Provably Correct Learning Algorithms in the Presence of Time-Varying Features Using a Variational Perspective

Joseph E. Gaudio [1] **Travis E. Gibson** [2] **Anuradha M. Annaswamy** [1] **Michael A. Bolender** [3]

Abstract

Features in machine learning problems are often time-varying and may be related to outputs in an algebraic or dynamical manner. The dynamic nature of these machine learning problems renders current higher order accelerated gradient descent methods unstable or weakens their convergence guarantees. Inspired by methods employed in adaptive control, this paper proposes new algorithms for the case when time-varying features are present, and demonstrates provable performance guarantees. In particular, we develop a unified variational perspective within a continuous time algorithm. This variational perspective includes higher order learning concepts and normalization, both of which stem from adaptive control, and allows stability to be established for dynamical machine learning problems where time-varying features are present. These higher order algorithms are also examined for provably correct learning in adaptive control and identification. This version is an extended abstract; refer to (Gaudio et al., 2019) for the full paper.

1. Introduction

As a field, machine learning has focused on both the processes by which computer systems automatically improve through experience, and on the underlying principles that govern learning systems (Duda et al., 2001; Bishop, 2006; Jordan & Mitchell, 2015; Goodfellow et al., 2016). A particularly useful approach for accomplishing this process of automatic improvement is to embody learning in the form of approximating a desired function and to employ optimization theory to reduce an approximation error as more data is observed. The field of adaptive control, on the other hand, has focused on the process of controlling engineering systems in order to accomplish regulation and tracking of critical variables of interest (e.g. speed in automotive systems, position and force in robotics, Mach number and altitude in aerospace systems, frequency and voltage in power systems) in the presence of uncertainties in the underlying system models, changes in the environment, and unforeseen variations in the overall infrastructure (Sastry & Bodson, 1989; Åström & Wittenmark, 1995; Ioannou & Sun, 1996; Narendra & Annaswamy, 2005). The approach used for accomplishing such regulation and tracking is to learn the underlying parameters through an online estimation algorithm. Stability theory is employed for enabling guarantees for the safe evolution of the critical variables, and convergence of the regulation and tracking errors to zero. In both machine learning and adaptive control the core algorithm is often inspired by gradient descent or gradient flow (Narendra & Annaswamy, 2005). As the scope of problems in both fields increases, the associated complexity and challenges increase as well, necessitating a better understanding of how the underlying algorithms can be designed to enhance learning and stability for dynamical, real-time learning problems.

Modifications to standard gradient descent have been actively researched within the optimization community since computing began. The seminal accelerated gradient method proposed by (Nesterov, 1983) has not only received significant attention in the optimization community (Nesterov, 2004; Beck & Teboulle, 2009; Bubeck, 2015; Carmon et al., 2018), but also in the neural network learning community (Krizhevsky et al., 2012; Sutskever et al., 2013). Nesterov's original method, or a variant (Duchi et al., 2011; Kingma & Ba, 2017; Wilson et al., 2017) are the standard methods for training deep neural networks. To gain insight into Nesterov's method, which is a difference equation, reference (Su et al., 2016) identified the second order ordinary differential equation (ODE) at the limit of zero step size. Still pushing further in the continuous time analysis of these higher order methods, several recent results have leveraged a variational approach showing that, at least in continuous time, there exists a broad class of higher order methods where one can obtain an arbitrarily fast convergence rate (Wibisono et al.,

[1]Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA [2]Harvard Medical School, Boston, MA, USA [3]Air Force Research Laboratory, WPAFB, OH, USA. Correspondence to: Joseph E. Gaudio <jegaudio@mit.edu>.

Real-world Sequential Decision Making workshop at ICML 2019.

2016; Wilson et al., 2016). Converting back to discrete time to obtain an implementable algorithm with rates matching that of the differential equation is also an active area of research (Betancourt et al., 2018; Wilson, 2018). It should be noted that in all the aforementioned work, while the parameter update is time-varying, the features and output of the cost function are static.

The adaptive control community has also analyzed several modifications to gradient descent over the past 40 years. These modifications have been introduced to ensure provably safe learning in the presence of both structured parametric uncertainty and unstructured uncertainty due to unmodeled dynamics, magnitude saturation, delays, and disturbances (Ioannou & Kokotovic, 1984; Karason & Annaswamy, 1994; Bekiaris-Liberis & Krstic, 2010; Gibson et al., 2013). A majority of these modifications are applied to first order gradient-like updates. One notable exception is the "high-order tuner" proposed by (Morse, 1992) which has been useful in providing stable algorithms for time-delay systems (Evesque et al., 2003).

