Connections Between Adaptive Control and Optimization in Machine Learning

Joseph E. Gaudio[1], Travis E. Gibson[2], Anuradha M. Annaswamy[1], Michael A. Bolender[3], and Eugene Lavretsky[4]

[1]Massachusetts Institute of Technology
[2]Brigham and Women's Hospital and Harvard Medical School
[3]Air Force Research Laboratory
[4]The Boeing Company

April 11, 2019

Abstract

This paper demonstrates many immediate connections between adaptive control and optimization methods commonly employed in machine learning. Starting from common output error formulations, similarities in update law modifications are examined. Concepts in stability, performance, and learning, common to both fields are then discussed. Building on the similarities in update laws and common concepts, new intersections and opportunities for improved algorithm analysis are provided. In particular, a specific problem related to higher order learning is solved through insights obtained from these intersections.

1 Introduction

The fields of adaptive control and machine learning have evolved in parallel over the past few decades, with a significant overlap in goals, problem statements, and tools. Machine learning as a field has focused on computer based systems that improve through experience [1–6]. Often times the process of learning is encapsulated in the form of a parameterized model, whose parameters are learned in order to approximate a function. Optimization methods are commonly employed to reduce the function approximation error using any and all available data. 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 [7–11]. The approach used for accomplishing such regulation and tracking in adaptive control is the learning of 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.

Learning parameters of a model in both machine learning and adaptive control occurs through the use of input-output data. In both cases, the main algorithm used for updating the parameters is based on a gradient descent-like algorithm [11]. Related tools of analysis, convergence, and

robustness in both fields have a tremendous amount of similarity. As the scope of problems in both fields increases, the associated complexity and challenges increase as well. Therefore it is highly attractive to understand these similarities and connections so that the two communities can develop new methods for addressing new challenges. In this paper, we discuss the similarities and connections in detail between the fields of adaptive control and machine learning. Using these connections, we state and provide a solution for a new problem in machine learning using methods developed in adaptive control.

In this paper, the adaptive control perspective will be presented in continuous time with machine learning material presented in discrete time. The paper organization is as follows. We introduce the formulation of output errors commonly employed in adaptive control and machine learning with their associated update laws in Section 2. Numerous connections between the two fields are then made with respect to the underlying parameter update laws in Section 3 and important concepts in Section 4. Examples of intersections between both fields are provided in Section 5, with concluding remarks in Section 6.

2 Problem Statements

In this section, we state typical problems that are addressed in the areas of adaptive control and machine learning. In both cases, we illustrate the role of learning, the input-output data used, and the overall problem that is desired to be solved.

2.1 Adaptive Control

The main goal in adaptive control is to carry out problems such as estimation or tracking in the presence of parametric uncertainties. The underlying model that relates inputs, outputs, and the unknown parameters is assumed to stem from either the underlying physics or from data-driven approaches. Often these models take the form

$$y(t) = f_1(\phi(t), \theta^*) \tag{1}$$

or

$$\dot{x}(t) = f_2(x(t), u(t), \theta^*), \quad y(t) = f_3(x(t), u(t), \theta^*) \tag{2}$$

where $u \in \mathbb{R}^m$ is an exogenous input, $x \in \mathbb{R}^n$ denotes the state, $y \in \mathbb{R}^p$ corresponds to output measurements, $\phi \in \mathbb{R}^N$ corresponds to measured and computed variables, and $\theta^* \in \mathbb{R}^N$ denotes the uncertain parameter. In an estimation problem, the goal is to estimate the state x in (2) and output y in both (1), (2), alongside the unknown parameter θ^* simultaneously, using all available variables. In a control problem, the goal is to determine a control input u so that the output y in (2) follows a desired output $\hat{y}$.

A typical approach taken in order to solve the estimation problem in (1) is to choose an estimator structure of the form

$$\hat{y}(t) = f(\phi(t), \theta(t)) \tag{3}$$

where $\theta \in \mathbb{R}^N$ denotes the estimate of θ^* and adjust θ so that the estimation error $e_y = \hat{y} - y$ is minimized, i.e., choose a function $g_1(e_y, \phi)$ with

$$\dot{\theta}(t) = g_1(e_y(t), \phi(t)) \tag{4}$$

so that the estimator has bounded signals, $e_y(t)$ converges to zero and $\theta(t)$ converges to θ^*. Similarly, the control problem consists of constructing an output tracking error $e_y = \hat{y} - y$, where $\hat{y}$ denotes
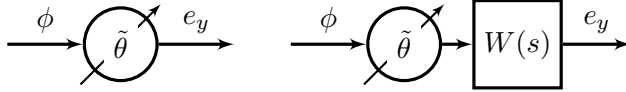
Figure 1: Error Models. **Left:** Regression (6), **Right:** Adaptive Control (7).

the desired output that y is required to track. The goal is to then choose functions $g_2(e_y, \phi, \theta)$ and $g_3(e_y, \phi, \theta)$ so that the control input u and a control parameter estimate θ can be chosen as

$$
\begin{aligned}
u(t) &= g_2(e_y(t), \phi(t), \theta(t)) \\
\dot{\theta}(t) &= g_3(e_y(t), \phi(t), \theta(t))
\end{aligned}
\tag{5}
$$

leading to closed-loop signals remaining bounded, $e_y(t)$ converging to zero and $\theta(t)$ converging to its true value θ^*. Denote the corresponding parameter errors as $\tilde{\theta} = \theta - \theta^*$.

In order to derive the function g_1 for the estimation problem in (1) and the functions g_2 and g_3 for the control problem in (2) so as to realize the underlying goals, a stability framework together with an error model approach is often employed in adaptive control. The error model approach consists of identifying the basic relationship between the two errors that are commonly present in these adaptive systems, which are the estimation (or tracking) error e_y and the parameter error $\tilde{\theta}$. While the estimation error is measurable and correlated with the parameter error, the parameter error is unknown but adjustable through the parameter estimate. In order to determine the update laws g_i, the relationship (error model) that relates these two errors is used as a cue.

