)\boldsymbol{\phi}(\mathbf{x}_i). \tag{A.2}$$

Equation A.2 shows that at time t,

$$\hat{\mathbf{v}}^t = -\frac{\lambda}{m}\sum_{i=1}^{m}\left(\sum_{j=1}^{t-1}\tilde{f}_i^j\right)\boldsymbol{\phi}(\mathbf{x}_i), \tag{A.3}$$

where $\tilde{f}_i^j$ in equation A.3 is the function approximation error on the ith input example at iteration j, $\tilde{f}_i^j = \hat{f}(\hat{\mathbf{w}}^j, \mathbf{x}_i) - f(\mathbf{x}_i)$.

Now, assuming that for the adaptive control problem $f(\mathbf{x}, \mathbf{a}, t) = u(\boldsymbol{\alpha}^{\mathrm{T}}(\mathbf{x}, t)\mathbf{a})$, setting $\mathbf{P} = \lambda\mathbf{I}$, $\hat{\mathbf{a}}(0) = \mathbf{0}$, and integrating both sides of equation 2.12, we see that at time t,

$$\hat{\mathbf{a}}(t) = -\lambda\int_0^t \tilde{f}(\mathbf{x}(t'), \hat{\mathbf{a}}(t'), \mathbf{a}, t')\boldsymbol{\alpha}(\mathbf{x}(t'), t')dt'. \tag{A.4}$$

The current function approximation $\hat{f}$ at time t for the parameters in equation A.4 can then be written as

$$\begin{aligned}\hat{f}(t) &= u\left(\boldsymbol{\alpha}^{\mathrm{T}}(\mathbf{x}, t)\hat{\mathbf{a}}(t)\right) = u\left(\int_0^t -\lambda\tilde{f}(\mathbf{x}(t'), \hat{\mathbf{a}}(t'), \mathbf{a}, t')\boldsymbol{\alpha}^{\mathrm{T}}(\mathbf{x}(t), t)\boldsymbol{\alpha}(\mathbf{x}(t'), t')dt'\right)\\ &= u\left(\int_0^t c(t')\mathcal{K}(t, t')dt'\right),\end{aligned} \tag{A.5}$$

where we have defined $c(t') = -\lambda\tilde{f}(\mathbf{x}(t'), \hat{\mathbf{a}}(t'), \mathbf{a}, t')$ and $\mathcal{K}(t, t') = \boldsymbol{\alpha}^{\mathrm{T}}(\mathbf{x}(t), t)\boldsymbol{\alpha}(\mathbf{x}(t'), t')$. Similarly, in the case of the Alphatron, the current approximation at iteration t is given by

$$\begin{aligned}\hat{f}(\hat{\mathbf{w}}^t, \mathbf{x}) &= u\left(\langle\hat{\mathbf{v}}^t, \boldsymbol{\phi}(\mathbf{x})\rangle_{\mathcal{H}}\right) = u\left(\sum_{i=1}^{m}\left(\sum_{j=1}^{t-1} -\frac{\lambda}{m}\tilde{f}_i^j\right)\langle\boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{x}_i)\rangle\right)\\ &= u\left(\sum_{i=1}^{m}\hat{w}_i^t\mathcal{K}(\mathbf{x}, \mathbf{x}_i)\right),\end{aligned} \tag{A.6}$$

where we have noted that with $\hat{w}_i^0 = 0$ for all i, $\hat{w}_i^t = \sum_{j=1}^{t-1} -\frac{\lambda}{m}\tilde{f}_i^j$. □

A.4 Proof of Proposition 11.

Proof. Consider the Lyapunov function candidate,

$$V = \frac{1}{2\gamma}\|\tilde{\mathbf{v}}\|^2 + \frac{1}{2\gamma}\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2,$$

which has time derivative

$$\begin{aligned}\dot{V} &= \frac{1}{\gamma}\tilde{\mathbf{v}}^{\mathrm{T}}\left(-\gamma\tilde{f}\boldsymbol{\alpha}\right) + \frac{1}{\gamma}\left(\hat{\mathbf{a}}-\hat{\mathbf{v}}\right)^{\mathrm{T}}\left(\beta\mathcal{N}\left(\hat{\mathbf{v}}-\hat{\mathbf{a}}\right)+\gamma\tilde{f}\boldsymbol{\alpha}\right)\\ &= -\left(\tilde{\mathbf{a}}^{\mathrm{T}}\boldsymbol{\alpha}\right)\tilde{f} - \frac{\beta}{\gamma}\mathcal{N}\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\|^2 + 2\left(\hat{\mathbf{a}}-\hat{\mathbf{v}}\right)^{\mathrm{T}}\boldsymbol{\alpha}\tilde{f}\\ &\leq -\frac{\tilde{f}^2}{D_1} - \frac{\beta}{\gamma}\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\|^2 - \frac{\beta\mu}{\gamma}\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\|^2\|\boldsymbol{\alpha}\|^2 + 2\|\boldsymbol{\alpha}\|\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\||\tilde{f}|\\ &\leq -\frac{\epsilon}{D_1}\tilde{f}^2 - \beta\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\|^2 - \left(\sqrt{\frac{1-\epsilon}{D_1}}|\tilde{f}| - \sqrt{\frac{D_1}{1-\epsilon}}\|\boldsymbol{\alpha}\|\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\|\right)^2,\end{aligned}$$

where $0 < \epsilon < 1$ is arbitrary and we have chosen $\mu = \frac{\gamma D_1}{(1-\epsilon)\beta}$. Because ϵ is arbitrary, this shows that $\hat{\mathbf{v}}$ and $\hat{\mathbf{a}}$ remain bounded for $\mu > \frac{\gamma D_1}{\beta}$ over the maximal interval of existence of $\mathbf{x}(t)$. By integrating $\dot{V}$, we see that $\tilde{f} \in \mathcal{L}_2$ over this same interval. Note that the bounds are independent of the length of the interval. Application of lemma 2 completes the proof. □

A.5 Proof of Proposition 12.

Proof. The Lyapunov-like function,

$$V = \frac{1}{2}\left(\tilde{\mathbf{a}}^{\mathrm{T}}\mathbf{P}^{-1}\tilde{\mathbf{a}} + \tilde{\tilde{\mathbf{a}}}^{\mathrm{T}}\mathbf{P}^{-1}\tilde{\tilde{\mathbf{a}}} + s^2\right),$$

has time derivative

$$\dot{V} = -\eta s^2 - k\left(\bar{\mathbf{a}}-\hat{\mathbf{a}}\right)^{\mathrm{T}}\mathbf{P}^{-1}\left(\bar{\mathbf{a}}-\hat{\mathbf{a}}\right).$$

This shows that s, $\hat{\mathbf{a}}$, and $\bar{\mathbf{a}}$ remain bounded. The remaining conclusions of the proposition are immediately drawn by integrating $\dot{V}$ and applying Barbalat's lemma (lemma 1). □

A.6 Proof of Proposition 13.

Proof. The Lyapunov-like function,

$$V = \frac{1}{2}\left(\tilde{\mathbf{a}}^{\mathrm{T}}\mathbf{P}^{-1}\tilde{\mathbf{a}} + \tilde{\tilde{\mathbf{a}}}^{\mathrm{T}}\mathbf{P}^{-1}\tilde{\tilde{\mathbf{a}}}\right),$$

has time derivative

$$\dot{V} \leq -\frac{1}{D_1}\tilde{f}^2 - k\left(\hat{\mathbf{a}}-\bar{\mathbf{a}}\right)^{\mathrm{T}}\mathbf{P}^{-1}\left(\hat{\mathbf{a}}-\bar{\mathbf{a}}\right).$$

This shows that $\hat{\mathbf{a}}$ and $\bar{\mathbf{a}}$ remain bounded over the maximal interval of existence of $\mathbf{x}(t)$. Integration of $\dot{V}$ shows $\tilde{f} \in \mathcal{L}_2$ and $(\hat{\mathbf{a}} - \bar{\mathbf{a}}) \in \mathcal{L}_2$ over the same interval. Note that the bounds are independent of the length of the interval. Application of lemma 2 completes the proof. □

A.7 Proof of Proposition 14.

