ANNUAL REVIEWS

Annual Review of Control, Robotics, and Autonomous Systems

Adaptive Control and Intersections with Reinforcement Learning

Anuradha M. Annaswamy

Active Adaptive Control Laboratory, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA; email: aanna@mit.edu

Annu. Rev. Control Robot. Auton. Syst. 2023. 6:65–93

First published as a Review in Advance on January 6, 2023

The *Annual Review of Control, Robotics, and Autonomous Systems* is online at control.annualreviews.org

https://doi.org/10.1146/annurev-control-062922-090153

Keywords

adaptive control, reinforcement learning, adaptation, optimality, stability, robustness

Abstract

This article provides an exposition of the field of adaptive control and its intersections with reinforcement learning. Adaptive control and reinforcement learning are two different methods that are both commonly employed for the control of uncertain systems. Historically, adaptive control has excelled at real-time control of systems with specific model structures through adaptive rules that learn the underlying parameters while providing strict guarantees on stability, asymptotic performance, and learning. Reinforcement learning methods are applicable to a broad class of systems and are able to produce near-optimal policies for highly complex control tasks. This is often enabled by significant offline training via simulation or the collection of large input-state datasets. This article attempts to compare adaptive control and reinforcement learning using a common framework. The problem statement in each field and highlights of their results are outlined. Two specific examples of dynamic systems are used to illustrate the details of the two methods, their advantages, and their deficiencies. The need for real-time control methods that leverage tools from both approaches is motivated through the lens of this common framework.

1. INTRODUCTION

The overarching goals in the control of a large-scale system are to understand the underlying dynamics and offer decision support in real time so as to realize high performance. A control problem can be stated as the determination of inputs into the system so as to have its outputs lead to desired performance metrics, often related to efficiency, reliability, and resilience. The challenge that arises is that these decisions must be undertaken in the presence of uncertainties in the system and in the environment it operates in. Adaptive control (AC) and reinforcement learning (RL) are two different methods that have been explored over the past several decades to address this difficult problem in a range of application domains. This article attempts to present them using a common framework.

An interesting point to note is that the solutions for this problem have been proposed by AC and RL using two distinct frameworks. A fundamental concept that is common to both of these methodologies is learning. Despite this commonality, there has been very little cross-fertilization between these methods. Both methods have distinct advantages in their approach and at the same time have gaps in application to real-time control. This article presents a first step in providing a comparison between these two methods, exploring the role of learning, and describing the challenges that these two fields have encountered.

Sections 2 and 3 are devoted to laying out the fundamentals of AC, with particular emphasis on how learning occurs in the proposed solutions. This is followed by a brief exploration of RL in Section 4. Section 5 presents two different examples of dynamic systems that are used to illustrate the distinction between the two approaches. Finally, Section 6 is devoted to comparisons, combinations of the two approaches, and a few concluding remarks.

2. ADAPTIVE CONTROL: PROBLEM STATEMENT

The aim in AC is to design an exogenous input $u(t) \in \mathbb{R}^m$ that affects the dynamics of a system given by

$$\begin{aligned} \dot{x} &= f(x, \theta, u, t), \\ y &= g(x, \theta, u, t), \end{aligned} \qquad 1.$$

where $x(t) \in \mathbb{R}^n$ represents the system state and $y(t) \in \mathbb{R}^p$ represents all measurable system outputs. For many physical systems, $n \gg p > m$ (1). $\theta \in \mathbb{R}^\ell$ represents system parameters that may be unknown, and $f(\cdot)$ and $g(\cdot)$ denote system dynamics (which may be nonlinear) that capture the underlying physics of the system. The functions $f(\cdot)$ and $g(\cdot)$ also vary with t, as disturbances and stochastic noise may affect the states and output. The goal is to choose $u(t)$ so that $y(t)$ tracks a desired command signal $y_c(t)$ at all t, and so that an underlying cost $J((y - y_c), x, u)$ is minimized. In what follows, the system that is being controlled is referred to as a plant.