Two types of error models frequently occur in adaptive systems, and are presented below (see Figure 1). The first corresponds to the case when the relation in (1) is linear, and the underlying error model is simply of the form (cf. [11])

$$
e_y(t) = \tilde{\theta}^T(t)\phi(t)
\tag{6}
$$

and as a result, the function g_1 in (4) can be determined simply using the gradient rule that minimizes $\|e_y\|^2$. The second is of the form (cf. [11])

$$
e_y(t) = W(s)[\tilde{\theta}^T(t)\phi(t)]
\tag{7}
$$

where $W(s)[\zeta]$ denotes a dynamic operator operating on $\zeta(t)$. It has been shown in the adaptive control literature [7–11] that for specific classes of dynamic operators $W(s)$, a stable, gradient-like rule can be determined for adjusting $\tilde{\theta}$. Most of these results apply uniformly to the case when u and y are scalars or vectors, with the latter introducing additional technicalities. In this paper we consider the case where inputs and outputs are scalars for notational simplicity, and to focus on the core of the learning problem with multi-dimensional regressors ϕ and parameter estimates θ. Often the unknown parameter θ^* is assumed to reside in a compact convex set, which we will denote as Θ.

2.2 Machine Learning

Machine learning is a broad field encompassing a wide variety of learning techniques and problems such as classification and regression. A large portion of machine learning considers supervised learning problems, where regressors ϕ and outputs y are related to one another in an unknown algebraic manner [1–6]. A typical approach taken in order to perform classification or regression is to choose an output estimator $\hat{y}_k$ parameterized with adjustable weights θ_k as

$$
\hat{y}_k = f(\phi_k, \theta_k).
\tag{8}
$$

3

A common form of the estimator as in (8) is that of neural networks, where the parameters θ_k represent the adjustable weights in the network [1–5].

Similar to adaptive control, θ_k is often adjusted using the output error $e_{y,k} = \hat{y}_k - y_k$. A loss function $L : \Theta \to \mathbb{R}$ of $e_{y,k}$ is minimized through the adjustable weights. An example loss function for regression is ℓ_p loss (with $p \in \mathbb{N}$, $p > 0$ and even) $L(\theta_k) = (1/p)\|e_{y,k}\|_p^p$. For binary classification ($y_k \in \{-1, 1\}$) common loss functions include hinge loss $L(\theta_k) = \max(0, 1 - y_k\hat{y}_k)$, and logistic loss $L(\theta_k) = \ln(1 + \exp(-y_k\hat{y}_k))$. Additionally, as in empirical risk minimization (ERM) [12], the total loss function considered for the purpose of a parameter update may be an average of loss functions over m samples as: $(1/m)\sum_{i=1}^{m} L_i(\theta_k)$. The above descriptions make it clear that the structure of the estimation problem in both adaptive control and machine learning are strikingly similar. In the next section, we examine the nature of the adjustment of θ_k.

2.3   Common Update Laws

As previously stated, the goal in adaptive control is to design a rule to adjust θ in an online continuous manner using knowledge of ϕ and e_y such that e_y tends toward zero. Given that the output errors may be corrupted by noise, an iterative, gradient-like update is usually employed. To do so for the algebraic error model (6), consider the squared loss cost function: $L(\theta(t)) = (1/2)e_y^2(t)$. The gradient of this function with respect to the parameters can be expressed as: $\nabla_\theta L(\theta(t)) = \phi(t)e_y(t)$. The standard gradient flow update law [7] may be expressed as follows with user-designed gain parameter $\gamma > 0$ as

$$\dot{\theta}(t) = -\gamma\nabla_\theta L(\theta(t)) = -\gamma\phi(t)e_y(t). \tag{9}$$

For dynamical error models such as (7), a stability approach rather than a gradient based one is taken using Lyapunov methods, which leads to an adaptive law identical to (9) for a class of dynamic systems $W(s)$ that are strictly positive real [7, 13].

The common update law for supervised machine learning problems, gradient descent[1], is akin to the time varying regression law (9) in discrete time, and of the form

$$\theta_{k+1} = \theta_k - \gamma_k\nabla_\theta L(\theta_k) \tag{10}$$

where the "stepsize" γ_k is usually chosen as a decreasing function of time [15–19], a standard feature of stochastic gradient algorithms.

3   Connections: Update Law

This section details a variety of connections between adaptive control and the optimization methods commonly used in machine learning as viewed from the perspective of their common update laws (9), (10).

3.1 σ-Modification, e-Modification, and Regularization

While the update laws in (9) and (10) are designed primarily to reduce the output error e_y, there are several secondary reasons to modify these update laws from robustness considerations due to perturbations stemming from disturbances, noise, and other unmodeled causes. We outline these updates in in this section.

---

[1]While this is not true of all machine learning as the field is broad, (for example Bayesian methods often use sampling based techniques such as Markov Chain Monte Carlo), even in the world of probabilistic inference, gradient based methods can also be used, cf. variational inference [14].

3.1.1 Adaptive Control

Historically the adaptive update law in (9) has been modified to ensure robustness in the presence of bounded disturbances as

$$\dot{\theta}(t) = -\gamma \left[\nabla_\theta L(\theta(t)) + \sigma \mathcal{G}(\theta(t), e_y(t)) \right] \tag{11}$$

where $\sigma > 0$ is a tuneable parameter that scales the extra term $\mathcal{G}$. Common choices for $\mathcal{G}$ include the σ-modification $\mathcal{G} = \theta$ [20], and the e-modification $\mathcal{G} = \|e_y\|\theta$ [21].