In this paper, we consider a general class of learning problems where features (regressors) are time-varying. In comparison, most higher order methods in the machine learning literature are analyzed for the case where features are assumed to be constant (Wibisono et al., 2016). References (Hopfield, 1982; 1984; Jordan, 1986; Hochreiter & Schmidhuber, 1997; Dietterich, 2002; Jordan & Mitchell, 2015; Kuznetsov & Mohri, 2015; Hall & Willett, 2015; Zinkevich, 2003; Hazan et al., 2007; 2008; Hazan, 2016; Shalev-Shwartz, 2011; Raginsky et al., 2010) emphasize the need for tools for problems where either the input features are time-varying, for time-series prediction, for recurrent networks with time-varying inputs, for sequential performance, or for online optimization. Capability to explicitly handle time-varying features sequentially processed online is essential as machine learning algorithms begin to be used in real-time safety critical applications. Utilizing a common variational perspective, inspired by (Wibisono et al., 2016), this paper will aim to realize two objectives. The first objective is the derivation of a provably correct higher order learning algorithm for regression problems with time-varying features. The second objective of this paper is to derive a provably correct higher order online learning algorithm for uncertain dynamical systems, as often occur in adaptive identification and control problems (Narendra & Annaswamy, 2005). Both objectives are realized in this paper using the notion of a "higher order tuner", first introduced in adaptive control in (Morse, 1992) and formally analyzed within the context of time delay systems in (Evesque et al., 2003). With the variational perspective in (Wibisono et al., 2016), it will be shown that the high-order learning will lead to a provably correct algorithm when time-varying features are present in a machine learning problem, and that it leads to stable learning for a general class of dynamical systems, going beyond the specific problems considered in (Morse, 1992; Evesque et al., 2003). Finally we note that while we focus on models that are linear in the parameters (for ease of exposition and clarity of presentation), nonlinearly parameterized models can also be analyzed using similar Lyapunov stability approaches (Annaswamy & Yu, 1996; Yu & Annaswamy, 1996; 1998; Loh et al., 1999; Cao et al., 2003).

2. Warmup: Time-Varying Features and Model Reference Adaptive Control

2.1. Time-Varying Regression

A time-varying regression system may be expressed as $y(t) = \theta^{*T}\phi(t)$, where $\theta^*, \phi(t) \in \mathbb{R}^N$ represent the unknown constant parameter and the known time-varying feature respectively. The variable $y(t) \in \mathbb{R}$ represents the known time-varying output. Given that θ^* is unknown, we formulate an estimator $\hat{y}(t) = \theta^T(t)\phi(t)$, where $\hat{y}(t) \in \mathbb{R}$ is the estimated output and the unknown parameter is estimated as $\theta(t) \in \mathbb{R}^N$. Define the error between the actual output and the estimated output as

$$e_y(t) = \hat{y}(t) - y(t) = \tilde{\theta}^T(t)\phi(t) \tag{1}$$

where $\tilde{\theta}(t) = \theta(t) - \theta^*$ is the parameter estimation error. An overview of the time-varying regression error model may be seen in Figure 1. The differential equation for the output error is of the form

$$\dot{e}_y(t) = \dot{\theta}^T(t)\phi(t) + \tilde{\theta}^T(t)\dot{\phi}(t). \tag{2}$$

where the time variation of the feature $\dot{\phi}(t)$ can be seen to appear. The goal is to design a rule to adjust the parameter estimate $\theta(t)$ in a continuous manner using knowledge of $\phi(t)$ and $e_y(t)$ such that $e_y(t)$ converges towards zero. A continuous, gradient descent-like algorithm is desired as the output of the regression system $y(t)$ may be corrupted by noise and feature dimensions may be large. To do so, consider the squared loss cost function: $L = \frac{1}{2}e_y^2(t)$. The gradient of this function with respect to the parameters can be expressed as: $\nabla_\theta L = \phi(t)e_y(t)$. The standard gradient flow algorithm (the continuous time limit of gradient descent) may be expressed as follows with user-designed gain parameter $\gamma > 0$ (Narendra & Annaswamy, 2005):

$$\dot{\theta}(t) = -\gamma\nabla_\theta L = -\gamma\phi(t)e_y(t). \tag{3}$$

The parameter error model is then $\dot{\tilde{\theta}}(t) = -\gamma\phi(t)\phi^T(t)\tilde{\theta}(t)$. Stability analysis of the update law in (3) for the error model in (1) is provided in (Gaudio et al., 2019).

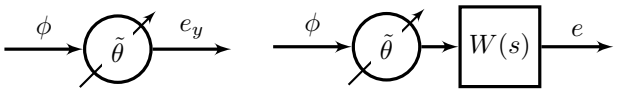

Figure 1: Error models. Left: Time-varying regression (1). Right: Adaptive identification and control (4).