As the description of the system in Equation 1 is based on a plant model, and as the goal is to determine the control input in real time, all control approaches make assumptions regarding what is known and unknown. The functions f and g are often not fully known, as the plant is subject to various perturbations and modeling errors due to environmental changes, complexities in the underlying mechanisms, aging, and anomalies. The field of AC takes a parametric approach to distinguish the known parts from the unknown. In particular, it is assumed that f is a known function, while the parameter θ is unknown. A real-time control input is then designed so as to ensure that the tracking goals are achieved by including an adaptive component that attempts to estimate the parameters online. A similar problem statement can be made for a linearized version of the problem in Equation 1, which is of the form

$$y = W(s, \theta)[u], \qquad 2.$$

where s denotes the differential operator $\mathrm{d}/\mathrm{d}t$, $W(s,\cdot)$ is a rational operator of s, and θ is a parameter vector. In this linear case, we assume that the structure of $W(s)$ (i.e., the order and net order) is known but that θ (i.e., the coefficients) is not known.

The following subsections further break down the approach taken to address these problems, especially in the context of learning and optimization. While the description below is in the context of deterministic continuous-time systems, similar efforts have been carried out in stochastic and discrete-time dynamic systems as well (2).

2.1. An Adapt–Learn–Optimize Approach

The goal of the adaptive controller is to ensure that

$$\lim_{t\to\infty} e(t) = 0, \qquad 3.$$

where $e(t) = y(t) - y_c(t)$. As these decisions are required to be made in real time, the focus of the AC approach is to lead to a closed-loop dynamic system that has bounded solutions at all time t and a desired asymptotic behavior as in Equation 3. The central question is whether this can be ensured even when there are parametric uncertainties in θ and several other nonparametric uncertainties, which may be due to unmodeled dynamics, disturbances, or other unknown effects. Once this is guaranteed, the question of learning, in the form of parameter convergence, is addressed. As a result, control for learning is a central question that is pursued in the class of problems addressed in AC rather than learning for control (3). Once the control and learning objectives are realized, one can then proceed to the optimization of a suitable cost J. This sequence of adapt–learn–optimize is an underpinning of much of AC.

The above sequence can be reconciled with the well-known certainty equivalence principle, which proceeds in the following manner: First, optimize under perfect foresight, then substitute optimal estimates for unknown values. This philosophy underlies all AC solutions by first determining a controller structure that leads to an optimal solution when the parameters are known and then replacing the parameters in the controller with their estimates. The difficulty in adopting this philosophy to its fullest stems from the dual nature of the adaptive controller, as it attempts to accomplish two tasks simultaneously, estimation and control. This simultaneous action introduces a strong nonlinearity into the picture and therefore renders a true deployment of the certainty equivalence principle intractable. Instead, an adapt–learn–optimize sequence is adopted, with the first step corresponding to an adaptive controller that leads to a stable solution. This is then followed by estimation of the unknown parameters, and optimization is addressed at the final step.

A typical solution of the adaptive controller takes the form

$$u = C_1(\theta_c(t), \phi(t), t), \qquad 4.$$

$$\dot{\theta}_c = C_2(\theta_c, \phi, t), \qquad 5.$$

where $\theta_c(t)$ is an estimate of a control parameter that is intentionally varied as a function of time, and $\phi(t)$ represents all available data at time t. The nonautonomous nature of C_1 and C_2 is due to the presence of exogenous signals such as set points and command signals. A stabilization task would render these functions autonomous. The functions $C_1(\cdot)$ and $C_2(\cdot)$ are deterministic constructions and make the overall closed-loop system nonlinear and nonautonomous. The challenge is to suitably construct functions $C_1(t)$ and $C_2(t)$ so as to have $\theta_c(t)$ approach its true value θ_c^* and ensure the stability and asymptotic stability properties of the overall adaptive systems. Several textbooks have delineated these constructions for deterministic systems (e.g., 4–9). The solutions

in these books and several papers in premier control journals have laid the foundation for the construction of C_1 and C_2 for a large class of dynamic systems as in Equation 1.

2.2. Links to Learning and Robustness

With the problem statement as above, it is perhaps clear to the reader that the organic connection between the AC problem and learning enters through parameters. Given that what is unknown about the dynamics is the plant parameter θ—or, equivalently, the control parameter θ_c^*—learning is synonymous with accurate parameter estimation. That is, it is of interest to have the parameter estimate θ_c converge to θ_c^* in the context of control problems and for an estimate $\widehat{\theta}$ to converge to θ in the context of identification problems. The learning goal in either case is to determine conditions under which this convergence takes place. These conditions are linked to properties defined as persistent excitation and uniform observability (10–14). These persistent excitation properties are usually associated with the underlying regressor ϕ and are typically realized by appropriately choosing the exogenous signals such as $r(t)$ (which is the input into the reference model $\mathcal{M}$).