3.1.2 Machine Learning

Regularization is often included in a machine learning optimization problem in order to help cope with overfitting by including constraints on the parameter, thus resulting in an augmented loss function [1–5, 16–18]: $\bar{L}(\theta) = L(\theta) + \sigma \mathcal{R}(\theta)$ where $\sigma > 0$ is a tunable parameter, often referred to as a Lagrange multiplier. The gradient descent update (10) for this augmented loss function is often referred to as the "regularized follow the leader" algorithm in online learning [17] and may be expressed as

$$\theta_{k+1} = \theta_k - \gamma_k \left[\nabla_\theta L(\theta_k) + \sigma \nabla_\theta \mathcal{R}(\theta_k) \right]. \tag{12}$$

The common choice of ℓ_p regularization in machine learning of $\mathcal{R} = (1/p)\|\theta\|_p^p$ with $p = 2$, (as in ridge regression), coincides with the σ-modification [20], as then $\nabla_\theta \mathcal{R} = \mathcal{G}$. Given that the dimension of the parameter vector may be large, a sparse representation is often obtained with ℓ_1 regularization (as in lasso), with $\mathcal{R} = \|\theta\|_1$ [2–5].

3.2 Deadzone Modification and Early Stopping

This subsection details common modifications of the adaptive law adopted to cease updating the parameter estimate after sufficient tuning.

3.2.1 Adaptive Control

Another method employed to increase robustness in the presence of bounded disturbances is to employ a "dead zone" [22], for the update in (9) as

$$\dot{\theta}(t) = \begin{cases} -\gamma \nabla_\theta L(\theta(t)), & \mathcal{D}(e_y) > d_0 + \epsilon \\ 0, & \mathcal{D}(e_y) \leq d_0 + \epsilon \end{cases} \tag{13}$$

where $d_0 > 0$ is the dead zone width that may correspond to an upper bound on the disturbance, and $\epsilon > 0$ is a small constant. The function $\mathcal{D}$ is a non-negative metric on the output error to stop adaptation in desired regions of the output space. A common choice is $\mathcal{D} = \|e_y\|$ such that adaptation stops after a small output error is achieved above a noise level with upper bound d_0.

3.2.2 Machine Learning

The training processes is often stopped in machine learning applications as a method to deal with overfitting [2–5, 23]. This may be done by using multiple data sets and stopping the parameter update process (10) when the loss computed for a validation data set starts to increase [23]. Early stopping is often seen to be needed for training neural networks due to their large number of parameters [2–5] and can act as regularization [24].

3.3 Projection

It is often desirable to define a compact region a priori for the parameters θ, such that during the learning process the parameters are not allowed to leave that region. In physical systems there are natural constraints which may aid in the design of that region, and for non physical systems, the constraints are often engineered by the algorithm designer.

3.3.1 Adaptive Control

A continuous projection algorithm is commonly employed to provide for robustness of the adaptive update law in the presence of unmodeled dynamics [25–27]. One such implementation is

$$\text{Proj}(\theta_i, \zeta_i) = \begin{cases} \frac{\theta_{i,\max}^2 - \theta_i^2}{\theta_{i,\max}^2 - \theta_{i,\max}'^2} \zeta_i, & \theta_i \in \Omega_i \wedge \theta_i \zeta_i > 0 \\ \zeta_i, & \text{otherwise} \end{cases} \tag{14}$$

where Ω, $\theta_{i,\max}$, $\theta_{i,\max}'$ define a user-specified boundary layer region inside of Θ (see [26]). The update law in (9) may then be modified as

$$\dot{\theta}(t) = -\gamma \text{Proj}\left[\theta(t), \nabla_\theta L(\theta(t))\right]. \tag{15}$$

3.3.2 Machine Learning

Projected gradient descent methods have a long history in optimization. The following projection operation finds the point in a convex set which is closest to a specified point, and may be defined as

$$\Pi_\Theta(\bar{\theta}) \triangleq \underset{\theta \in \Theta}{\arg\min} \|\theta - \bar{\theta}\| \tag{16}$$

which may be employed in the update sequence [15–19]

$$\bar{\theta}_{k+1} = \theta_k - \gamma_k \nabla_\theta L(\theta_k), \qquad \theta_{k+1} = \Pi_\Theta(\bar{\theta}_{k+1}). \tag{17}$$

3.4 Adaptive Gains and Stepsizes

3.4.1 Adaptive Control

The following parameter update law for the algebraic error model[2] (6) is one example which alters the gain of the standard update law (9) as a function of the time varying regressors ϕ [7, 10]:

$$\dot{\theta}(t) = -\gamma \Gamma(t) \nabla_\theta L(\theta(t))$$
$$\dot{\Gamma}(t) = \begin{cases} \Upsilon\Gamma(t) - \frac{\Gamma(t)\phi(t)\phi^T(t)\Gamma(t)}{\mathcal{N}(t)}, & \|\Gamma(t)\| \leq \Gamma_{\max} \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

where $\Upsilon \geq 0$ is a *forgetting factor* and $\mathcal{N}(t)$ is a *normalizing signal*, with common choice $\mathcal{N}(t) = (1 + \mu\phi^T(t)\phi(t))$ for $\mu > 0$ chosen appropriately (see for example [10] for a discussion of the choice of parameters). It can be seen that the update for Γ may be integrated and used in the update for θ to result in a gain adaptive to the regressor ϕ.

---

[2]This update law has not been proven stable for the error model in (7).