Proof. The Lyapunov-like function,

$$V = \frac{1}{2\gamma}\left(\|\tilde{\mathbf{v}}\|^2 + \|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2 + \|\bar{\mathbf{a}} - \hat{\mathbf{a}}\|^2\right),$$

has time derivative

$$\begin{aligned}
\dot{V} &= -\left(\tilde{\mathbf{a}}^{\mathrm{T}}\boldsymbol{\alpha}\right)\tilde{f} - \frac{\beta}{\gamma}\mathcal{N}\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2 + \frac{k\beta}{\gamma}\mathcal{N}\left(\hat{\mathbf{a}} - \hat{\mathbf{v}}\right)^{\mathrm{T}}\left(\bar{\mathbf{a}} - \hat{\mathbf{a}}\right) + 2\tilde{f}\left(\hat{\mathbf{a}} - \hat{\mathbf{v}}\right)^{\mathrm{T}}\boldsymbol{\alpha} \\
&\quad -2\frac{k\beta}{\gamma}\mathcal{N}\|\hat{\mathbf{a}} - \bar{\mathbf{a}}\|^2 + \frac{\beta}{\gamma}\mathcal{N}\left(\hat{\mathbf{a}} - \bar{\mathbf{a}}\right)^{\mathrm{T}}\left(\hat{\mathbf{v}} - \hat{\mathbf{a}}\right) \\
&\leq -\frac{\tilde{f}^2}{D_1} - \frac{\beta\mathcal{N}}{2\gamma}(1-k)\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2 - \frac{\beta\mathcal{N}}{2\gamma}\|\hat{\mathbf{a}} - \bar{\mathbf{a}}\|^2(3k-1) + 2|\tilde{f}|\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|\|\boldsymbol{\alpha}\| \\
&\leq -\frac{\tilde{f}^2}{D_1} - \frac{\beta}{2\gamma}(1-k)\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2 - \frac{\beta\mu}{2\gamma}(1-k)\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2\|\boldsymbol{\alpha}\|^2 \\
&\quad -\frac{\beta\mathcal{N}}{2\gamma}\|\hat{\mathbf{a}} - \bar{\mathbf{a}}\|^2(3k-1) + 2|\tilde{f}|\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|\|\boldsymbol{\alpha}\| \\
&\leq -\frac{\epsilon}{D_1}\tilde{f}^2 - \left(\frac{\sqrt{1-\epsilon}|\tilde{f}|}{\sqrt{D_1}} - \sqrt{\frac{D_1}{1-\epsilon}}\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|\|\boldsymbol{\alpha}\|\right)^2 - \frac{\beta}{2\gamma}(1-k)\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2 \\
&\quad -\frac{\beta\mathcal{N}}{2\gamma}(3k-1)\|\hat{\mathbf{a}} - \bar{\mathbf{a}}\|^2,
\end{aligned}$$

where $0 < \epsilon < 1$ is arbitrary and we have chosen $\mu = \frac{2\gamma D_1}{\beta(1-\epsilon)(1-k)}$. From above, we conclude $\hat{\mathbf{v}}$, $\hat{\mathbf{a}}$, and $\bar{\mathbf{a}}$ remain bounded over the maximal interval of existence of $\mathbf{x}(t)$ for $\frac{1}{3} \leq k < 1$. By integrating $\dot{V}$, we see that $\tilde{f} \in \mathcal{L}_2$, $(\hat{\mathbf{a}} - \bar{\mathbf{a}}) \in \mathcal{L}_2$, and $(\hat{\mathbf{a}} - \hat{\mathbf{v}}) \in \mathcal{L}_2$ over the same interval. Note that the bounds are independent of the length of the interval. Application of lemma 2 completes the proof. □

A.8 Proof of Proposition 15.

Proof. The Lyapunov-like function,

$$V = \frac{1}{\gamma}\left(\|\tilde{\mathbf{v}}\|^2 + \|\tilde{\bar{\mathbf{v}}}\|^2 + \|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2\right),$$

has time derivative

$$\begin{aligned}\dot{V} &= -\left(\tilde{\mathbf{a}}^{\mathrm{T}}\boldsymbol{\alpha}\right)\tilde{f} + 2\tilde{f}\left(\hat{\mathbf{a}}-\hat{\mathbf{v}}\right)^{\mathrm{T}}\boldsymbol{\alpha} - \frac{\beta\mathcal{N}}{\gamma}\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\|^2 - \frac{\rho}{\gamma}\|\hat{\mathbf{v}}-\bar{\mathbf{v}}\|^2 \\ &\quad - \frac{\rho}{\gamma}\left(\hat{\mathbf{a}}-\hat{\mathbf{v}}\right)^{\mathrm{T}}\left(\bar{\mathbf{v}}-\hat{\mathbf{v}}\right) \\ &\leq -\frac{\tilde{f}^2}{D_1} + 2|\tilde{f}|\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\|\|\boldsymbol{\alpha}\| - \left(\frac{\beta}{\gamma}-\frac{\rho}{2\gamma}\right)\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\|^2 - \frac{\beta\mu}{\gamma}\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\|^2\|\boldsymbol{\alpha}\|^2 \\ &\quad - \frac{\rho}{2\gamma}\|\hat{\mathbf{v}}-\bar{\mathbf{v}}\|^2 \\ &\leq -\frac{\epsilon}{D_1}\tilde{f}^2 - \left(\sqrt{\frac{1-\epsilon}{D_1}}|\tilde{f}| - \sqrt{\frac{D_1}{1-\epsilon}}\|\hat{\mathbf{v}}-\hat{\mathbf{a}}\|\|\boldsymbol{\alpha}\|\right)^2 - \frac{\rho}{2\gamma}\|\bar{\mathbf{v}}-\hat{\mathbf{v}}\|^2 \\ &\quad - \frac{1}{2\gamma}\left(2\beta-\rho\right)\|\hat{\mathbf{v}}-\hat{\mathbf{a}}\|^2\end{aligned}$$

where $0 < \epsilon < 1$ is arbitrary and we have chosen $\mu = \frac{\gamma D_1}{\beta(1-\epsilon)}$. From above, we conclude $\hat{\mathbf{v}}$, $\bar{\mathbf{v}}$, and $\hat{\mathbf{a}}$ remain bounded over the maximal interval of existence of $\mathbf{x}(t)$ for $\rho < 2\beta$. Integrating $\dot{V}$ shows that $\tilde{f} \in \mathcal{L}_2$, $(\hat{\mathbf{v}}-\bar{\mathbf{v}}) \in \mathcal{L}_2$, and $(\hat{\mathbf{v}}-\hat{\mathbf{a}}) \in \mathcal{L}_2$ over the same interval. Note that the bounds are independent of the length of the interval. Application of lemma 2 completes the proof. □

A.9 Proof of Proposition 16.

Proof. The Lyapunov-like function,

$$V = \frac{1}{2\gamma}\left(\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\|^2 + \|\hat{\mathbf{a}}-\bar{\mathbf{a}}\|^2 + \|\tilde{\mathbf{v}}\|^2 + \|\tilde{\bar{\mathbf{v}}}\|^2\right),$$

has time derivative

$$\begin{aligned}\dot{V} &= -\left(\tilde{\mathbf{a}}^{\mathrm{T}}\boldsymbol{\alpha}\right)\tilde{f} - \frac{\beta\mathcal{N}}{\gamma}\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\|^2 - \frac{2k\beta\mathcal{N}}{\gamma}\|\bar{\mathbf{a}}-\hat{\mathbf{a}}\|^2 + 2\tilde{f}\left(\hat{\mathbf{a}}-\hat{\mathbf{v}}\right)^{\mathrm{T}}\boldsymbol{\alpha} \\ &\quad + \beta N\left(\hat{\mathbf{a}}-\bar{\mathbf{a}}\right)^{\mathrm{T}}\left(\hat{\mathbf{v}}-\hat{\mathbf{a}}\right) + \frac{k\beta\mathcal{N}}{\gamma}\left(\hat{\mathbf{a}}-\hat{\mathbf{v}}\right)^{\mathrm{T}}\left(\bar{\mathbf{a}}-\hat{\mathbf{a}}\right) - \frac{\rho}{\gamma}\|\hat{\mathbf{v}}-\bar{\mathbf{v}}\|^2 \\ &\quad - \frac{\rho}{\gamma}\left(\hat{\mathbf{a}}-\hat{\mathbf{v}}\right)^{\mathrm{T}}\left(\bar{\mathbf{v}}-\hat{\mathbf{v}}\right) \\ &\leq -\frac{1}{D_1}\tilde{f}^2 - \frac{1}{2\gamma}\left(\beta\left(1-k\right)-\rho\right)\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\|^2 - \frac{1}{2\gamma}\left(\beta\mu\left(1-k\right)\right)\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\|^2\|\boldsymbol{\alpha}\|^2 \\ &\quad - \frac{\mathcal{N}\beta}{2\gamma}\left(3k-1\right)\|\bar{\mathbf{a}}-\hat{\mathbf{a}}\|^2 - \frac{\rho}{2\gamma}\|\hat{\mathbf{v}}-\bar{\mathbf{v}}\|^2 + 2|\tilde{f}|\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\|\|\mathbf{Y}\|\end{aligned}$$

$$\leq -\frac{\epsilon}{D_1}\tilde{f}^2 - \left(\sqrt{\frac{1-\epsilon}{D_1}}|\tilde{f}| - \sqrt{\frac{D_1}{1-\epsilon}}\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\|\|\boldsymbol{\alpha}\|\right)^2 - \frac{\rho}{2\gamma}\|\overline{\mathbf{v}}-\hat{\mathbf{v}}\|^2$$
$$-\frac{\beta\mathcal{N}}{2\gamma}(3k-1)\|\hat{\mathbf{a}}-\overline{\mathbf{a}}\|^2 - \frac{1}{2\gamma}\left((1-k)\beta-\rho\right)\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\|^2,$$

where $0 < \epsilon < 1$ is arbitrary and we have chosen $\mu = \frac{2\gamma D_1}{\beta(1-k)(1-\epsilon)}$. This immediately shows that $\hat{\mathbf{a}}$, $\hat{\mathbf{v}}$, $\overline{\mathbf{a}}$, and $\overline{\mathbf{v}}$ remain bounded over the maximal interval of existence of $\mathbf{x}(t)$ for $\frac{1}{3} \leq k < 1$ and $\rho < \beta(1-k)$. Integrating $\dot{V}$ shows that $\tilde{f} \in \mathcal{L}_2$, $(\overline{\mathbf{v}}-\hat{\mathbf{v}}) \in \mathcal{L}_2$, $(\hat{\mathbf{a}}-\overline{\mathbf{a}}) \in \mathcal{L}_2$, and $(\hat{\mathbf{a}}-\hat{\mathbf{v}}) \in \mathcal{L}_2$ over the same interval. Note that the bounds are independent of the length of the interval. Application of lemma 2 completes the proof. □

A.10 Proof of Proposition 17.