The assumption that the uncertainties in Equations 1 and 2 are limited to just the parameter θ, and that otherwise f and g or $W(s)$ is known, is indeed an idealization. Several departures from this assumption can take place in the form of unmodeled dynamics, time-varying parameters, disturbances, and noise. For example, the linear plant may have a form

$$y = [W(s, (\theta(t))) + \Delta(s)]\,[u + d(t) + n(t) + g(t)], \qquad 6.$$

where $d(t)$ is an exogenous bounded disturbance, $n(t)$ represents measurement noise, and $g(t)$ represents nonlinear effects. The parameter θ is time-varying and is of the form

$$\theta(t) = \theta^* + \vartheta(t), \qquad 7.$$

where θ^* is an unknown constant parameter but is accompanied by additional unknown variations in the form of $\vartheta(t)$. Finally $\Delta(s)$ is due to higher-order dynamics that is not known, poorly known, or even deliberately ignored for the sake of computational or algorithmic simplicity. In all of these cases, a robust adaptive controller needs to be designed to ensure that the underlying signals remain bounded, with errors that are proportional to the size of these perturbations. Similar departures of unknown effects that cannot be anticipated during online operation exist for the nonlinear system in Equation 1 as well. All AC methods have been developed with these departures from the idealized problem statements (as addressed in Section 2.1).

As will become apparent in Section 3, the results that have been proposed for a robust adaptive controller are intricately linked to learning of the underlying parameters. These aspects and implications of imperfect learning will be addressed in Section 3 as well.

3. ADAPTIVE CONTROL: RESULTS

A tractable procedure for determining the structure of the functions C_1 and C_2, denoted as model reference AC, uses the notion of a reference model and a two-step design, consisting of an algebraic part for determining C_1 and an analytic part for finding C_2. A reference model provides a structure to the class of command signals $y_c(t)$ that the plant output y can follow. For a controller to exist for a given plant model so as to guarantee output tracking, the signal y_c needs to be constrained in some sense. A reference model is introduced to provide such a constraint.

In particular, a model $\mathcal{M}$ and a reference input r is designed in such a way that the output $y_m(t)$ of $\mathcal{M}$ for an input $r(t)$ approximates the class of signals $y_c(t)$ that is desired to be followed. With a reference model in $\mathcal{M}$, the algebraic part of the model reference AC corresponds to the choice of C_1 with a fixed parameter θ_c^* such that if $\theta_c(t) \equiv \theta_c^*$ in Equation 4, then $\lim_{t\to\infty} y_p(t) - y_m(t) = 0$. The existence of such a θ^* is referred to as a matching condition. With

such a C_1 determined, and noting that θ_c^* could be unknown due to the parametric uncertainty in the plant, the analytic part focuses on finding C_2 such that output following takes place with the closed-loop system remaining bounded. An alternative to the above direct approach of identifying the control parameters is an indirect one where the plant parameters are first estimated, and these estimates are then used to determine the control parameter $\theta_c(t)$ at each t. The following sections describe the details of the model reference AC approach for various classes of dynamic systems, ranging from simple and algebraic cases to nonlinear dynamic ones.

3.1. Linear Plants

This section delineates four different classes of linear plants with parametric uncertainties and describes the adaptive solution to the problem.

3.1.1. Algebraic systems. Many problems in adaptive estimation and control may be expressed as (2)

$$y(t) = \theta^{*\mathrm{T}}\phi(t), \qquad 8.$$

where $\theta^*, \phi(t) \in \mathbb{R}^N$ represent an unknown parameter and measurable regressor, respectively, and $y(t) \in \mathbb{R}$ represents an output that can be determined at each t. Given that θ^* is unknown, we formulate an estimator $\hat{y}(t) = \theta^{\mathrm{T}}(t)\phi(t)$, where $\hat{y}(t) \in \mathbb{R}$ is the estimated output and the unknown parameter is estimated as $\theta(t) \in \mathbb{R}^N$. This in turn results in two types of errors, a performance error $e_y(t)$ and a learning error $\widetilde{\theta}(t)$,[1]

$$e_y = \hat{y} - y, \qquad \widetilde{\theta} = \theta - \theta^*, \qquad 9.$$

where the former can be measured but the latter is unknown, though adjustable. From Equation 8 and the estimator, it is easy to see that e_y and $\widetilde{\theta}$ are related using a simple regression relation:

$$e_y(t) = \widetilde{\theta}^{\mathrm{T}}\phi(t). \qquad 10.$$

A common approach for adjusting the estimate $\theta(t)$ at each time t is to use a gradient rule and a suitable loss function. One example is the choice

$$L_1(\theta) = \frac{1}{2}e_y^2, \qquad 11.$$

leading to the gradient rule

$$\dot{\theta}(t) = -\gamma\nabla_\theta L_1(\theta(t)), \qquad \gamma > 0. \qquad 12.$$