3.4.2 Machine Learning

Adaptive step size methods [28–31] have seen widespread use in machine learning problems due to their ability to handle sparse and small gradients by adjusting the step size as a function of features as they are processed online. Define the following: $g_k = \nabla_\theta L(\theta_k)$, $m_k = \mathcal{F}_{1,k}(g_1, \ldots, g_k)$, $V_k = \mathcal{F}_{2,k}(g_1, \ldots, g_k)$ for user defined averaging functions $\mathcal{F}_{1,k}$, $\mathcal{F}_{2,k}$. A common update law for adaptive step size methods [31] can then be seen to be similar to (17) as

$$\bar{\theta}_{k+1} = \theta_k - \gamma_k m_k / V_k^{1/2}, \qquad \theta_{k+1} = \Pi_\Theta(\bar{\theta}_{k+1}) \tag{19}$$

where the following parameterizations are common [31]: (i) projected gradient descent[3] (17), (ii) AdaGrad[4] [28], and (iii) Adam[5] [30]. It can be noted that the normalization in these update laws is a function of the gradient, which can be compared to the normalization by the regressor in (18).

4 Connections: Tools and Concepts

This section details concepts and tools common to both machine learning and adaptive control.

4.1 Lyapunov Functions and Regret

Stability and convergence tools in adaptive control and online machine learning are analyzed in this section.

4.1.1 Adaptive Control

Suppose we consider the error model in (7) where $W(s) = c(sI - A)^{-1}b$, and a corresponding state space representation of the form [7]

$$\dot{e}(t) = Ae(t) + b\tilde{\theta}^T(t)\hat{\phi}(t) + b\theta^{*T}\tilde{\phi}(t)$$
$$e_y(t) = ce(t). \tag{20}$$

The term $\tilde{\phi}$ is due to exponentially decaying terms in the regressor ϕ. That is, $\tilde{\phi} = \hat{\phi} - \phi$ and $\dot{\tilde{\phi}} = \Lambda\tilde{\phi}$ for a Hurwitz matrix $\Lambda \in \mathbb{R}^{N \times N}$.[6] Stability is often proven in adaptive control by the use of a Lyapunov function V, such as

$$V = \gamma^{-1}\tilde{\theta}^T\tilde{\theta} + e^T Pe + \alpha\tilde{\phi}^T\bar{P}\tilde{\phi}. \tag{21}$$

It should be noted that the last two terms in V are not needed for the algebraic error model in (6). The time derivative of the Lyapunov function may then be stated using the update law in (9) and the KYP lemma [7] as

$$\dot{V} = -e^T Qe - \alpha\tilde{\phi}^T\bar{Q}\tilde{\phi} + 2e^T Pb\theta^{*T}\tilde{\phi} \tag{22}$$

---

[3] $\mathcal{F}_{1,k} = g_k$, $\mathcal{F}_{2,k} = I$.

[4] $\mathcal{F}_{1,k} = g_k$, $\mathcal{F}_{2,k} = \epsilon I + \text{diag}(\sum_{i=1}^k g_i^2)$, where $g_i^2 = g_i \odot g_i$.

[5] $\mathcal{F}_{1,k} = (1 - \beta_1)\sum_{i=1}^k \beta_1^{k-i}g_i$, $\mathcal{F}_{2,k} = (1 - \beta_2)\text{diag}(\sum_{i=1}^k \beta_2^{k-i}g_i^2)$.

[6] This formulation is common in the design of non-minimal adaptive observers [7]. It can be noted that $\hat{\phi} \to \phi$ as $t \to \infty$ as Λ is Hurwitz. Also for $\hat{\phi} = \phi$, (20) is the same as (7). A Hurwitz matrix Λ implies the existence of a positive definite matrix $\bar{P} = \bar{P}^T \in \mathbb{R}^{N \times N}$ and $0 < \bar{Q} = \bar{Q}^T \in \mathbb{R}^{N \times N}$ such that: $\Lambda^T\bar{P} + \bar{P}\Lambda = -\bar{Q}$.

where $\dot{V} \leq 0$ for $\alpha > (4\|Pb\|^2\|\theta^*\|^2/(\min eig(Q) \cdot \min eig(\bar{Q}))$. It can be shown [7] that $\delta(t) \triangleq 2e^T Pb\theta^{*T}\tilde{\phi}$ is an exponentially decaying signal with $\tilde{\phi}, e \in \mathcal{L}_2 \cap \mathcal{L}_\infty$. By integrating $\dot{V}$ from t_0 to T, we obtain

$$\int_{t_0}^{T} e^T Qe\,dt - \int_{t_0}^{T} \delta(t)dt \leq -\int_{t_0}^{T} \dot{V}dt = V(t_0) - V(T). \tag{23}$$

Given that $\dot{V} \leq 0$, $V(t_0) - V(T) \leq V(t_0) < \infty$.

4.1.2 Machine Learning

In online learning, efficiency of an algorithm is often analyzed using the notion of "regret" as

$$\text{regret}_T = \sum_{k=1}^{T} \mathcal{C}_k(\theta_k) - \min_{\theta \in \Theta} \sum_{k=1}^{T} \mathcal{C}_k(\theta) \tag{24}$$

where regret can be seen to correspond to the sum of the time varying convex costs $\mathcal{C}_k$ associated with the choice of the time varying parameter estimate θ_k, minus the cost associated with the best static parameter estimate choice, over a time horizon of T steps [15–17,19]. Suppose we consider a quadratic cost $\mathcal{C}_k = e_k^T Q e_k$, $Q = Q^T > 0$. A continuous time limit of (24) leads to an integral as

$$\text{continuous regret}_T = \int_{t_0}^{T} e^T Qe\,dt - \int_{t_0}^{T} \bar{\delta}(t)dt \tag{25}$$

where $\bar{\delta}(t)$ is an exponentially decaying signal which is due to nonzero initial conditions in (7) or similarly in (20).[7] A strong similarity can thus be seen between (23) and (25).