Proof. The Lyapunov-like function,

$$V = \frac{1}{2}s^2 + \frac{1}{2}\tilde{\mathbf{a}}^{\mathrm{T}}\mathbf{P}^{-1}\tilde{\mathbf{a}},$$

has time derivative

$$\dot{V} = -\eta s^2 - \frac{1}{2}\tilde{f}^2 - \frac{\lambda}{2}\tilde{\mathbf{a}}^{\mathrm{T}}\mathbf{P}^{-1}\tilde{\mathbf{a}},$$

which shows that s and $\hat{\mathbf{a}}$ remain bounded. Because s remains bounded, $\mathbf{x}$ remains bounded. Integrating $\dot{V}$ shows that $s \in \mathcal{L}_2$ and $\tilde{f} \in \mathcal{L}_2$. The proof is completed by application of lemma 2 or directly by Barbalat's Lemma (lemma 1). □

A.11 Proof of Proposition 18.

Proof. Consider the Lyapunov-like function,

$$V = \frac{1}{2}s^2 + \frac{1}{2}\tilde{\mathbf{v}}^{\mathrm{T}}\mathbf{P}^{-1}\tilde{\mathbf{v}} + \frac{1}{2}\left(\hat{\mathbf{v}}-\hat{\mathbf{a}}\right)^{\mathrm{T}}\mathbf{P}^{-1}\left(\hat{\mathbf{v}}-\hat{\mathbf{a}}\right),$$

which has time derivative

$$\dot{V} = -\eta s^2 + s\tilde{f} - \left(\hat{\mathbf{v}}-\hat{\mathbf{a}}+\tilde{\mathbf{a}}\right)^{\mathrm{T}}\left(s+\tilde{f}\right)\mathbf{Y}^{\mathrm{T}} + \frac{1}{2}\left(\tilde{\mathbf{v}}^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}\right)^2 - \frac{\lambda(t)}{2}\tilde{\mathbf{v}}^{\mathrm{T}}\mathbf{P}^{-1}\tilde{\mathbf{v}}$$
$$+\left(\hat{\mathbf{v}}-\hat{\mathbf{a}}\right)^{\mathrm{T}}\left(-\beta\mathcal{N}\left(\hat{\mathbf{v}}-\hat{\mathbf{a}}\right) - \left(s+\tilde{f}\right)\mathbf{Y}^{\mathrm{T}}\right) + \frac{1}{2}\left[\left(\hat{\mathbf{v}}-\hat{\mathbf{a}}\right)^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}\right]^2$$
$$-\frac{\lambda(t)}{2}\left(\hat{\mathbf{v}}-\hat{\mathbf{a}}\right)^{\mathrm{T}}\mathbf{P}^{-1}\left(\hat{\mathbf{v}}-\hat{\mathbf{a}}\right)$$

$$= -\eta s^2 - \tilde{f}^2 - 2\left(\hat{\mathbf{v}} - \hat{\mathbf{a}}\right)^{\mathrm{T}}\left(s + \tilde{f}\right)\mathbf{Y}^{\mathrm{T}} - \beta\mathcal{N}\|\hat{\mathbf{v}} - \hat{\mathbf{a}}\|^2 + \frac{1}{2}\left(\tilde{\mathbf{v}}^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}\right)^2$$
$$+ \frac{1}{2}\left[\left(\hat{\mathbf{v}} - \hat{\mathbf{a}}\right)^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}\right]^2 - \frac{\lambda(t)}{2}\left(\tilde{\mathbf{v}}^{\mathrm{T}}\mathbf{P}^{-1}\tilde{\mathbf{v}} + \left(\hat{\mathbf{v}} - \hat{\mathbf{a}}\right)^{\mathrm{T}}\mathbf{P}^{-1}\left(\hat{\mathbf{v}} - \hat{\mathbf{a}}\right)\right).$$

Now we use that $\tilde{\mathbf{v}}^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}} = \left(\hat{\mathbf{v}} - \hat{\mathbf{a}}\right)^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}} + \tilde{f}$ to say that $\frac{1}{2}\left(\tilde{\mathbf{v}}^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}\right)^2 = \frac{1}{2}\left[\left(\hat{\mathbf{v}} - \hat{\mathbf{a}}\right)^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}\right]^2 + \left(\hat{\mathbf{v}} - \hat{\mathbf{a}}\right)^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}\tilde{f} + \frac{1}{2}\tilde{f}^2$. Hence,

$$\begin{aligned}
\dot{V} &= -\eta s^2 - \frac{1}{2}\tilde{f}^2 - 2s\left(\hat{\mathbf{v}} - \hat{\mathbf{a}}\right)^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}} - \tilde{f}\left(\hat{\mathbf{v}} - \hat{\mathbf{a}}\right)^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}} - \beta\mathcal{N}\|\hat{\mathbf{v}} - \hat{\mathbf{a}}\|^2 \\
&\quad + \left[\left(\hat{\mathbf{v}} - \hat{\mathbf{a}}\right)^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}\right]^2 - \frac{\lambda(t)}{2}\left(\tilde{\mathbf{v}}^{\mathrm{T}}\mathbf{P}^{-1}\tilde{\mathbf{v}} + \left(\hat{\mathbf{v}} - \hat{\mathbf{a}}\right)^{\mathrm{T}}\mathbf{P}^{-1}\left(\hat{\mathbf{v}} - \hat{\mathbf{a}}\right)\right) \\
&= -\eta s^2 - \frac{1}{2}\tilde{f}^2 - 2s\left(\hat{\mathbf{v}} - \hat{\mathbf{a}}\right)^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}} - \tilde{f}\left(\hat{\mathbf{v}} - \hat{\mathbf{a}}\right)^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}} - \beta\|\hat{\mathbf{v}} - \hat{\mathbf{a}}\|^2 \\
&\quad -\beta\mu\|\mathbf{Y}\|^2\|\hat{\mathbf{v}} - \hat{\mathbf{a}}\|^2 + \left[\left(\hat{\mathbf{v}} - \hat{\mathbf{a}}\right)^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}\right]^2 \\
&\quad - \frac{\lambda(t)}{2}\left(\tilde{\mathbf{v}}^{\mathrm{T}}\mathbf{P}^{-1}\tilde{\mathbf{v}} + \left(\hat{\mathbf{v}} - \hat{\mathbf{a}}\right)^{\mathrm{T}}\mathbf{P}^{-1}\left(\hat{\mathbf{v}} - \hat{\mathbf{a}}\right)\right) \\
&\leq -\eta s^2 - \frac{1}{2}\tilde{f}^2 + 2|s|\|\left(\hat{\mathbf{v}} - \hat{\mathbf{a}}\right)\|\|\mathbf{Y}^{\mathrm{T}}\| + |\tilde{f}|\|\left(\hat{\mathbf{v}} - \hat{\mathbf{a}}\right)\|\|\mathbf{Y}^{\mathrm{T}}\| - \beta\|\hat{\mathbf{v}} \\
&\quad -\hat{\mathbf{a}}\|^2 - (\beta\mu - 1)\|\mathbf{Y}\|^2\|\hat{\mathbf{v}} - \hat{\mathbf{a}}\|^2 \\
&\quad - \frac{\lambda(t)}{2}\left(\tilde{\mathbf{v}}^{\mathrm{T}}\mathbf{P}^{-1}\tilde{\mathbf{v}} + \left(\hat{\mathbf{v}} - \hat{\mathbf{a}}\right)^{\mathrm{T}}\mathbf{P}^{-1}\left(\hat{\mathbf{v}} - \hat{\mathbf{a}}\right)\right) \\
&\leq -\eta\epsilon_1 s^2 - \frac{\epsilon_2}{2}\tilde{f}^2 - \left(\sqrt{(1-\epsilon_1)\eta}|s| - \frac{1}{\sqrt{(1-\epsilon_1)\eta}}\|\hat{\mathbf{v}} - \hat{\mathbf{a}}\|\|\mathbf{Y}\|\right)^2 \\
&\quad - \left(\sqrt{\frac{1-\epsilon_2}{2}}|\tilde{f}| - \frac{1}{2}\sqrt{\frac{2}{1-\epsilon_2}}\|\hat{\mathbf{v}} - \hat{\mathbf{a}}\|\|\mathbf{Y}^{\mathrm{T}}\|\right)^2 \\
&\quad -\beta\|\hat{\mathbf{v}} - \hat{\mathbf{a}}\|^2 - \frac{\lambda(t)}{2}\left(\tilde{\mathbf{v}}^{\mathrm{T}}\mathbf{P}^{-1}\tilde{\mathbf{v}} + \left(\hat{\mathbf{v}} - \hat{\mathbf{a}}\right)^{\mathrm{T}}\mathbf{P}^{-1}\left(\hat{\mathbf{v}} - \hat{\mathbf{a}}\right)\right),
\end{aligned}$$

where $0 < \epsilon_1 < 1$ and $0 < \epsilon_2 < 1$ are both arbitrary and where we have chosen $\mu = \frac{1}{\beta}\left(1 + \frac{1}{\eta(1-\epsilon_1)} + \frac{1}{2(1-\epsilon_2)}\right)$. This shows that s, $\hat{\mathbf{v}}$, and $\hat{\mathbf{a}}$ remain bounded. Because s remains bounded, $\mathbf{x}$ remains bounded. Integrating $\dot{V}$ shows that $s \in \mathcal{L}_2$ and $\tilde{f} \in \mathcal{L}_2$. By local boundedness of $\tilde{f}$ in $\mathbf{x}$ and $\hat{\mathbf{a}}$ uniformly in t, $\tilde{f}$ remains bounded and hence $\dot{s}$ remains bounded. By Barbalat's lemma (lemma 1), $s \to 0$ and $\mathbf{x} \to \mathbf{x}_d$. □