2.2. Adaptive Control and Identification

In the previous subsection, the output was an algebraic combination of the elements of the feature. In a class of problems (including adaptive identification and adaptive control) the features may be related to the errors of a dynamical system. To demonstrate this, features $\phi(t) \in \mathbb{R}^N$ may be related to a measurable state $x(t) \in \mathbb{R}^n$ through a dynamical system with unknown constant parameter $\theta^* \in \mathbb{R}^N$ as $\dot{x}(t) = Ax(t) + b(u(t) + \theta^{*T}\phi(t))$, where $u(t) \in \mathbb{R}$ is an input to the system.[1] A single input system is considered here for notational simplicity, where $A \in \mathbb{R}^{n\times n}$, and $b \in \mathbb{R}^{n\times 1}$ are known stable dynamics and input matrices respectively. It can be noted that the results of this paper extend naturally to multiple input systems. Additionally, it should be noted that it is common in adaptive control for the feature to be a function of the state, i.e., $\phi(t) = \phi(x(t))$. This dynamical system is akin to a linearized recurrent neural network and is similar to the dynamical systems considered in (Hazan et al., 2017; 2018; Recht, 2018; Dean et al., 2018a;b;c). Similar to the linear regression case where an output estimator was created with the same form as the time-varying system, but with an estimate of the unknown parameter, a state estimator with state $\hat{x}(t) \in \mathbb{R}^n$ may be designed for this system as $\dot{\hat{x}}(t) = A\hat{x}(t) + b(u(t) + \theta^T(t)\phi(t))$. Define the error between the considered dynamical system and estimator dynamical system as $e(t) = \hat{x}(t) - x(t)$. The error model for identification and control schemes may then be stated as

$$\dot{e}(t) = Ae(t) + b\tilde{\theta}^T(t)\phi(t) \tag{4}$$

where the relation of the feature to the error can be seen to be through a differential equation, which is fundamentally different from (1). An overview of the dynamical error model may be seen in Figure 1, with transfer function $W(s) := (sI - A)^{-1}b$. Rather than employing a gradient flow based rule, a stability based algorithm may be chosen as follows, with a gain $\gamma > 0$ selected to adjust the learning rate (Narendra & Annaswamy, 2005):

$$\dot{\theta}(t) = -\gamma\phi(t)e^T(t)Pb \tag{5}$$

where $P = P^T \in \mathbb{R}^{n\times n}$ is a positive definite matrix that solves the equation $A^TP + PA = -Q$, where $Q = Q^T \in \mathbb{R}^{n\times n}$ is a user selected positive definite matrix (Narendra & Annaswamy, 2005). Comparing (5) to (3), it can be noticed that the structure is similar with the multiplication of the feature by the error. The difference between them is through the inclusion of elements from the differential equation relating the parameter error to the model tracking error (4). Stability analysis of the update in (5) for the error model in (4) is provided in (Gaudio et al., 2019).[2]

[1] The input is usually designed as $u(t) = -\theta^T(t)\phi(x(t))$.

3. Algorithm Derivation from a Variational Perspective

This section derives higher order update algorithms for both the time-varying regression, as well as the adaptive control and identification problems. For the time-varying regression problem, the goal is to derive a higher order algorithm to adjust the parameter estimate $\theta(t)$ (as compared to (3)), for minimization of the estimation error $e_y(t)$ in the algebraic error model (1). For the adaptive control and identification problem, the goal is to derive a higher order algorithm to adjust $\theta(t)$ (as compared to (5)), such that the error $e(t)$ in the dynamical error model in (4) converges to zero. For the remainder of the paper, the notation of time dependence of variables will be omitted when it is clear from the context. We begin with a common variational perspective in order to derive our higher order algorithms. In particular the Bregman Lagrangian (see (Wibisono et al., 2016), Equation 1) is restated below as

$$\mathcal{L}(\theta, \dot{\theta}, t) = \mathrm{e}^{\bar{\alpha}_t + \bar{\gamma}_t}\left(D_h(\theta + \mathrm{e}^{-\bar{\alpha}_t}\dot{\theta}, \theta) - \mathrm{e}^{\bar{\beta}_t}L(\theta)\right)$$

where D_h is the Bregman divergence defined with a distance-generating function h as: $D_h(y, x) = h(y) - h(x) - \langle \nabla h(x), y - x\rangle$. This Lagrangian can be seen to weight the potential energy (loss) $L(\theta)$ versus kinetic energy $D_h(\theta + \mathrm{e}^{-\bar{\alpha}_t}\dot{\theta}, \theta)$, with a term $\mathrm{e}^{\bar{\alpha}_t + \bar{\gamma}_t}$ which adjusts the damping. The user defined time-varying parameters $(\bar{\alpha}_t, \bar{\beta}_t, \bar{\gamma}_t)$ will be defined in the following section.