That this leads to a stable estimation scheme can be shown using a Lyapunov function, $V = \widetilde{\theta}^{\mathrm{T}}\widetilde{\theta}$, as its time derivative, $\dot{V} = -e_y^2$.

3.1.2. Dynamic systems with states accessible. The next class of problems that has been addressed in AC corresponds to plants with all states accessible. This section presents the solution for the simple case of a scalar input:

$$\dot{x} = A_p x + b_p u, \qquad 13.$$

where A_p and b_p are unknown, u is the control input and is a scalar, and x is the state and is accessible for measurement. As mentioned in Section 2.1, the first step is to find a reference model $\mathcal{M}$, which takes the form

$$\dot{x}_m = A_{\mathrm{h}}x_m + br \qquad 14.$$

[1] In what follows, the argument (t) is suppressed unless needed for emphasis.

and is such that the state $x_m(t)$ encapsulates the desired solution expected from the controlled plant. This can be accomplished by choosing a reference input r, choosing A_{h} to be a Hurwitz matrix, and choosing (A_{h}, b) to be controllable so that together they produce an $x_m(t)$ that approximates the signal that the plant is required to track.

With the reference model chosen as above, the next step pertains to the matching condition (4), stated as follows.

Assumption 1. A vector θ^* and a scalar k^* exist that satisfy

$$A_p + b_p \theta^{*\mathrm{T}} = A_{\mathrm{h}}, \tag{15.}$$

$$b_p k^* = b. \tag{16.}$$

Assumption 1 implies that a fixed control exists of the form

$$u(t) = \theta^{*\mathrm{T}} x(t) + k^* r(t) \tag{17.}$$

that matches the closed-loop system to the reference model. This corresponds to the algebraic part of the problem described in Section 2.1.

The final step is the analytic part—the rule for estimating the unknown parameters θ^* and k^* and the corresponding AC input that replaces the input choice in Equation 17. These solutions are given by

$$u = \theta^{\mathrm{T}}(t) x + k(t) r, \tag{18.}$$

$$\dot{\theta} = -\mathrm{sign}(k^*) \Gamma_\theta (e^{\mathrm{T}} P b_m) x, \tag{19.}$$

$$\dot{k} = -\mathrm{sign}(k^*) \gamma_k (e^{\mathrm{T}} P b_m) r, \tag{20.}$$

where $\Gamma_\theta > 0$ is a positive definite matrix, $\gamma_k > 0$ is a positive constant, $e = x - x_m$, and $P = P^{\mathrm{T}} \in \mathbb{R}^{n \times n}$ is a positive definite matrix that solves the Lyapunov equation

$$A_m^{\mathrm{T}} P + P A_m = -Q \tag{21.}$$

with a positive definite matrix $Q = Q^{\mathrm{T}} \in \mathbb{R}^{n \times n}$. It can be shown that

$$V = e^{\mathrm{T}} P e + |k^*| \left[(\theta - \theta^*)^{\mathrm{T}} \Gamma^{-1} (\theta - \theta^*) + (1/\gamma_k)(k - k^*)^2 \right] \tag{22.}$$

is a Lyapunov function with $\dot{V} = -e^{\mathrm{T}} Q e$ and that $\lim_{t \to \infty} e(t) = 0$ (for further details, see chapter 3 in Reference 4). In summary, the adaptive controller that is proposed here can be viewed as an action–response–correction sequence where the action is the control input given by Equation 18, the response is the resulting state error e, and the correction is the parameter-adaptive laws in Equations 19 and 20