A.12 Proof of Proposition 19.

Proof. Consider the Lyapunov-like function,

$$V = \frac{1}{2}\tilde{\mathbf{a}}^{\mathrm{T}}\mathbf{P}^{-1}\tilde{\mathbf{a}}, \tag{A.7}$$

which has time derivative

$$\begin{aligned}\dot{V} &= -\tilde{f}\boldsymbol{\alpha}^{\mathrm{T}}\tilde{\mathbf{a}} + \frac{1}{2}\left(\tilde{\mathbf{a}}^{\mathrm{T}}\boldsymbol{\alpha}\right)^2 - \frac{\lambda}{2}\tilde{\mathbf{a}}^{\mathrm{T}}\mathbf{P}^{-1}\tilde{\mathbf{a}},\\ &\leq -\frac{1}{D_1}\tilde{f}^2 + \frac{1}{2D_2^2}\tilde{f}^2 = -\left(\frac{1}{D_1} - \frac{1}{2D_2^2}\right)\tilde{f}^2.\end{aligned}$$

For $D_1 < 2D_2^2$, $\dot{V} \leq 0$ and $\tilde{f} \in \mathcal{L}_2$ over the maximal interval of existence of $\mathbf{x}(t)$. Alternatively, using the same Lyapunov function,

$$\dot{V} \leq -\left(\boldsymbol{\alpha}^{\mathrm{T}}\tilde{\mathbf{a}}\right)^2\left(D_2 - \frac{1}{2}\right).$$

For $D_2 > \frac{1}{2}$, $\dot{V} \leq 0$ and $\boldsymbol{\alpha}^{\mathrm{T}}\tilde{\mathbf{a}} \in \mathcal{L}_2$ over the maximal interval of existence of $\mathbf{x}(t)$. By assumption 2, this implies that $\tilde{f} \in \mathcal{L}_2$ over the same interval. Hence, both approaches demonstrate that $\hat{\mathbf{a}}$ remains bounded over the maximal interval of existence of $\mathbf{x}(t)$ and that $\tilde{f} \in \mathcal{L}_2$ over the same interval. Furthermore, these bounds are independent of the length of the interval. By lemma 2, the proposition is proved. □

A.13 Proof of Proposition 20.

Proof. Consider the Lyapunov-like function,

$$V = \frac{1}{2}\left(\tilde{\mathbf{v}}^{\mathrm{T}}\mathbf{P}^{-1}\tilde{\mathbf{v}} + (\hat{\mathbf{a}} - \hat{\mathbf{v}})^{\mathrm{T}}\mathbf{P}^{-1}(\hat{\mathbf{a}} - \hat{\mathbf{v}})\right),$$

which has time derivative

$$\begin{aligned}\dot{V} &= -\tilde{\mathbf{v}}^{\mathrm{T}}\boldsymbol{\alpha}\tilde{f} + \frac{1}{2}\tilde{\mathbf{v}}^{\mathrm{T}}\left(-\lambda\mathbf{P}^{-1} + \boldsymbol{\alpha}\boldsymbol{\alpha}^{\mathrm{T}}\right)\tilde{\mathbf{v}} + (\hat{\mathbf{a}} - \hat{\mathbf{v}})^{\mathrm{T}}\left(\beta\mathcal{N}(\hat{\mathbf{v}} - \hat{\mathbf{a}}) + \boldsymbol{\alpha}\tilde{f}\right)\\ &\quad + \frac{1}{2}(\hat{\mathbf{a}} - \hat{\mathbf{v}})^{\mathrm{T}}\left(-\lambda\mathbf{P}^{-1} + \boldsymbol{\alpha}\boldsymbol{\alpha}^{\mathrm{T}}\right)(\hat{\mathbf{a}} - \hat{\mathbf{v}})\\ &\leq -\left(\tilde{\mathbf{a}}^{\mathrm{T}}\boldsymbol{\alpha}\right)\tilde{f} + \frac{1}{2}\left(\tilde{\mathbf{v}}^{\mathrm{T}}\boldsymbol{\alpha}\right)^2 + \frac{1}{2}\left((\hat{\mathbf{a}} - \hat{\mathbf{v}})^{\mathrm{T}}\boldsymbol{\alpha}\right)^2 - \beta\mathcal{N}\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2 + 2\tilde{f}(\hat{\mathbf{a}} - \hat{\mathbf{v}})^{\mathrm{T}}\boldsymbol{\alpha}\\ &\leq -D_2\left(\boldsymbol{\alpha}^{\mathrm{T}}\tilde{\mathbf{a}}\right)^2 + \frac{1}{2}\left(\tilde{\mathbf{v}}^{\mathrm{T}}\boldsymbol{\alpha}\right)^2 + \frac{1}{2}\left((\hat{\mathbf{a}} - \hat{\mathbf{v}})^{\mathrm{T}}\boldsymbol{\alpha}\right)^2 - \beta\mathcal{N}\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2 + 2|\tilde{f}|\,\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|\,\|\boldsymbol{\alpha}\|.\end{aligned}$$

Now, we use the fact that $\frac{1}{2}\left(\tilde{\mathbf{v}}^{\mathrm{T}}\boldsymbol{\alpha}\right)^2 = \frac{1}{2}\left[(\hat{\mathbf{v}}-\hat{\mathbf{a}})^{\mathrm{T}}\boldsymbol{\alpha}\right]^2 + (\hat{\mathbf{v}}-\hat{\mathbf{a}})^{\mathrm{T}}\boldsymbol{\alpha}\left(\boldsymbol{\alpha}^{\mathrm{T}}\tilde{\mathbf{a}}\right) + \frac{1}{2}\left(\tilde{\mathbf{a}}^{\mathrm{T}}\boldsymbol{\alpha}\right)^2$ to rewrite

$$\begin{aligned}
\dot{V} &\leq -\left(D_2 - \frac{1}{2}\right)\left(\boldsymbol{\alpha}^{\mathrm{T}}\tilde{\mathbf{a}}\right)^2 + \left[(\hat{\mathbf{a}}-\hat{\mathbf{v}})^{\mathrm{T}}\boldsymbol{\alpha}\right]^2 - \beta\mathcal{N}\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\|^2 + \|\hat{\mathbf{v}} \\
&\quad - \hat{\mathbf{a}}\|\|\boldsymbol{\alpha}\||\boldsymbol{\alpha}^{\mathrm{T}}\tilde{\mathbf{a}}|\,(2D_1+1) \\
&\leq -\left(D_2 - \frac{1}{2}\right)\left(\boldsymbol{\alpha}^{\mathrm{T}}\tilde{\mathbf{a}}\right)^2 - \beta\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\|^2 - (\beta\mu - 1)\,\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\|^2\|\boldsymbol{\alpha}\|^2 + \|\hat{\mathbf{v}} \\
&\quad - \hat{\mathbf{a}}\|\|\boldsymbol{\alpha}\||\boldsymbol{\alpha}^{\mathrm{T}}\tilde{\mathbf{a}}|\,(2D_1+1) \\
&\leq -\epsilon\left(D_2 - \frac{1}{2}\right)\left(\boldsymbol{\alpha}^{\mathrm{T}}\tilde{\mathbf{a}}\right)^2 \\
&\quad - \left(\sqrt{(1-\epsilon)\left(D_2 - \frac{1}{2}\right)}|\boldsymbol{\alpha}^{\mathrm{T}}\tilde{\mathbf{a}}| - \frac{2D_1+1}{2\sqrt{(1-\epsilon)\left(D_2-\frac{1}{2}\right)}}\|\hat{\mathbf{v}}-\hat{\mathbf{a}}\|\|\boldsymbol{\alpha}\|\right)^2 \\
&\quad -\beta\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\|^2,
\end{aligned}$$

where $0<\epsilon<1$ is arbitrary and where we have chosen $\mu = \frac{1}{\beta}\left(1 + \frac{(2D_1+1)^2}{(1-\epsilon)(4D_2-2)}\right)$. $\dot{V}$ is clearly negative semidefinite for $D_2 < \frac{1}{2}$, which shows that $\hat{\mathbf{v}}$ and $\hat{\mathbf{a}}$ remain bounded over the maximal interval of existence of $\mathbf{x}(t)$. Integrating $\dot{V}$ shows that $\left(\boldsymbol{\alpha}^{\mathrm{T}}\tilde{\mathbf{a}}\right) \in \mathcal{L}_2$ over this interval, which implies that $\tilde{f} \in \mathcal{L}_2$ over the same interval by assumption 2. Note that the bounds are independent of the length of the interval. By lemma 2, the proposition is proven. □

A.14 Proof of Proposition 21.