It is desired to have regret grow sub-linearly with time, such that average regret, $(1/T)\text{regret}_T$, goes to zero in the limit $T \to \infty$, to provide for an efficient algorithm [17]. Average regret can be connected to convergence in the case of a constant cost and by applying Jensen's inequality as [17]

$$\mathcal{C}(\bar{\theta}_T) - \mathcal{C}(\theta^*) \leq \frac{1}{T}\sum_{k=1}^{T}[\mathcal{C}(\theta_k) - \mathcal{C}(\theta^*)] = \frac{\text{regret}_T}{T} \tag{26}$$

where $\bar{\theta}_T = (1/T)\sum_{k=1}^{T} \theta_k$ is the average parameter estimate. Here sub-linear regret helps show convergence of the costs in (26). For adaptive control, convergence of state/output errors is shown from a similar integral which is akin to *constant* regret upper bounded by $V(t_0)$ in (23).

4.2 Unmodeled Dynamics and Generalization

This section discusses robustness to unforeseen perturbations such as unmodeled dynamics and unseen data.

4.2.1 Adaptive Control

Models used to design adaptive controllers, including the examples of (6) and (7), are linearized approximations with a certain amount of modeling errors. As such, they may only hold about an operating point and need to contend with unmodeled dynamics. This implies that any stabilizing controllers must be designed to not only adapt to parametric uncertainties, but also be robust

---

[7]This may be seen by setting $\theta(t) \equiv \theta^*$ in (7) or (20), thus resulting in an exponentially decreasing $e^T Qe$. Note that this exponentially decaying term is absent in the time varying regression case (6).

to unmodeled dynamics. In addition, constraints on the state and input may also be present in adaptive control problems [32, 33]. Analysis becomes more complicated when considering such unmodeled dynamics and constraints, resulting in non-global guarantees. Many of the update law modification in adaptive control from Section 3 were initially derived to ensure robustness in such cases.

4.2.2 Machine Learning

This same notion of robustness to modeling errors exists in machine learning in which an estimator $\hat{y}$ is constructed from a finite training data set, often with a finite number of tuneable parameters. It is then desired that this estimator produces a low prediction error based on a test data set consisting of not just seen data, but unseen data as well. Generalization in machine learning thus refers to the concept of a designed estimator having low loss when applied to new problems. In particular it can be seen that in specific cases, generalization pertains to stability, where algorithms that are stable and train in a small amount of time result in a small generalization error [34, 35].

4.3 Persistence of Excitation and Stochastic Perturbations

This section discusses conditions under which parameter estimates can be guaranteed to converge to their true values.

4.3.1 Adaptive Control

Persistence of excitation (PE) of the system regressor in adaptive control is a condition that has been shown to be necessary and sufficient for parameter convergence [36]. It can be shown that if the regressor ϕ is persistently exciting, then the algebraic error model (6) parameter estimation error $\tilde{\theta}(t)$ converges to zero uniformly in time [7]. Similar conditions can be imposed for the dynamical error model (7) and update law (9) [7]. The PE condition essentially corresponds to certain spectral conditions being satisfied by the regressor [37].[8] Parameter convergence can also occur through the use of "the hybrid algorithm", "the integral algorithm", "the algorithm with time-varying adaptive gains", and "the algorithm using multiple models" as is discussed in [7]. A detailed exposition of system identification and parameter convergence in both deterministic and stochastic cases can be found in [39–43]. Another way to think of the PE condition is that it leads to a perfect test error, since it provides for convergence of the parameter error to zero, and therefore zero output/state error once transients decay to zero.

4.3.2 Machine Learning

Many machine learning problems consider the case when stochastic perturbations are present. In this context, significant improvements may be possible by leveraging well known concepts in system identification [43]. For example [44] purposely includes a Gaussian random input into a dynamical system in order to provide for PE by construction. Such stochastic perturbations can guarantee a PE condition only in the limit, when infinite samples can be obtained. In order to address the realistic case of finite samples, approaches in machine learning algorithms for system identification and control have attempted to obtain performance bounds with probability $1 - p_f$ for $p_f \in (0, 1)$.[9] The probability of failure given by the choice of p_f allows for error due to the presence of finite samples.

---

[8]In particular, [38] established a condition on spectral lines of signals.
[9]The performance bound usually scales inversely with p_f as well.

4.4   Tracking vs Exploration

The concept of exploration can be viewed as the opposite of tracking, with the former often employed in machine learning while the latter is one of the main control goals.

4.4.1   Adaptive Control

The goal of adaptive control is to adjust the parameter θ in such a way to minimize the output error e_y in (6) and (7). It can be seen from the error models in (6) and (7) with the update in (9), that as the output error e_y goes to zero, learning becomes less and less, and that it is possible for a large parameter error to remain even with zero output (or tracking) error. That is, in many adaptive control applications, stability and tracking are successfully accomplished even without parameter convergence.

4.4.2   Machine Learning

In many machine learning methods, including reinforcement learning, there exist explicit modifications to update laws to promote exploration of the parameter space. These modifications include restarting trajectories with random initial conditions, adding random perturbations to algorithms, and driving the system towards a non-zero error regions [44–46]. This preference of exploration and learning over stability is motivated by the desire to find optimal parameters of a system. Stability is not always crucial as models are often trained with offline data on a computer, allowing for many iterations without the financial cost of failure present in physical systems (i.e. a nonzero probability of failure p_f is acceptable).

4.5   Convergence Guarantees

Notions of convergence guarantees are of importance in both fields, and are discussed here.

4.5.1   Adaptive Control

Adaptive control problems are often parameterized in a specific way such that e_y goes to zero asymptotically as in (6) and (7). Parameter convergence is shown to occur in these cases with a persistence of excitation condition (see Section 4.3.1). The specific parameterizations in the output space ensure that a global minimum of $e_y = 0$ exists and is unique. In the absence of PE, standard adaptive control algorithms converge to one of the many local minima in the parameter space (i.e. $\dot{\tilde{\theta}} \to 0$ but $\tilde{\theta} \neq 0$) [7].