It should be noted that the adaptation rules in Equations 19 and 20 can also be expressed as the gradient of a loss function (15),

$$L_2(\bar{\theta}) = \frac{\mathrm{d}}{\mathrm{d}t} \left\{ \frac{e^{\mathrm{T}} P e}{2} \right\} + \frac{e^{\mathrm{T}} Q e}{2}, \tag{23.}$$

where $\bar{\theta} = [\theta^{\mathrm{T}}, k]^{\mathrm{T}}$, and it is assumed that $k^* > 0$ for ease of exposition. It is noted that this loss function L_2 differs from that in Equation 11 and includes an additional component that reflects the dynamics in the system. It is easy to see that

$$\dot{\bar{\theta}}(t) = -\Gamma \nabla_{\bar{\theta}} L_2(\bar{\theta}(t)), \qquad \Gamma > 0, \tag{24.}$$

and that $\dot{\bar{\theta}}(t)$ is implementable as $\nabla_\theta L_2(\theta) = \phi e^{\mathrm{T}} P b_m$ and can be computed at each time t, where $\phi = [x_p^{\mathrm{T}}, r]^{\mathrm{T}}$ (15). This implies that a real-time control solution that is stable depends critically on choosing an appropriate loss function.

The matching condition given in Equation 16 is akin to the controllability condition, albeit somewhat stronger, as it requires the existence of a θ^* for a known Hurwitz matrix A_m (4, 16). The other requirement is that the sign of k^* must be known, which is required to ensure that V is a Lyapunov function.

3.1.3. Adaptive observers. The AC solution in Equations 18 and 19 in Section 3.1.2 required that the state $x(t)$ be available for measurement at each t. A central challenge in developing adaptive solutions for plants whose states are not accessible is the simultaneous generation of estimates of both states and parameters in real time. Unlike the Kalman filter in the stochastic case or the Luenberger observer in the deterministic case, the problem becomes significantly more complex, as state estimates require plant parameters, and parameter estimation is facilitated when states are accessible. This loop is broken using a nonminimal representation of the plant, leading to a tractable observer design. Starting with a plant model as in Equation 2, a state representation of the same can be derived as given by Luders & Narendra (17):

$$\begin{aligned} \dot{\omega}_1 &= \Lambda\omega_1 + \ell u, \\ \dot{\omega}_2 &= \Lambda\omega_2 + \ell y, \\ y &= \theta_1^{\mathrm{T}}\omega_1 + \theta_2^{\mathrm{T}}\omega_2, \end{aligned} \qquad 25.$$

where $\omega = [\omega_1^{\mathrm{T}}, \omega_2^{\mathrm{T}}]^{\mathrm{T}}$ is a nonminimal state of the plant transfer function $W_p(s)$ between the input u and the output y. $\Lambda \in \mathbb{R}^{n\times n}$ is a Hurwitz matrix, and Λ and ℓ are controllable and are known parameters. Assuming that $W_p(s)$ has n poles and m coprime zeros, in contrast to a minimal nth-order representation, Equation 25 is nonminimal and has $2n$ states. The adaptive observer leverages Equation 25 and generates a state estimate $\widehat{\omega}$ and a plant estimate $\widehat{\theta}$ as follows:

$$\begin{aligned} \dot{\hat{\omega}}_1 &= \Lambda\hat{\omega}_1 + \ell u, \\ \dot{\hat{\omega}}_2 &= \Lambda\hat{\omega}_2 + \ell y, \\ \hat{y} &= \widehat{\theta}_1^{\mathrm{T}}\hat{\omega}_1 + \widehat{\theta}_2^{\mathrm{T}}\hat{\omega}_2, \end{aligned} \qquad 26.$$

where $\widehat{\theta} = [\widehat{\theta}_1^{\mathrm{T}}, \widehat{\theta}_2^{\mathrm{T}}]^{\mathrm{T}}$ and $\hat{\omega} = [\hat{\omega}_1^{\mathrm{T}}, \hat{\omega}_2^{\mathrm{T}}]^{\mathrm{T}}$. The adaptive law that adjusts the parameter estimates is chosen as

$$\dot{\widehat{\theta}} = -\Gamma\left(\hat{y}_p - y_p\right)\widehat{\omega}, \qquad 27.$$

where Γ is a known symmetric, positive definite matrix.

Analytical guarantees of the stability of the parameter estimate $\widehat{\theta}$ in Equations 26 and 27 as well as asymptotic convergence of $\widehat{\theta}(t)$ to θ can be found in References 10 and 12. Necessary and sufficient conditions for this convergence require that the regressor $\hat{\omega}$ be persistently exciting. Several results also exist in ensuring accelerated convergence of these estimates (14, 18–22) using matrix regressors, a time-varying learning rate for Γ, and dynamic regressor extension and mixing.