3.1. Time-Varying Regression

For ease of exposition, we will use the squared Euclidean norm $h(x) = \frac{1}{2}\|x\|^2$ in the Bregman divergence along with the squared loss $L = \frac{1}{2}e_y^2$ as was used in Section 2.1. The following are our choice of the time-varying scaling parameters: $\bar{\alpha}_t = \ln(\beta\mathcal{N}_t)$, $\bar{\beta}_t = \ln(\gamma/(\beta\mathcal{N}_t))$, and $\bar{\gamma}_t = \int_{t_0}^t \beta\mathcal{N}_\nu d\nu$, where $\gamma, \beta > 0$ are design choices and

$$\mathcal{N}_t \triangleq (1 + \mu\phi^T\phi) \tag{6}$$

with scalar $\mu > 0$ is a function of the time-varying feature, and is referred to as a *normalizing signal*. It can be noticed that the second "ideal scaling condition" (Equation 2b, $\dot{\bar{\gamma}}_t = e^{\bar{\alpha}_t}$) of (Wibisono et al., 2016) holds but the first "ideal

[2] Open loop unstable plants may also be considered in the model tracking problem for a controllable system by choosing $\dot{\hat{x}}(t) = A_m\hat{x}(t) + b(u(t) + \theta^T(t)\phi(t))$, with $A_m \triangleq A - bK$ chosen stable with $K \in \mathbb{R}^{1\times n}$ (Narendra & Annaswamy, 2005).

scaling condition" (Equation 2a, $\dot{\bar{\beta}}_t \leq e^{\bar{\alpha}_t}$) does not need to hold in general. In this sense, the results of this paper are applicable to a larger class of algorithms. With this choice of parameters, distance-generating function and loss function, the following non-autonomous Lagrangian results:

$$\mathcal{L}(\theta, \dot{\theta}, t) = \mathrm{e}^{\int_{t_0}^{t} \beta \mathcal{N}_\nu d\nu} \frac{1}{\beta \mathcal{N}_t} \left(\frac{1}{2} \dot{\theta}^T \dot{\theta} - \gamma \beta \mathcal{N}_t \frac{1}{2} e_y^2 \right). \quad (7)$$

The Lagrangian in equation (7) is the central idea which will produce the first higher order algorithm in this paper. This Lagrangian is a function of not only the parameter and its time derivative, but is also a function of the time-varying feature ϕ directly through normalizing signal $\mathcal{N}_t$. Using the Lagrangian in (7), a functional may be defined as: $J(\theta) = \int_{\mathbb{T}} \mathcal{L}(\theta, \dot{\theta}, t) dt$, where $\mathbb{T}$ is a time interval. To minimize this functional, a necessary condition from the calculus of variations is that the Lagrangian solves the Euler-Lagrange equation (Luenberger, 1969): $\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}}(\theta, \dot{\theta}, t)\right) = \frac{\partial \mathcal{L}}{\partial \theta}(\theta, \dot{\theta}, t)$. The second order differential equation resulting from the application of the Euler-Lagrange equation is:

$$\ddot{\theta} + \left[\beta \mathcal{N}_t - \frac{\dot{\mathcal{N}}_t}{\mathcal{N}_t} \right] \dot{\theta} = -\gamma \beta \mathcal{N}_t \phi e_y. \quad (8)$$

Here β can be seen to adjust "friction". Taking $\beta \to \infty$ (strong friction limit) results in the standard first order algorithm (3).[3] The second order differential equation in (8) may be implemented using two first-order differential equations, similar to (Evesque et al., 2003):

$$\dot{\vartheta} = -\gamma \nabla_\theta L = -\gamma \phi e_y, \qquad \dot{\theta} = -\beta(\theta - \vartheta)\mathcal{N}_t. \quad (9)$$

Equation (9) is the higher order algorithm that represents the first main contribution of this paper. It can be seen that the first of the two equations in (9) is identical to the first-order update (3); the second equation can be viewed as a filter, normalized by the feature-dependent $\mathcal{N}_t$. Alternately, the first equation can be viewed as a gradient step and the second as a mixing step. Similar in form to batch normalization (Ioffe & Szegedy, 2015) and the update in ADAM (Kingma & Ba, 2017), the normalization present in this algorithm is different in that it normalizes by the time-varying feature itself as opposed to estimated moments. In equation (9) it can be seen that $\beta \to \infty$ decreases the nominal time constant (for a given ϕ) of the time-varying filter, and thus in the limit $\beta \to \infty$ the first order algorithm (3) is recovered.[3]

3.2. Adaptive Control and Identification

In a similar manner to (7) the following non-autonomous Lagrangian is defined:

$$\mathcal{L}(\theta, \dot{\theta}, t) = \mathrm{e}^{\int_{t_0}^{t} \beta \mathcal{N}_\nu d\nu} \frac{1}{\beta \mathcal{N}_t} \left(\frac{1}{2} \dot{\theta}^T \dot{\theta} - \gamma \beta \mathcal{N}_t \left[\frac{d}{dt} \left\{ \frac{e^T P e}{2} \right\} + \frac{e^T Q e}{2} \right] \right). \quad (10)$$

[3] This notion will be more rigorously shown in Section 4.