4.5.2   Machine Learning

Machine learning has rapidly grown in recent years, as demonstrated by highly popular and well attended conferences such as ICML and NeurIPS, where rigorous proofs of stability are not always the main focus, instead focusing on empirical performance on large scale problems. A notable exception is a body of work that is emerging which consists of optimization-centric problem formulations, and the examination of the loss landscape, where recent results have shown that in certain classes of problems, local minimums are nearly equivalent to global minimums in terms of performance on test data [47–49].

4.6 Neural Networks

This section discusses neural networks, a topic common to both fields.

4.6.1 Adaptive Control

Gradient based methods to solve for estimates of unknown parameters via back propagation, in what would develop into the foundations of neural networks have been used for decades in control, with early examples consisting of finding optimal trajectories [50] in flight control [51], and resource allocation problems [52] (see [53] for a brief history). Since then, the use of neural networks in control systems has expanded to include stabilizing nonlinear dynamical systems [54]. Design and analysis of stable controllers based on neural networks was taken up by the adaptive control community due to the the similarities of gradient-like update laws used in neural networks and adaptive control. The adaptive control community developed a well established literature for the use of neural networks in nonlinear dynamical systems in the 1990s [54–58].

4.6.2 Machine Learning

The use of neural networks in the machine learning community greatly expanded as of recent due to the increase in computing power available and an increase in applications [5, 59, 60]. Recurrent neural networks [61–63], while often similar in structure to nonlinear dynamical systems, have historically been trained in a manner similar to feed-forward neural networks [64] using back propagation through time [65].[10] While a theoretical understanding of why deep neural networks work as well as they do for given problems has been lacking, the machine learning community has worked to rigorously analyze sub-classes of deep neural network architectures such as deep linear networks [67, 68]. The update laws employed in training deep neural networks often include selections of modifications of the update laws as discussed in Section 3. For an overview of the history of neural networks see [69].

4.7 Other Parameterization Schemes

In addition to neural networks as discussed in the previous section, other parameterizations are often considered in adaptive control and machine learning.

4.7.1 Adaptive Control

Adaptive control schemes often consider the case where an unknown parameter occurs linearly with respect to a regressor vector ϕ and may be related to an output error e_y algebraically (6) or dynamically (7). Often times the vector ϕ is a nonlinear function of the state of the system or reference model in order to approximate a more general nonlinear function D as: $D(x) = \theta^{*T}\phi(x)$ [70]. Common parameterizations for unknown nonlinearities include Gaussian radial basis functions [70]. Another class of parameterizations consist of nonlinearly parameterized uncertainty $D(\theta^*, \phi)$ in dynamical systems, for which there exists stabilizing adaptive control methods [71, 72].

4.7.2 Machine Learning

Parametric methods are common in machine learning as well, and are useful in many regression and classification based tasks [1–5]. However, Bayesian based approaches are also widespread in areas

---

[10]Hebbian learning [66] based approaches have also been considered.

such as topic models [73], clustering [74] and graphical models [75]. Additionally, new results in high dimensional statistics are increasingly being considered in which the model may be of higher dimension than the sample size [76].

5   Advantageous Combinations of Machine Learning and Adaptive Control Tools

Given the enormous number of similarities in problem statements, tools, concepts, and algorithms, it is natural to examine what the benefits are that accrue by combining insights obtained in these two different communities. Two examples of such an exercise is delineated in this section.

5.1   Higher Order Learning

Many of the update laws addressed thus far were first-order in nature, and made use of gradient-like quantities for learning. A question of increasing interest in the ML community is when accelerated learning can occur for higher-order learning methods. Higher order learning methods are commonly used in machine learning practice [59, 60, 77] as they can provide for a guaranteed bound on a faster rate of convergence. In particular, Nesterov's accelerated method [78] was able to certify a convergence rate of $O(1/k^2)$ as compared to the standard gradient descent (10) rate of $O(1/k)$ for a class of convex functions. A parameterization of Nesterov's accelerated method may be stated as

$$
\begin{aligned}
\theta_{k+1} &= \vartheta_k - \gamma \nabla_\theta L(\vartheta_k) \\
\vartheta_k &= \theta_k + \beta(\theta_k - \theta_{k-1})
\end{aligned}
\tag{27}
$$

where $\beta > 0$ is a design parameter that weighs the effect of past parameters. Continuous time problem formulations have been explored in [79, 80], with rate-matching discretizations established in [81–83]. Many of these methods however become inadequate for time varying inputs.

Adaptive update laws which include additional levels of integration appeared in the "higher order tuners" in [84, 85], and take the form

$$
\begin{aligned}
\dot{\vartheta}(t) &= -\gamma \nabla_\theta L(\theta(t)) \\
\dot{\theta}(t) &= -\beta(\theta(t) - \vartheta(t)) \mathcal{N}(t)
\end{aligned}
\tag{28}
$$

where $\mathcal{N}(t) \triangleq (1 + \mu \phi^T(t)\phi(t))$ for a $\mu > 0$. This update law can be seen to be the standard first order update (9) passed through a time varying filter normalized by the regressor. It was shown in [86] that (28) can provide for rates comparable to accelerated methods in machine learning for static features [80]. In addition, in contrast to (27), the update law in (28) can be shown to be stable in the presence of time varying regressors as in (6) and as well as in adaptive control applications with error model as in (7) [86]. This extension of accelerated methods in machine learning to include time varying and dynamic error models was only possible by leveraging techniques from adaptive control [86].

5.2   Improved Algorithm Performance Bounds

Regret analysis common in online machine learning (see Section 4.1.2) can result in overly conservative bounds for the performance of an algorithm. In particular, in online projected gradient descent (17) for regression (6) with squared output error cost $\mathcal{C} = (1/2)e_y^2$, regret analysis guarantees

regret$_T = O(\sqrt{T})$ (cf. [17]). For the same regret cost function, one can guarantee regret$_T = O(1)$ (constant regret)[11], using adaptive control methods.