Proof. Consider the Lyapunov-like function candidate,

$$V = \frac{1}{2}s^2 + \frac{1}{\gamma}\left(d_\psi\left(\mathbf{a},\hat{\mathbf{v}}\right) + \frac{1}{2}\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\|^2\right).$$

This function has time derivative

$$\begin{aligned}
\dot{V} &= -\eta s^2 + s\mathbf{Y}\tilde{\mathbf{a}} + \frac{1}{\gamma}\left((\hat{\mathbf{v}}-\mathbf{a})^{\mathrm{T}}\nabla^2\psi(\hat{\mathbf{v}})\dot{\hat{\mathbf{v}}} + (\hat{\mathbf{a}}-\hat{\mathbf{v}})^{\mathrm{T}}\left(\dot{\hat{\mathbf{a}}}-\dot{\hat{\mathbf{v}}}\right)\right) \\
&= -\eta s^2 + s\mathbf{Y}\tilde{\mathbf{a}} + (\mathbf{a}-\hat{\mathbf{v}})^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}s + \frac{1}{\gamma}(\hat{\mathbf{a}}-\hat{\mathbf{v}})^{\mathrm{T}}\left(\beta\mathcal{N}(\hat{\mathbf{v}}-\hat{\mathbf{a}}) + \gamma\nabla^2\psi\,(\hat{\mathbf{v}})^{-1}\mathbf{Y}^{\mathrm{T}}s\right) \\
&= -\eta s^2 - \frac{\beta\mathcal{N}}{\gamma}\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\|^2 + (\hat{\mathbf{a}}-\hat{\mathbf{v}})^{\mathrm{T}}\left(\left[\nabla^2\psi(\hat{\mathbf{v}})\right]^{-1} + \mathbf{I}\right)\mathbf{Y}^{\mathrm{T}}s.
\end{aligned}$$

By l-strong convexity of ψ, $\left(\nabla^2\psi(\hat{\mathbf{v}})\right)^{-1} \leq l^{-1}\mathbf{I}$. Hence, using that $\mathcal{N} = 1 + \mu\|\mathbf{Y}\|^2$,

$$\dot{V} \leq -\eta s^2 - \frac{\beta}{\gamma}\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2 - \frac{\beta\mu}{\gamma}\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2\|\mathbf{Y}\|^2 + \left(\frac{l+1}{l}\right)|s|\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|\|\mathbf{Y}\| \tag{A.8}$$

$$\leq -\epsilon\eta s^2 - \left(\sqrt{(1-\epsilon)\eta}|s| - \frac{l+1}{2l\sqrt{(1-\epsilon)\eta}}\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|\|\mathbf{Y}\|\right)^2 - \frac{\beta}{\gamma}\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|^2, \tag{A.9}$$

where $0 < \epsilon < 1$ is arbitrary and we have chosen $\mu = \frac{\gamma(l+1)^2}{4\beta l^2(1-\epsilon)\eta}$. This shows that $\dot{V}$ is negative semidefinite, so that $\hat{\mathbf{a}}$, $\hat{\mathbf{v}}$, and s remain bounded. Because s remains bounded, $\mathbf{x}$ remains bounded. Integrating $\dot{V}$ shows that $s \in \mathcal{L}_2$, so that $s \in \mathcal{L}_2 \cap \mathcal{L}_\infty$. By local boundedness of $\mathbf{Y}$ in $\mathbf{x}$, $\dot{s}$ remains bounded, and hence by Barbalat's lemma (lemma 1) $s \to 0$. Then $\mathbf{x} \to \mathbf{x}_d$ by definition of s. □

A.15 Proof of Proposition 22.

Proof. Consider the Lyapunov-like function candidate,

$$\begin{aligned} V &= \frac{1}{2}s^2 + \frac{1}{\gamma}\left(d_\psi(\mathbf{a}, \hat{\mathbf{v}}) + d_\psi(\hat{\mathbf{v}}, \hat{\mathbf{a}})\right) \\ &= \frac{1}{2}s^2 + \frac{1}{\gamma}\big(\psi(\mathbf{a}) - \psi(\hat{\mathbf{v}}) - \nabla\psi(\hat{\mathbf{v}})^{\mathrm{T}}(\mathbf{a} - \hat{\mathbf{v}}) + \psi(\hat{\mathbf{v}}) - \psi(\hat{\mathbf{a}}) \\ &\quad - \nabla\psi(\hat{\mathbf{a}})^{\mathrm{T}}(\hat{\mathbf{v}} - \hat{\mathbf{a}})\big). \end{aligned}$$

These individual terms satisfy

$$\begin{aligned} \frac{d}{dt}\frac{1}{2}s^2 &= -\eta s^2 + \mathbf{Y}\tilde{\mathbf{a}}s \\ \frac{1}{\gamma}\frac{d}{dt}d_\psi(\mathbf{a}, \hat{\mathbf{v}}) &= -\mathbf{Y}\tilde{\mathbf{a}}s + (\hat{\mathbf{a}} - \hat{\mathbf{v}})^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}s \\ \frac{1}{\gamma}\frac{d}{dt}d_\psi(\hat{\mathbf{v}}, \hat{\mathbf{a}}) &= -\frac{1}{\gamma}(\hat{\mathbf{v}} - \hat{\mathbf{a}})^{\mathrm{T}}\nabla^2\psi(\hat{\mathbf{a}})\dot{\hat{\mathbf{a}}} - s\mathbf{Y}\left(\nabla^2\psi(\hat{\mathbf{v}})\right)^{-1}(\nabla\psi(\hat{\mathbf{v}}) - \nabla\psi(\hat{\mathbf{a}})). \end{aligned}$$

Note that by l-strong convexity and L-smoothness, the second term in the last line above can be bounded as

$$-s\mathbf{Y}\left(\nabla^2\psi(\hat{\mathbf{v}})\right)^{-1}(\nabla\psi(\hat{\mathbf{v}}) - \nabla\psi(\hat{\mathbf{a}})) \leq \frac{\gamma L}{l}|s|\|\mathbf{Y}\|\|\hat{\mathbf{a}} - \hat{\mathbf{v}}\|.$$

First consider equation 7.6. Then, by l-strong convexity of ψ,

$$-\frac{1}{\gamma}(\hat{\mathbf{v}}-\hat{\mathbf{a}})^{\mathrm{T}}\nabla^2\psi(\hat{\mathbf{a}})\dot{\hat{\mathbf{a}}} \leq -\frac{l\beta\mathcal{N}}{\gamma}\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\|^2.$$

For equation 7.8, we obtain

$$-\frac{1}{\gamma}(\hat{\mathbf{v}}-\hat{\mathbf{a}})^{\mathrm{T}}\nabla^2\psi(\hat{\mathbf{a}})\dot{\hat{\mathbf{a}}} \leq -\frac{\beta\mathcal{N}}{\gamma}\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\|^2$$

Hence, for equations 7.5 and 7.6,

$$\dot{V} \leq -\eta s^2 - \frac{l\beta}{\gamma}\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\|^2 - \frac{l\beta\mu}{\gamma}\|\mathbf{Y}\|^2\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\|^2 + \frac{l+\gamma L}{l}|s|\|\mathbf{Y}\|\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\|$$

Similarly, for equations 7.7 and 7.8,

$$\dot{V} \leq -\eta s^2 - \frac{\beta}{\gamma}\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\|^2 - \frac{\beta\mu}{\gamma}\|\mathbf{Y}\|^2\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\|^2 + \frac{l+\gamma L}{l}|s|\|\mathbf{Y}\|\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\|.$$

In both cases,

$$\dot{V} \leq -\epsilon\eta s^2 - \left(\sqrt{(1-\epsilon)\eta}|s| - \frac{l+\gamma L}{2l\sqrt{(1-\epsilon)\eta}}\|\mathbf{Y}\|\|\hat{\mathbf{a}}-\hat{\mathbf{v}}\|\right)^2.$$

In the former case, we have chosen $\mu = \frac{\gamma(l+\gamma L)^2}{4\beta\eta(1-\epsilon)l^3}$ and in the latter we have chosen $\mu = \frac{\gamma(l+\gamma L)^2}{4\beta\eta(1-\epsilon)l^2}$. This shows that $\dot{V}$ is negative semidefinite, so that $\hat{\mathbf{a}}$, $\hat{\mathbf{v}}$, and s remain bounded. Because s remains bounded, $\mathbf{x}$ remains bounded. Integrating $\dot{V}$ shows that $s \in \mathcal{L}_2$, so that $s \in \mathcal{L}_2 \cap \mathcal{L}_\infty$. By local boundedness of $\mathbf{Y}$ in $\mathbf{x}$, $\dot{s}$ remains bounded, and hence by Barbalat's lemma (lemma 1) $s \to 0$. Then $\mathbf{x} \to \mathbf{x}_d$ by definition of s. □