Comparing the Lagrangian in (10) to that in (7), it can be seen that they only differ by the term in the square brackets, representing the loss function considered. The extra terms in the square brackets account for energy storage in the error model dynamics in equation (4). This may be seen as: $\left[\frac{d}{dt} \left\{ \frac{e^T P e}{2} \right\} + \frac{e^T Q e}{2} \right] = e^T P(\dot{e} - Ae) = e^T P b \tilde{\theta}^T \phi$, where the loss is only zero for this dynamical error model when both e and $\dot{e}$ are zero. Using this Lagrangian, a functional may be defined as $J(\theta) = \int_{\mathbb{T}} \mathcal{L}(\theta, \dot{\theta}, t) dt$. The minimization of this functional with the Euler-Lagrange equation and error dynamics (4) results in the following second order differential equation:

$$\ddot{\theta} + \left[\beta \mathcal{N}_t - \frac{\dot{\mathcal{N}}_t}{\mathcal{N}_t} \right] \dot{\theta} = -\gamma \beta \mathcal{N}_t \phi e^T P b. \quad (11)$$

Again, β can be seen to represent "friction", with $\beta \to \infty$ resulting in the first order algorithm (5).[3] The second order differential equation in (11) may be implemented using two first-order differential equations, similar to (Evesque et al., 2003):

$$\dot{\vartheta} = -\gamma \phi e^T P b, \qquad \dot{\theta} = -\beta(\theta - \vartheta)\mathcal{N}_t \quad (12)$$

Equation (12) is the higher order algorithm that represents the second main contribution of this paper. Similar to (9), the first equation may be viewed as the stability based update (5); the second equation may be viewed as a filter, normalized by the feature-dependent $\mathcal{N}_t$.

4. Stability Analysis and Regret Bounds

In this section we state the main stability and convergence results, as well as regret bounds for the algorithms derived in this paper. Unless otherwise specified, $\|\cdot\|$ represents the 2-norm. Stability analysis using Lyapunov functions have been of increased use in recent years in state of the art machine learning approaches (Wilson et al., 2016; Wilson, 2018). A brief overview of Lyapunov functions and their use in stability analysis is presented in (Gaudio et al., 2019).

Theorem 1 (Time-varying regression)**.** *For the second order update law in (9) applied to the time-varying regression problem in (1) the following*

$$V = \frac{1}{\gamma} \|\vartheta - \theta^*\|^2 + \frac{1}{\gamma} \|\theta - \vartheta\|^2 \quad (13)$$

is a Lyapunov function with time derivative $\dot{V} \leq -\frac{2\beta}{\gamma} \|\theta - \vartheta\|^2 - \|e_y\|^2 - \left[\|e_y\| - 2\|\theta - \vartheta\| \|\phi\|\right]^2 \leq 0$ *and therefore* $(\vartheta - \theta^*) \in \mathcal{L}_\infty$ *and* $(\theta - \vartheta) \in \mathcal{L}_\infty$*. If in addition it assumed that* $\phi, \dot{\phi} \in \mathcal{L}_\infty$ *then* $\lim_{t\to\infty} e_y(t) = 0$, $\lim_{t\to\infty}(\theta(t) - \vartheta(t)) = 0$, $\lim_{t\to\infty} \dot{\vartheta}(t) = 0$, *and* $\lim_{t\to\infty} \dot{\theta}(t) = 0$.

Corollary 1. *The second order update law in* (9) *applied to the time-varying regression problem in* (1) *has bounded regret: Regret*$_{\text{continuous}} := \int_0^T \|e_y(\tau)\|^2 d\tau = \mathcal{O}(1)$.