6   Conclusions

This paper explored many immediate connections between adaptive control and machine learning, both through common update laws as well as common concepts. Adaptive control as a field has focused on mathematical rigor and guaranteed convergence. The rapid advances in machine learning on the other hand have brought about a plethora of new techniques and problems for learning. This paper was written to elucidate the numerous common connections between both fields such that results from both may be leveraged together to solve new problems.

7   Acknowledgements

References

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

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

[3] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, 2009.

[4] B. Efron and T. Hastie, *Computer Age Statistical Inference: Algorithms, Evidence and Data Science.* Cambridge University Press, 2016.

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

[6] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, pp. 255–260, jul 2015.

[7] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems.* NJ: Prentice-Hall, Inc., 1989. (out of print).

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

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

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

---

[11]For regression as in (6), regret contains a sum of non-negative costs and is therefore a non-decreasing function of the time horizon T. Thus $O(1)$ regret is the best achievable regret.

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

[12] V. Vapnik, "Principles of risk minimization for learning theory," in *Advances in Neural Information Processing Systems 4* (J. E. Moody, S. J. Hanson, and R. P. Lippmann, eds.), pp. 831–838, Morgan-Kaufmann, 1992.

[13] P. C. Parks, "Liapunov redesign of model reference adaptive control systems," *IEEE Transactions on Automatic Control*, vol. 11, pp. 362–367, jul 1966.

[14] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American Statistical Association*, vol. 112, pp. 859–877, feb 2017.

[15] E. Hazan, A. Agarwal, and S. Kale, "Logarithmic regret algorithms for online convex optimization," *Machine Learning*, vol. 69, pp. 169–192, aug 2007.

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

[17] E. Hazan, "Introduction to online convex optimization," *Foundations and Trends® in Optimization*, vol. 2, no. 3-4, pp. 157–325, 2016.

[18] S. Bubeck, "Convex optimization: Algorithms and complexity," *Foundations and Trends® in Machine Learning*, vol. 8, no. 3-4, pp. 231–357, 2015.

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

[20] P. A. Ioannou and P. V. Kokotovic, "Robust redesign of adaptive control," *IEEE Transactions on Automatic Control*, vol. 29, pp. 202–211, mar 1984.

[21] K. S. Narendra and A. M. Annaswamy, "A new adaptive law for robust adaptation without persistent excitation," *IEEE Transactions on Automatic Control*, vol. 32, pp. 134–145, feb 1987.

[22] B. B. Peterson and K. S. Narendra, "Bounded error adaptive control," *IEEE Transactions on Automatic Control*, vol. 27, pp. 1161–1168, dec 1982.

[23] L. Prechelt, "Automatic early stopping using cross validation: quantifying the criteria," *Neural Networks*, vol. 11, pp. 761–767, jun 1998.

[24] J. Sjöberg and L. Ljung, "Overtraining, regularization and searching for a minimum, with application to neural networks," *International Journal of Control*, vol. 62, pp. 1391–1407, dec 1995.

[25] G. Kreisselmeier and K. S. Narendra, "Stable model reference adaptive control in the presence of bounded disturbances," *IEEE Transactions on Automatic Control*, vol. 27, pp. 1169–1175, dec 1982.

[26] H. S. Hussain, *Robust Adaptive Control in the Presence of Unmodeled Dynamics.* PhD thesis, MIT, 2017.

[27] E. Lavretsky, T. E. Gibson, and A. M. Annaswamy, "Projection operator in adaptive systems," *arXiv preprint arXiv:1112.4232*, 2012.

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

[29] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

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

[31] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," in *International Conference on Learning Representations*, 2018.

[32] S. P. Karason and A. M. Annaswamy, "Adaptive control in the presence of input constraints," *IEEE Transactions on Automatic Control*, vol. 39, no. 11, pp. 2325–2330, 1994.

[33] A. M. Annaswamy and S. P. Kárason, "Discrete-time adaptive control in the presence of input constraints," *Automatica*, vol. 31, pp. 1421–1431, oct 1995.

[34] O. Bousquet and A. Elisseeff, "Stability and generalization," *Journal of Machine Learning Research*, vol. 2, pp. 499–526, Mar. 2002.

[35] M. Hardt, B. Recht, and Y. Singer, "Train faster, generalize better: Stability of stochastic gradient descent," in *Proceedings of The 33rd International Conference on Machine Learning* (M. F. Balcan and K. Q. Weinberger, eds.), vol. 48 of *Proceedings of Machine Learning Research*, (New York, New York, USA), pp. 1225–1234, PMLR, June 2016.

[36] B. M. Jenkins, A. M. Annaswamy, E. Lavretsky, and T. E. Gibson, "Convergence properties of adaptive systems and the definition of exponential stability," *SIAM Journal on Control and Optimization*, vol. 56, pp. 2463–2484, jan 2018.

[37] S. Boyd and S. S. Sastry, "Necessary and sufficient conditions for parameter convergence in adaptive control," *Automatica*, vol. 22, pp. 629–639, nov 1986.

[38] S. Boyd and S. Sastry, "On parameter convergence in adaptive control," *Systems & Control Letters*, vol. 3, pp. 311–319, dec 1983.

[39] G. C. Goodwin and K. S. Sin, *Adaptive Filtering Prediction and Control*. Prentice Hall, 1984.

[40] B. D. Anderson and C. Johnson, "Exponential convergence of adaptive identification and control algorithms," *Automatica*, vol. 18, pp. 1–13, jan 1982.

[41] K. S. Narendra and A. M. Annaswamy, "Robust adaptive control in the presence of bounded disturbances," *IEEE Transactions on Automatic Control*, vol. 31, pp. 306–315, apr 1986.

[42] K. S. Narendra and A. M. Annaswamy, "Persistent excitation in adaptive systems," *International Journal of Control*, vol. 45, pp. 127–160, jan 1987.