Appendix B: Further Results on Dynamics Prediction for Hamiltonian Systems

We now provide some extensions to the results in section 4.2 by exploiting the structure of separable Hamiltonians. With a separable Hamiltonian, it is natural to estimate the kinetic and potential energies separately,

$$T(\hat{\mathbf{p}}) = \mathbf{Y}_p(\hat{\mathbf{p}})\hat{\mathbf{a}}_p,$$
$$U(\hat{\mathbf{q}}) = \mathbf{Y}_q(\hat{\mathbf{q}})\hat{\mathbf{a}}_q,$$

where $\mathbf{Y}_p$ and $\mathbf{Y}_q$ are row vectors of basis functions for the kinetic and potential energies, respectively. In this case, following the same derivation as in section 4.1, the error dynamics become

$$\dot{\tilde{\mathbf{p}}} = -\nabla_{\hat{\mathbf{q}}}\mathbf{Y}_q(\hat{\mathbf{q}})\tilde{\mathbf{a}}_q - k_p\tilde{\mathbf{p}} - \left(\nabla_{\hat{\mathbf{q}}}U(\hat{\mathbf{q}}) - \nabla_{\mathbf{q}}U(\mathbf{q})\right),$$
$$\dot{\tilde{\mathbf{q}}} = \nabla_{\hat{\mathbf{p}}}\mathbf{Y}_p(\hat{\mathbf{p}})\tilde{\mathbf{a}}_p - k_q\tilde{\mathbf{q}} + \left(\nabla_{\hat{\mathbf{p}}}T(\hat{\mathbf{p}}) - \nabla_{\mathbf{p}}T(\mathbf{p})\right).$$

Consider the adaptation laws,

$$\dot{\hat{\mathbf{a}}}_p = -\gamma_p\left[\nabla^2\psi_p(\hat{\mathbf{a}}_p)\right]^{-1}\left(\nabla_{\hat{\mathbf{p}}}\mathbf{Y}_p(\hat{\mathbf{p}})\right)^{\mathrm{T}}\tilde{\mathbf{q}},$$
$$\dot{\hat{\mathbf{a}}}_q = \gamma_q\left[\nabla^2\psi_q(\hat{\mathbf{a}}_q)\right]^{-1}\left(\nabla_{\hat{\mathbf{q}}}\mathbf{Y}_q(\hat{\mathbf{q}})\right)^{\mathrm{T}}\tilde{\mathbf{p}},$$

where $\psi_p(\cdot)$ and $\psi_q(\cdot)$ are strongly convex functions, and where $\gamma_p > 0$ and $\gamma_q > 0$ are positive learning rates. The Lyapunov-like function

$$V = \frac{1}{2}\tilde{\mathbf{p}}^{\mathrm{T}}\tilde{\mathbf{p}} + \frac{1}{2}\tilde{\mathbf{q}}^{\mathrm{T}}\tilde{\mathbf{q}} + \frac{1}{\gamma_p}d_{\psi_p}\left(\mathbf{a}_p \,\|\, \hat{\mathbf{a}}_p\right) + \frac{1}{\gamma_q}d_{\psi_q}\left(\mathbf{a}_q \,\|\, \hat{\mathbf{a}}_q\right) \tag{B.1}$$

shows that a sufficient condition for convergence $\tilde{\mathbf{p}} \to 0$ and $\tilde{\mathbf{q}} \to 0$ is for the Jacobian

$$\mathbf{J} = \begin{pmatrix} -k_p\mathbf{I} & -\nabla^2_{\mathbf{q}}U(\mathbf{q}) \\ \nabla^2_{\mathbf{p}}T(\mathbf{p}) & -k_q\mathbf{I} \end{pmatrix}$$

to be uniformly negative definite. A sufficient condition for uniform negative definiteness is given by equation 4.5.

While separable Hamiltonians encompass many physical systems, some, such as robotic systems, do not have this structure. A more general form encompassing robotic systems is

$$\mathcal{H}(\mathbf{p}, \mathbf{q}) = T(\mathbf{p}, \mathbf{q}) + U(\mathbf{q}).$$

Parameterizing these terms independently,

$$T(\hat{\mathbf{p}}, \hat{\mathbf{q}}) = \mathbf{Y}_p(\hat{\mathbf{p}}, \hat{\mathbf{q}})\hat{\mathbf{a}}_p,$$
$$U(\hat{\mathbf{q}}) = \mathbf{Y}_q(\hat{\mathbf{q}})\hat{\mathbf{a}}_q,$$

the error dynamics becomes

$$\dot{\tilde{\mathbf{p}}} = -\left(\nabla_{\hat{\mathbf{q}}}\mathbf{Y}_p(\hat{\mathbf{p}}, \hat{\mathbf{q}})\right)\tilde{\mathbf{a}}_p - \left(\nabla_{\hat{\mathbf{q}}}\mathbf{Y}_q(\hat{\mathbf{q}})\right)\tilde{\mathbf{a}}_q - k_p\tilde{\mathbf{p}} - \left(\nabla_{\hat{\mathbf{q}}}U(\hat{\mathbf{q}}) - \nabla_{\mathbf{q}}U(\mathbf{q})\right)$$
$$- \left(\nabla_{\hat{\mathbf{q}}}T(\hat{\mathbf{p}}, \hat{\mathbf{q}}) - \nabla_{\mathbf{q}}T(\mathbf{p}, \mathbf{q})\right),$$
$$\dot{\tilde{\mathbf{q}}} = \left(\nabla_{\hat{\mathbf{p}}}\mathbf{Y}_p(\hat{\mathbf{p}}, \hat{\mathbf{q}})\right)\tilde{\mathbf{a}}_p - k_q\tilde{\mathbf{q}} + \left(\nabla_{\hat{\mathbf{p}}}T(\hat{\mathbf{p}}, \hat{\mathbf{q}}) - \nabla_{\mathbf{p}}T(\mathbf{p}, \mathbf{q})\right).$$

Now consider the adaptation laws,

$$\dot{\hat{\mathbf{a}}}_p = \gamma_p \left[\nabla^2 \psi_p(\hat{\mathbf{a}}_p)\right]^{-1} \left(\left(\nabla_{\hat{\mathbf{q}}} \mathbf{Y}_p(\hat{\mathbf{p}}, \hat{\mathbf{q}})\right)^{\mathrm{T}} \tilde{\mathbf{p}} - \left(\nabla_{\hat{\mathbf{p}}} \mathbf{Y}_p(\hat{\mathbf{p}}, \hat{\mathbf{q}})\right)^{\mathrm{T}} \tilde{\mathbf{q}} \right),$$

$$\dot{\hat{\mathbf{a}}}_q = \gamma_q \left[\nabla^2 \psi_q(\hat{\mathbf{a}}_q)\right]^{-1} \left(\nabla_{\hat{\mathbf{q}}} \mathbf{Y}_q(\hat{\mathbf{q}})\right)^{\mathrm{T}} \tilde{\mathbf{p}},$$

again where $\psi_p(\cdot)$ and $\psi_q(\cdot)$ are strongly convex functions and $\gamma_p > 0$ and $\gamma_q > 0$ are positive learning rates. The Lyapunov-like function, equation B.1, shows that a sufficient condition for convergence is for the Jacobian matrix,

$$\mathbf{J} = \begin{pmatrix} -k_p \mathbf{I} - \nabla_{\mathbf{p}} \nabla_{\mathbf{q}} T(\mathbf{p}, \mathbf{q}) & -\nabla^2_{\mathbf{q}} U(\mathbf{q}) - \nabla^2_{\mathbf{q}} T(\mathbf{p}, \mathbf{q}) \\ \nabla^2_{\mathbf{p}} T(\mathbf{p}, \mathbf{q}) & -k_q \mathbf{I} + \nabla_{\mathbf{q}} \nabla_{\mathbf{p}} T(\mathbf{p}, \mathbf{q}) \end{pmatrix},$$

to be uniformly negative definite. Sufficient conditions for this are now given by

$$k_p > -\frac{1}{2} \lambda_{\min} \left(\nabla_{\hat{\mathbf{p}}} \nabla_{\hat{\mathbf{q}}} T(\hat{\mathbf{p}}, \hat{\mathbf{q}}) + \nabla_{\hat{\mathbf{q}}} \nabla_{\hat{\mathbf{p}}} T(\hat{\mathbf{p}}, \hat{\mathbf{q}}) \right),$$

$$k_q > \frac{1}{2} \lambda_{\max} \left(\nabla_{\hat{\mathbf{p}}} \nabla_{\hat{\mathbf{q}}} T(\hat{\mathbf{p}}, \hat{\mathbf{q}}) + \nabla_{\hat{\mathbf{q}}} \nabla_{\hat{\mathbf{p}}} T(\hat{\mathbf{p}}, \hat{\mathbf{q}}) \right),$$

$$\lambda_p \lambda_q > \frac{1}{4} \lambda^2_{\max} \left[\nabla^2_{\hat{\mathbf{p}}} T(\hat{\mathbf{p}}, \hat{\mathbf{q}}) - \nabla^2_{\hat{\mathbf{q}}} T(\hat{\mathbf{p}}, \hat{\mathbf{q}}) - \nabla^2_{\hat{\mathbf{q}}} U(\hat{\mathbf{q}}) \right],$$

similar to the fully general case handled in section 4.2. More general results can be obtained by using a non-Euclidean metric as a replacement for the momentum and position estimation error terms in equation B.1.