Theorem 2 (Model reference adaptive control)**.** *For the second order update law in (12) applied to the model reference adaptive control problem in (4) the following*

$$V = \frac{1}{\gamma}\|\vartheta - \theta^*\|^2 + \frac{1}{\gamma}\|\theta - \vartheta\|^2 + e^T P e \qquad (14)$$

is a Lyapunov function with time derivative $\dot{V} \leq -\frac{2\beta}{\gamma}\|\theta - \vartheta\|^2 - \|e\|^2 - [\|e\| - 2\|Pb\|\|\theta - \vartheta\|\|\phi\|]^2 \leq 0$ *and therefore* $e \in \mathcal{L}_\infty$, $(\vartheta - \theta^*) \in \mathcal{L}_\infty$, *and* $(\theta - \vartheta) \in \mathcal{L}_\infty$. *If in addition it assumed that* $\phi \in \mathcal{L}_\infty$ *then* $\lim_{t\to\infty} e(t) = 0$. *Also if* $\dot{\phi} \in \mathcal{L}_\infty$, *then* $\lim_{t\to\infty}(\theta(t) - \vartheta(t)) = 0$, $\lim_{t\to\infty} \dot{\vartheta}(t) = 0$, *and* $\lim_{t\to\infty} \ddot{\tilde{\theta}}(t) = 0$.

Corollary 2. *The second order update law in (12) applied to the adaptive control problem in (4) has bounded regret:* $Regret_{\text{continuous}} := \int_0^T \|e(\tau)\|^2 d\tau = \mathcal{O}(1)$.

For proofs of Theorems 1 and 2 see (Gaudio et al., 2019). Corollaries 1 and 2 follow from $\dot{V}(t) \leq -\|e_y(t)\|^2$, $\dot{V}(t) \leq -\|e(t)\|^2$ and V is bounded, as demonstrated in (Gaudio et al., 2019).

5. Comparison of Approaches

Table 1 shows a comparison of the Lagrangian functional and the resulting second order ODE from a given parameterization of the algorithm proposed by (Wibisono et al., 2016) to the results provided in this paper for regression. This parameterization was chosen to coincide with the notions used in this paper (squared loss and the squared Euclidean norm for the error in (1)) along with parameters chosen as in Equation 12 of (Wibisono et al., 2016). It can be seen that both Lagrangians have an increasing function multiplying the kinetic and potential energies with an additional time-varying term weighting the potential energy. Our approach however is a function of the feature ϕ as compared to an explicit function of time. This more natural parameterization results in an algorithm shown for comparison purposes in Table 1 that does not have a damping term that decays to zero with time. Therefore our algorithm does not change from an overdamped to underdamped system as time progresses as is commonly seen in higher order accelerated methods (Su et al., 2016). Thus our approach is capable of running continuously as features are processed. No restart is required as is often used in accelerated algorithms in machine learning (O'Donoghue & Candès, 2013). The more natural damping term shown in our higher order ODE is an explicit function of both the feature and time derivative of the feature vector. It can be noted once more that the time derivative of the feature does not need to be known as this ODE may be implemented as the higher order algorithm in (9), which allows for online processing of the features, without a priori knowledge of its future variation. Therefore the higher order algorithms derived in this paper can be used in real-time sequential decision making systems, where features and output errors are processed online.

Table 1: Comparison of approaches for regression.

Parameterization from (Wibisono et al., 2016)
$\mathcal{L}(\theta, \dot{\theta}, t) = \frac{t^{p+1}}{p}\left(\frac{1}{2}\dot{\theta}^T\dot{\theta} - Cp^2t^{p-2}\frac{1}{2}e_y^2\right)$
$\ddot{\theta} + \frac{p+1}{t}\dot{\theta} = -Cp^2t^{p-2}\phi e_y$
Our Approach
$\mathcal{L}(\theta, \dot{\theta}, t) = \mathrm{e}^{\int_{t_0}^t \beta\mathcal{N}_\nu d\nu}\frac{1}{\beta\mathcal{N}_t}\left(\frac{1}{2}\dot{\theta}^T\dot{\theta} - \gamma\beta\mathcal{N}_t\frac{1}{2}e_y^2\right)$
$\ddot{\theta} + \left[\beta\mathcal{N}_t - \frac{\dot{\mathcal{N}}_t}{\mathcal{N}_t}\right]\dot{\theta} = -\gamma\beta\mathcal{N}_t\phi e_y$