[43] L. Ljung, *System Identification: Theory for the User*. Prentice-Hall, 1987.

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

[45] R. S. Sutton, A. G. Barto, and R. J. Williams, "Reinforcement learning is direct adaptive optimal control," *IEEE Control Systems*, vol. 12, pp. 19–22, apr 1992.

[46] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction.* MIT Press, 2018.

[47] A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun, "The loss surfaces of multilayer networks," in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics* (G. Lebanon and S. V. N. Vishwanathan, eds.), vol. 38 of *Proceedings of Machine Learning Research*, pp. 192–204, PMLR, 2015.

[48] R. Ge, J. D. Lee, and T. Ma, "Matrix completion has no spurious local minimum," in *Advances in Neural Information Processing Systems 29* (D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, eds.), pp. 2973–2981, Curran Associates, Inc., 2016.

[49] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht, "Gradient descent only converges to minimizers," in *29th Annual Conference on Learning Theory* (V. Feldman, A. Rakhlin, and O. Shamir, eds.), vol. 49 of *Proceedings of Machine Learning Research*, (Columbia University, New York, New York, USA), pp. 1246–1257, PMLR, 23–26 Jun 2016.

[50] L. Pontryagin, *Mathematical Theory of Optimal Processes.* Routledge, may 1961.

[51] H. J. Kelley, "Gradient theory of optimal flight paths," *ARS Journal*, vol. 30, pp. 947–954, oct 1960.

[52] A. E. Bryson, "A gradient method for optimizing multistage allocation processes," in *Proc. Harvard Univ. Symposium on digital computers and their applications*, 1961.

[53] S. E. Dreyfus, "Artificial neural networks, back propagation, and the kelley-bryson gradient procedure," *Journal of Guidance, Control, and Dynamics*, vol. 13, pp. 926–928, sep 1990.

[54] W. T. Miller, R. S. Sutton, and P. J. Werbos, *Neural Networks for Control.* MIT press, 1995.

[55] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, pp. 4–27, mar 1990.

[56] K. S. Narendra and K. Parthasarathy, "Gradient methods for the optimization of dynamical systems containing neural networks," *IEEE Transactions on Neural Networks*, vol. 2, pp. 252–262, mar 1991.

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

[58] S.-H. Yu and A. M. Annaswamy, "Stable neural controllers for nonlinear dynamic systems," *Automatica*, vol. 34, pp. 641–650, may 1998.

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

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

[61] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences*, vol. 79, pp. 2554–2558, apr 1982.

[62] G. E. Hinton and T. J. Sejnowski, "Optimal perceptual inference," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (Washington, DC), pp. 448–453, June 1983.

[63] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, nov 1997.

[64] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, oct 1986.

[65] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[66] D. O. Hebb, *The Organization of Behavior*. Wiley, 1949.

[67] S. Arora, N. Cohen, and E. Hazan, "On the optimization of deep networks: Implicit acceleration by overparameterization," in *Proceedings of the 35th International Conference on Machine Learning* (J. Dy and A. Krause, eds.), vol. 80 of *Proceedings of Machine Learning Research*, (Stockholmsmässan, Stockholm Sweden), pp. 244–253, PMLR, July 2018.

[68] S. Arora, N. Cohen, N. Golowich, and W. Hu, "A convergence analysis of gradient descent for deep linear neural networks," in *International Conference on Learning Representations*, 2019.

[69] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, jan 2015.

[70] R. M. Sanner and J.-J. E. Slotine, "Gaussian networks for direct adaptive control," *IEEE Transactions on Neural Networks*, vol. 3, no. 6, pp. 837–863, 1992.

[71] A.-P. Loh, A. M. Annaswamy, and F. P. Skantze, "Adaptation in the presence of a general nonlinear parameterization: An error model approach," *IEEE Transactions on Automatic Control*, vol. 44, no. 9, pp. 1634–1652, 1999.

[72] C. Cao, A. M. Annaswamy, and A. Kojic, "Parameter convergence in nonlinearly parameterized systems," *IEEE Transactions on Automatic Control*, vol. 48, pp. 397–412, mar 2003.

[73] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, Jan. 2003.

[74] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Sharing clusters among related groups: Hierarchical dirichlet processes," in *Advances in Neural Information Processing Systems 17* (L. K. Saul, Y. Weiss, and L. Bottou, eds.), pp. 1385–1392, MIT Press, 2005.

[75] M. J. Wainwright and M. I. Jordan, *Graphical Models, Exponential Families, and Variational Inference*, vol. 1. Now Publishers, 2007.

[76] M. J. Wainwright, *High-Dimensional Statistics*. Cambridge University Press, feb 2019.

[77] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, pp. 183–202, jan 2009.

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

[79] W. Su, S. Boyd, and E. J. Candès, "A differential equation for modeling nesterov's accelerated gradient method: Theory and insights," *Journal of Machine Learning Research*, vol. 17, no. 153, pp. 1–43, 2016.

[80] A. Wibisono, A. C. Wilson, and M. I. Jordan, "A variational perspective on accelerated methods in optimization," *Proceedings of the National Academy of Sciences*, vol. 113, pp. E7351–E7358, nov 2016.

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

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

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

[84] A. S. Morse, "High-order parameter tuners for the adaptive control of linear and nonlinear systems," in *Systems, Models and Feedback: Theory and Applications*, pp. 339–364, Birkhäuser Boston, 1992.

[85] S. Evesque, A. M. Annaswamy, S. Niculescu, and A. P. Dowling, "Adaptive control of a class of time-delay systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 125, no. 2, p. 186, 2003.

[86] J. E. Gaudio, T. E. Gibson, A. M. Annaswamy, and M. A. Bolender, "Accelerated learning in the presence of time varying features with applications to machine learning and adaptive control," *arXiv preprint arXiv:1903.04666*, 2019.