B.1 Implicit Regularization for Higher-Order Laws. For simplicity, we only consider the linearly parameterized setting. The nonlinearly parameterized setting can be handled immediately.

Proposition 25. *Consider the natural gradient-like higher-order adaptation law for a linearly parameterized dynamics,*

$$\dot{\hat{\mathbf{a}}} = \beta \mathcal{N}(\hat{\mathbf{v}} - \hat{\mathbf{a}})$$

$$\dot{\hat{\mathbf{v}}} = -\left[\nabla^2 \psi(\hat{\mathbf{v}})\right]^{-1} \mathbf{Y}^{\mathrm{T}} s,$$

where $\psi(\cdot)$ is a strongly convex function. Assume that $\hat{\mathbf{a}}(t) \to \hat{\mathbf{a}}_\infty \in \mathcal{A}$ where $\mathcal{A}$ is defined in equation 3.1. Then

$$\hat{\mathbf{a}}_\infty = \arg\min_{\boldsymbol{\theta} \in \mathcal{A}} d_\psi(\boldsymbol{\theta} \,\|\, \hat{\mathbf{v}}(0)).$$

In particular, if $\hat{\mathbf{v}}(0) = \arg\min_{\boldsymbol{\theta} \in \mathbb{R}^p} \psi(\boldsymbol{\theta})$, then

$$\hat{\mathbf{a}}_\infty = \arg\min_{\boldsymbol{\theta} \in \mathcal{A}} \psi(\boldsymbol{\theta}).$$

Proof. First note that if $\hat{\mathbf{a}} \to \hat{\mathbf{a}}_\infty$, then $\hat{\mathbf{v}} \to \hat{\mathbf{a}}_\infty$. Now let $\boldsymbol{\theta}$ be any constant vector of parameters. The Bregman divergence has time derivative

$$\frac{d}{dt} d_\psi(\boldsymbol{\theta} \,\|\, \hat{\mathbf{v}}) = -\left(\frac{d}{dt}\nabla\psi(\hat{\mathbf{a}})\right)^{\mathrm{T}} (\boldsymbol{\theta} - \hat{\mathbf{v}}).$$

Using that $\frac{d}{dt}\nabla\psi(\hat{\mathbf{a}}) = -\mathbf{Y}^{\mathrm{T}} s$ and integrating both sides of the above shows

$$d_\psi(\boldsymbol{\theta} \,\|\, \hat{\mathbf{v}}(0)) = d_\psi(\boldsymbol{\theta} \,\|\, \hat{\mathbf{a}}_\infty) + \int_0^\infty s(\tau)\mathbf{Y}(\mathbf{x}(\tau), \tau)\,(\hat{\mathbf{v}}(\tau) - \boldsymbol{\theta})\, d\tau.$$

Taking $\boldsymbol{\theta} \in \mathcal{A}$, the integral becomes independent of $\boldsymbol{\theta}$. The proof from here is identical to the first-order case. □

References

Ai-Poh Loh, Annaswamy, A. M., & Skantze, F. P. (1999). Adaptation in the presence of a general nonlinear parameterization: An error model approach. *IEEE Transactions on Automatic Control*, *44*(9), 1634–1652. doi:10.1109/9.788531

Alemi, A., Machens, C., Deneve, S., & Slotine, J.-J. (2018). Learning nonlinear dynamics in efficient, balanced spiking networks using local plasticity rules. In *Proceedings of the AAAI Conference on Artificial Intelligence*. Palo Alto, CA: AAAI.

Amari, S.-I. (1998). Natural gradient works efficiently in learning. *Neural Computation*, *10*(2), 251–276. doi:https://doi.org/10.1162/089976698300017746

Andrievskii, B. R., Stotskii, A. A., & Fradkov, A. L. (1988). Velocity-gradient algorithms in control and adaptation problems. *Automation and Remote Control*, *49*, 1533–1564.

Annaswamy, A. M., Skantze, F. P., & Loh, A.-P. (1998). Adaptive control of continuous time systems with convex/concave parameterization. *Automatica*, *34*(1), 33–49. doi:https://doi.org/10.1016/S0005-1098(97)00159-3

Astolfi, A., & Ortega, R. (2003). Immersion and invariance: A new tool for stabilization and adaptive control of nonlinear systems. *IEEE Transactions on Automatic Control*, *48*(4), 590–606.

Azizan, N., & Hassibi, B. (2019). Stochastic gradient/mirror descent: Minimax optimality and implicit regularization. In *Proceedings of the International Conference on Learning Representations*. OpenReview.

Azizan, N., Lale, S., & Hassibi, B. (2019). *Stochastic mirror descent on overparameterized nonlinear models: Convergence, implicit regularization, and generalization*. arXiv:1906.03830.

Bartlett, P. L., Long, P. M., Lugosi, G., & Tsigler, A. (2020). Benign overfitting in linear regression. In *Proceedings of the national Academy of Sciences, U.S.A.*, *117*, 30063–30070.

Beck, A., & Teboulle, M. (2003). Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, *31*(3), 167–175. doi:https://doi.org/10.1016/S0167-6377(02)00231-6

Belkin, M., Hsu, D., Ma, S., & Mandal, S. (2019). Reconciling modern machine-learning practice and the classical bias-variance trade-off. In *Proceedings of the National Academy of Sciences*, *116*(32), 15849–15854. doi:10.1073/pnas.1903070116

Betancourt, M., Jordan, M. I., & Wilson, A. C. (2018). *On symplectic optimization*. arXiv:1802.03653.

Boffi, N. M., & Slotine, J.-J. E. (2020). A continuous-time analysis of distributed stochastic gradient. *Neural Computation*, *32*(1), 36–96. doi:https://doi.org/10.1162/neco_a_01248

Bregman, L. (1967). The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, *7*(3), 200–217. doi:https://doi.org/10.1016/0041-5553(67)90040-7

Champion, K., Lusch, B., Kutz, J. N., & Brunton, S. L. (2019). Data-driven discovery of coordinates and governing equations. In *Proceedings of the National Academy of Sciences*, *116*(45), 22445–22451. https://www.pnas.org/content/116/45/22445. doi:10.1073/pnas.1906995116

Chen, Y., Paiton, D., & Olshausen, B. (2018). The sparse manifold transform. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in neural information processing systems*, *31* (pp. 10513–10524). Red Hook, NY: Curran.

Chen, Z., Zhang, J., Arjovsky, M., & Bottou, L. (2020). Symplectic recurrent neural networks. In *Proceedings of the International Conference on Learning Repmresentations*. OpenReview.

Diakonikolas, J., & Jordan, M. I. (2019). *Generalized momentum-based methods: A Hamiltonian perspective*. arXiv:1906.00436.

Feynman, R. P., Leighton, R. B., & Sands, M. (1977). *The Feynman lectures on physics* (Vol. 2). Reading, MA: Addison-Wesley.

Foster, D. J., Rakhlin, A., & Sarkar, T. (2020). *Learning nonlinear dynamical systems from a single trajectory*. arXiv:2004.14681.

Fradkov, A. L. (1980). Speed-gradient scheme and its application in adaptive control problems. *Automation and Remote Control*, *40*, 1333–1342.

Fradkov, A. L. (1986). Integrodifferentiating velocity gradient algorithms. *Sov. Phys. Dokl.*, *31*, 97–98.

Fradkov, A. L., Miroshnik, I. V., & Nikiforov, V. O. (1999). *Nonlinear and adaptive control of complex systems*. Berlin: Springer.

França, G., Sulam, J., Robinson, D. P., & Vidal, R. (2019). *Conformal symplectic and relativistic optimization*. arXiv:1903.04100.

Gaudio, J. E., Annaswamy, A. M., Bolender, M. A., Lavretsky, E., & Gibson, T. E. (2021). A class of high order tuners for adaptive systems. *IEEE Control Systems Letters*, *5*(2), 391–396.

Gaudio, J. E., Gibson, T. E., Annaswamy, A. M., & Bolender, M. A. (2019). *Provably correct learning algorithms in the presence of time-varying features using a variational perspective*. arXiv:1903.04666.

Gentile, C. (2003). The robustness of the *p*-norm algorithms. *Machine Learning*, *53*(3), 265–299. doi:https://doi.org/10.1023/A:1026319107706

Gilra, A., & Gerstner, W. (2017). Predicting non-linear dynamics by stable local learning in a recurrent spiking neural network. *eLife*, *6*, e28295.

Goel, S., & Klivans, A. (2017). *Learning neural networks with two nonlinear layers in polynomial time.* arXiv:1709.06010.