Normalization by the magnitude of the time-varying feature (6) can be seen to be explicitly included in our algorithm. This normalization is in fact necessary in order to provide a proof of stability as was found by (Evesque et al., 2003), due to the required time-varying feature dependent scaling. The candidate Lyapunov function proposed by (Wibisono et al., 2016) applied to our algorithm represents a scaled kinetic plus potential energy, and results in a time derivative that cannot be guaranteed to be non-increasing for arbitrary initial conditions and time variations of the feature as shown in (Gaudio et al., 2019). Our Lyapunov function is fundamentally different in its construction and is indeed able to verify stability. It should be noted that the class of algorithms in (Wibisono et al., 2016) was not designed for time-varying features and that the comparisons are due to its general form in continuous time, representing a large class of higher order learning algorithms commonly used in machine learning, including Nesterov acceleration (Nesterov, 1983). It can also be noted that the higher order algorithms proposed in this paper are *proven stable regardless of the initial condition* of the system (see Section 4). That is to say that an optimization problem-specific schedule on the parameters of the problem is not required to set in order to cope with the initial conditions of the algorithm, as is usually required for momentum methods commonly used in machine learning (Sutskever et al., 2013). Our regret bounds do not increase as a function of time as is common in online machine learning approaches (Zinkevich, 2003; Hazan et al., 2007; 2008; Shalev-Shwartz, 2011; Hazan, 2016). Thus $\mathcal{O}(1)$, constant regret attained by our algorithms is the best achievable regret, up to constants which do not vary with time. The provably correct algorithms proposed in this paper are proven to be stable, with $\mathcal{O}(1)$ regret bounds, and provide for a unified framework using a variational perspective for convergence in output (9) (respectively model tracking (12)) error for *time-varying features with arbitrary initial conditions* where the relation between feature and error may be algebraic (1) or dynamical (4). Numerical experiments of the algorithms in this paper are provided in (Gaudio et al., 2019).

Acknowledgments

This work was supported by the Air Force Research Laboratory, Collaborative Research and Development for Innovative Aerospace Leadership (CRDInAL), Thrust 3 - Control Automation and Mechanization grant FA 8650-16-C-2642 and the Boeing Strategic University Initiative.

References

Annaswamy, A. M. and Yu, S.-H. θ-adaptive neural networks: a new approach to parameter estimation. *IEEE Transactions on Neural Networks*, 7(4):907–918, jul 1996. doi: 10.1109/72.508934.

Åström, K. J. and Wittenmark, B. *Adaptive Control: Second Edition*. Addison-Wesley Publishing Company, 1995.

Beck, A. and Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, jan 2009. doi: 10.1137/080716542.

Bekiaris-Liberis, N. and Krstic, M. Delay-adaptive feedback for linear feedforward systems. *Systems & Control Letters*, 59(5):277–283, may 2010. doi: 10.1016/j.sysconle.2010.03.001.

Betancourt, M., Jordan, M. I., and Wilson, A. C. On symplectic optimization. *arXiv preprint arXiv:1802.03653*, 2018.

Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006.

Bubeck, S. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8 (3-4):231–357, 2015. doi: 10.1561/2200000050.

Cao, C., Annaswamy, A. M., and Kojic, A. Parameter convergence in nonlinearly parameterized systems. *IEEE Transactions on Automatic Control*, 48(3):397–412, mar 2003. doi: 10.1109/TAC.2003.809146.

Carmon, Y., Duchi, J. C., Hinder, O., and Sidford, A. Accelerated methods for NonConvex optimization. *SIAM Journal on Optimization*, 28(2):1751–1772, jan 2018. doi: 10.1137/17M1114296.

Dean, S., Mania, H., Matni, N., Recht, B., and Tu, S. Regret bounds for robust adaptive control of the linear quadratic regulator. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 4192–4201. Curran Associates, Inc., 2018a.

Dean, S., Mania, H., Matni, N., Recht, B., and Tu, S. On the sample complexity of the linear quadratic regulator. *arXiv preprint arXiv:1710.01688*, 2018b.

Dean, S., Tu, S., Matni, N., and Recht, B. Safely learning to control the constrained linear quadratic regulator. *arXiv preprint arXiv:1809.10121*, 2018c.

Dietterich, T. G. Machine learning for sequential data: A review. In *Lecture Notes in Computer Science*, pp. 15–30. Springer Berlin Heidelberg, 2002. doi: 10.1007/3-540-70659-3_2.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, July 2011.

Duda, R. O., Hart, P. E., and Stork, D. G. *Pattern Classification, 2nd Edition*. John Wiley & Sons, 2001.

Evesque, S., Annaswamy, A. M., Niculescu, S., and Dowling, A. P. Adaptive control of a class of time-delay systems. *Journal of Dynamic Systems, Measurement, and Control*, 125(2):186, 2003. doi: 10.1115/1.1567755.

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

Gibson, T. E., Annaswamy, A. M., and Lavretsky, E. On adaptive control with closed-loop reference models: Transients, oscillations, and peaking. *IEEE Access*, 1:703–717, 2013. doi: 10.1109/ACCESS.2013.2284005.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016.

Hall, E. C. and Willett, R. M. Online convex optimization in dynamic environments. *IEEE Journal of Selected Topics in Signal Processing*, 9(4):647–662, jun 2015. doi: 10.1109/JSTSP.2015.2404790.

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

Hazan, E., Agarwal, A., and Kale, S. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, aug 2007. doi: 10.1007/s10994-007-5016-8.

Hazan, E., Rakhlin, A., and Bartlett, P. L. Adaptive online gradient descent. In Platt, J. C., Koller, D., Singer, Y., and Roweis, S. T. (eds.), *Advances in Neural Information Processing Systems 20*, pp. 65–72. Curran Associates, Inc., 2008.

Hazan, E., Singh, K., and Zhang, C. Learning linear dynamical systems via spectral filtering. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan,

S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 6702–6712. Curran Associates, Inc., 2017.

Hazan, E., Lee, H., Singh, K., Zhang, C., and Zhang, Y. Spectral filtering for general linear dynamical systems. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 4639–4648. Curran Associates, Inc., 2018.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Computation*, 9(8):1735–1780, nov 1997. doi: 10.1162/neco.1997.9.8.1735.

Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, apr 1982. doi: 10.1073/pnas.79.8.2554.

Hopfield, J. J. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences*, 81(10): 3088–3092, may 1984. doi: 10.1073/pnas.81.10.3088.

Ioannou, P. A. and Kokotovic, P. V. Robust redesign of adaptive control. *IEEE Transactions on Automatic Control*, 29(3):202–211, mar 1984. doi: 10.1109/TAC.1984.1103490.

Ioannou, P. A. and Sun, J. *Robust Adaptive Control*. PTR Prentice-Hall, 1996.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Jordan, M. I. Attractor dynamics and parallelism in a connectionist sequential machine. In *Proc. of the Eighth Annual Conference of the Cognitive Science Society*, 1986.

Jordan, M. I. and Mitchell, T. M. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, jul 2015. ISSN 0036-8075. doi: 10.1126/science.aaa8415.

Karason, S. P. and Annaswamy, A. M. Adaptive control in the presence of input constraints. *IEEE Transactions on Automatic Control*, 39(11):2325–2330, 1994. doi: 10.1109/9.333787.

Kingma, D. P. and Ba, J. L. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2017.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 25*, pp. 1097–1105. Curran Associates, Inc., 2012.

Kuznetsov, V. and Mohri, M. Learning theory and algorithms for forecasting non-stationary time series. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 28*, pp. 541–549. Curran Associates, Inc., 2015.

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

Luenberger, D. G. *Optimization by Vector Space Methods*. John Wiley & Sons, 1969.

Morse, A. S. High-order parameter tuners for the adaptive control of linear and nonlinear systems. In *Systems, Models and Feedback: Theory and Applications*, pp. 339–364. Birkhuser Boston, 1992. doi: 10.1007/978-1-4757-2204-8_23.

Narendra, K. S. and Annaswamy, A. M. *Stable Adaptive Systems*. Dover, 2005.

Nesterov, Y. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27:372–376, 1983.

Nesterov, Y. *Introductory Lectures on Convex Optimization*. Springer, 2004. doi: 10.1007/978-1-4419-8853-9.

O'Donoghue, B. and Candès, E. Adaptive restart for accelerated gradient schemes. *Foundations of Computational Mathematics*, 15(3):715–732, jul 2013. doi: 10.1007/s10208-013-9150-3.

Raginsky, M., Rakhlin, A., and Yuksel, S. Online convex programming and regularization in adaptive control. In *49th IEEE Conference on Decision and Control (CDC)*. IEEE, 2010. doi: 10.1109/CDC.2010.5717262.

Recht, B. A tour of reinforcement learning: The view from continuous control. *arXiv preprint arXiv:1806.09460*, 2018.

Sastry, S. and Bodson, M. *Adaptive Control: Stability, Convergence and Robustness*. Prentice-Hall, 1989.

Shalev-Shwartz, S. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2011. doi: 10.1561/2200000018.

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

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. On the importance of initialization and momentum in deep learning. In Dasgupta, S. and McAllester, D. (eds.), *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 1139–1147. PMLR, 2013.

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

Wilson, A. *Lyapunov Arguments in Optimization*. PhD thesis, University of California, Berkeley, 2018.

Wilson, A. C., Recht, B., and Jordan, M. I. A lyapunov analysis of momentum methods in optimization. *arXiv preprint arXiv:1611.02635*, 2016.

Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., and Recht, B. The marginal value of adaptive gradient methods in machine learning. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 4148–4158. Curran Associates, Inc., 2017.

Yu, S.-H. and Annaswamy, A. M. Neural control for nonlinear dynamic systems. In Touretzky, D. S., Mozer, M. C., and Hasselmo, M. E. (eds.), *Advances in Neural Information Processing Systems 8*, pp. 1010–1016. MIT Press, 1996.

Yu, S.-H. and Annaswamy, A. M. Stable neural controllers for nonlinear dynamic systems. *Automatica*, 34(5):641–650, may 1998. doi: 10.1016/S0005-1098(98)00012-0.

Zinkevich, M. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 928–936, 2003.