Goel, S., Klivans, A., & Meka, R. (2018). *Learning one convolutional layer with overlapping patches.* arXiv:1802.02547.

Gunasekar, S., Lee, J., Soudry, D., & Srebro, N. (2018a). *Characterizing implicit bias in terms of optimization geometry.* arXiv:1802.08246.

Gunasekar, S., Lee, J. D., Soudry, D., & Srebro, N. (2018b). Implicit bias of gradient descent on linear convolutional networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in neural information processing systems*, *31* (pp. 9461–9471). Red Hook, NY: Curran.

Hazan, E. (2016). Introduction to online convex optimization. *Foundations and Trends in Optimization*, *2*(3–4), 157–325. doi:10.1561/2400000013

Ioannou, P. A., & Sun, J. (2012). *Robust adaptive control*. New York: Dover.

Kakade, S., Kalai, A. T., Kanade, V., & Shamir, O. (2011). *Efficient learning of generalized linear and single index models with isotonic regression.* arXiv:1104.2018.

Karagiannis, D., Sassano, M., & Astolfi, A. (2009). Dynamic scaling and observer design with application to adaptive control. *Automatica*, *45*(12), 2883–2889. doi:https://doi.org/10.1016/j.automatica.2009.09.013

Kojić, A., & Annaswamy, A. M. (2002). Adaptive control of nonlinearly parameterized systems with a triangular structure. *Automatica*, *38*(1), 115–123. doi:https://doi.org/10.1016/S0005-1098(01)00173-X

Krichene, W., Bayen, A., & Bartlett, P. L. (2015). Accelerated mirror descent in continuous and discrete time. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, & R. Garnett (Eds.), *Advances in neural information processing systems*, *28* (pp. 2845–2853). Red Hook, NY: Curran.

Lee, T., Kwon, J., & Park, F. C. (2018). A natural adaptive control law for robot manipulators. In *Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 1–9). Piscataway, NJ: IEEE.

Lee, T., Wensing, P. M., & Park, F. C. (2020). Geometric robot dynamic identification: A convex programming approach. *IEEE Transactions on Robotics*, *36*, 348–365. doi:10.1109/TRO.2019.2926491

Liu, X., Ortega, R., Su, H., & Chu, J. (2010). Immersion and invariance adaptive control of nonlinearly parameterized nonlinear systems. *IEEE Transactions on Automatic Control*, *55*(9), 2209–2214.

Liu, Y.-Y., Slotine, J.-J., & Barabási, A.-L. (2013). Observability of complex systems. In *Proceedings of the National Academy of Sciences*, *110*(7), 2460–2465. https://www.pnas.org/content/110/7/2460. doi:10.1073/pnas.1215508110

Lohmiller, W., & Slotine, J.-J. E. (1998). On contraction analysis for non-linear systems. *Automatica*, *34*(6), 683–696. doi:http://dx.doi.org/10.1016/S0005-1098(98)00019-3

Lopez, B. T., & Slotine, J.-J. E. (2021). Adaptive nonlinear control with contraction metrics. *IEEE Control Systems Letters*, *5*(1), 205–210. doi:10.1109/LCSYS.2020.3000190.

Luenberger, D. G. (1979). *Introduction to dynamic systems*. New York: Wiley.

Maddison, C. J., Paulin, D., Teh, Y. W., O'Donoghue, B., & Doucet, A. (2018). *Hamiltonian descent methods*. arXiv:1809.05042.

Morse, A. S. (1992). High-order parameter tuners for the adaptive control of linear and nonlinear systems. In A. Isidori & T.-J. Tarn (Eds.), *Systems, models and feedback* (pp. 339–364). Boston: Birkhäuser.

Muehlebach, M., & Jordan, M. I. (2019). *A dynamical systems perspective on Nesterov acceleration.* arXiv:1905.07436.

Muehlebach, M., & Jordan, M. I. (2020). *Optimization with momentum: Dynamical, control-theoretic, and symplectic perspectives.* arXiv:2002.12493.

Muthukumar, V., Vodrahalli, K., & Sahai, A. (2019). Harmless interpolation of noisy data in regression. In *Proceedings of the 2019 IEEE International Symposium on Information Theory* (pp. 2299–2303). Piscataway, NJ: IEEE. doi:10.1109/ISIT.2019.8849614

Narendra, K. S., & Annaswamy, A. M. (2005). *Stable adaptive systems*. New York: Dover.

Nemirovski, A., & Yudin, D. (1983). *Problem complexity and method efficiency in optimization*. New York: Wiley.

Nesterov, Y. (1983). A method for solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, *26*, 367–372.

Ortega, R., Gromov, V., Nuño, E., Pyrkin, A., & Romero, J. G. (2019). *Parameter estimation of nonlinearly parameterized regressions without overparameterization nor persistent excitation: Application to system identification and adaptive control*. arXiv:1910.08016.

Polyak, B. T. (1963). Gradient methods for minimizing functionals. *U.S.S.R. Comput. Math. Math. Phys.*, *3*, 643–653. (in Russian)

Polyak, B. (1964). Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, *4*(5), 1–17. doi:https://doi.org/10.1016/0041-5553(64)90137-5

Polyak, B. T., & Juditsky, A. B. (1992). Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, *30*(4), 838–855. doi:https://doi.org/10.1137/0330046

Sanner, R. M., & Slotine, J. E. (1992). Gaussian networks for direct adaptive control. *IEEE Transactions on Neural Networks*, *3*(6), 837–863. doi:10.1109/72.165588

Sanner, R. M., & Slotine, J.-J. E. (1995). Stable adaptive control of robot manipulators using "neural" networks. *Neural Computation*, *7*(4), 753–790. https://doi.org/10.1162/neco.1995.7.4.753

Shi, B., Du, S. S., Su, W. J., & Jordan, M. I. (2019). *Acceleration via symplectic discretization of high-resolution differential equations*. arXiv:1902.03694.

Slotine, J.-J. E. (2003). Modular stability tools for distributed computation and control. *International Journal of Adaptive Control and Signal Processing*, *17*(6), 397–416. doi:10.1002/acs.754

Slotine, J.-J., & Coetsee, J. (1986). Adaptive sliding controller synthesis for nonlinear systems. *International Journal of Control*, *43*(6), 1631–1651. doi:https://doi.org/10.1080/00207178608933564

Slotine, J.-J. E., & Li, W. (1987). On the adaptive control of robot manipulators. *International Journal of Robotics Research*, *6*(3), 49–59. doi:https://doi.org/10.1177/027836498700600303

Slotine, J.-J., & Li, W. (1991). *Applied nonlinear control*. Upper Saddle River, NJ: Prentice Hall.

Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., & Srebro, N. (2018). The implicit bias of gradient descent on separable data. *J. Mach. Learn. Res.*, *19*(1), 2822–2878.

Su, W., Boyd, S., & Candès, E. J. (2016). A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights. *Journal of Machine Learning Research*, *17*(153), 1–43.

Sussillo, D., & Abbott, L. F. (2009). Generating coherent patterns of activity from chaotic neural networks. *Neuron*, *63*(4), 544–557.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267–288. doi:10.1111/j.2517-6161.1996.tb02080.x

Tyukin, I. Y. (2003). Adaptation algorithms in finite form for nonlinear dynamic objects. *Automation and Remote Control*, *64*(6), 951–974. doi:https://doi.org/10.1023/A:1024141700331

Tyukin, I. (2011). *Adaptation in dynamical systems*. Cambridge: Cambridge University Press.

Tyukin, I. Y., Prokhorov, D. V., & van Leeuwen, C. (2007). Adaptation and parameter estimation in systems with unstable target dynamics and nonlinear parameterization. *IEEE Transactions on Automatic Control*, *52*(9), 1543–1559. doi:10.1109/TAC.2007.904448

Wensing, P. M., Kim, S., & Slotine, J. E. (2018). Linear matrix inequalities for physically consistent inertial parameter identification: A statistical perspective on the mass distribution. *IEEE Robotics and Automation Letters*, *3*(1), 60–67. doi:10.1109/LRA.2017.2729659

Wensing, P. M., & Slotine, J. (2018). Cooperative adaptive control for cloud-based robotics. In *Proceedings of the 2018 IEEE International Conference on Robotics and Automation* (pp. 6401–6408). Piscataway, N: IEEE. doi:10.1109/ICRA.2018.8460856

Wensing, P. M., & Slotine, J.-J. E. (2020). Beyond convexity: Contraction and global convergence of gradient descent. *PLOS One*, *15*(8), e0236661

Wibisono, A., Wilson, A. C., & Jordan, M. I. (2016). A variational perspective on accelerated methods in optimization. In *Proceedings of the National Academy of Sciences*, *113*(47), E7351–E7358. doi:10.1073/pnas.1614734113

Wilson, A. C., Recht, B., & Jordan, M. I. (2016). *A Lyapunov analysis of momentum methods in optimization.* arXiv:1611.02635.

Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2016). *Understanding deep learning requires rethinking generalization.* arXiv:1611.03530.

Zhang, S., Choromanska, A., & LeCun, Y. (2014). *Deep learning with elastic averaging SGD*. arXiv:1412.6651.

Received February 23, 2020; accepted September 28, 2020.