3.1.4. Adaptive control with output feedback. The two assumptions made in the development of adaptive systems in Section 3.1.2 include matching conditions and the availability of states of the underlying dynamic system at each instant t. Both are often violated in many problems, which led to the development of adaptive systems when only partial measurements are available. With the focus primarily on linear time-invariant plants, the first challenge was to address the problem of the separation principle employed in the control of linear time-invariant plants (23, 24). The idea therein is to allow a simultaneous estimation of states using an observer and a feedback control using state estimates with a linear–quadratic regulator to be implemented, and to allow

them both to proceed simultaneously in real time and guarantee the stability of the closed-loop system. The challenge in the current context is that parameters are unknown, introducing an additional estimate (of the plant parameter) to be generated in real time. In contrast to the classical problem, where the closed loop remains linear, the simultaneous problem of generating the parameter estimate to determine the controller and the feedback control to ensure the generation of well-behaved parameter estimates introduced intractable challenges.

The starting point is an input–output representation of the plant model as in Equation 2. Recognizing that estimation and control are duals of each other (25), a similar nonminimal representation of the plant as in Equation 25 was used as the starting point to decouple the estimation of the state from the design of the control input. In particular, an AC input of the form

$$u(t) = \theta_c^{\mathrm{T}}(t)\omega(t) + k(t)r(t) \tag{28.}$$

enabled a tractable problem formulation, where $\omega(t)$ is generated as in Equation 25. The added advantage of the nonminimal representation is that it ensures the existence of a solution that matches the controlled plant using Equation 28 to that of the reference model. That is, the existence of a control parameter θ^* and k^* such that

$$u(t) = \theta^{*\mathrm{T}}\omega(t) + k^* r(t) \tag{29.}$$

ensured that the closed-loop transfer function from r to y matched that of a reference model with a transfer function $W_m(s)$, specified as

$$y_m(t) = W_m(s)r(t). \tag{30.}$$

That is, the controller in Equation 29 is guaranteed to exist for which the output error $e_y = y_p - y_m$ has a limiting property of $\lim_{t\to\infty} e_y(t) = 0$. For this purpose, the well-known Bézout identity (23) and the requirement that $W_p(s)$ has stable zeros were leveraged.

When the adaptive controller in Equation 28 is used, the plant model in Equation 2 and the existence of θ^* and k^* that guarantee that the output error $e_y(t)$ goes to zero lead to an error model of the form

$$e_y = (1/k^*)W_m(s)[\tilde{\theta}^{\mathrm{T}}\phi], \tag{31.}$$

where $\phi = [\omega^{\mathrm{T}}, r]^{\mathrm{T}}$ and $\tilde{\theta} = [(\theta - \theta^*)^{\mathrm{T}}, (k - k^*)]^{\mathrm{T}}$.

The problem of determining the adaptive rule for adjusting $\tilde{\theta}$ was solved in a very elegant manner when the relative degree (i.e., the net order) of $W_m(s)$ is equal to 1. In this case, a fundamental systems concept of a strictly positive real (SPR) transfer function as well as an elegant tool known as the Kalman–Yakubovich lemma (KYL) (13, 26–30) can be leveraged. The KYL, which was first proposed by Yakubovich (26) and then extended by Kalman (27), came out of stability theory of nonlinear systems, Popov's absolute stability, and the circle criterion (30) and is briefly described below.

The concept of positive realness arose in the context of stability of a class of linear systems with an algebraic nonlinearity in feedback. It was demonstrated, notably by Popov, that under certain conditions on the frequency response of the linear system, a Lyapunov function can be shown to exist. The KYL establishes the relation between these frequency domain conditions and the existence of the Lyapunov function (2).

Using the KYL, the following adaptive laws are proposed for the adjustment of the control parameters:

$$\dot{\theta} = -\mathrm{sign}(k^*)e_y\omega, \tag{32.}$$

$$\dot{k} = -\mathrm{sign}(k^*)e_y r. \tag{33.}$$

It can be shown that

$$V = e^{\mathrm{T}}Pe + (1/|k^*|)\left(||(\theta_c - \theta^*)||^2 + |(k - k^*)|^2\right) \qquad 34.$$

is a Lyapunov function where P is the solution of the KYL for the realization $\{A_m, b, c\}$ of $W_m(s)$, which is SPR. This follows from first noting that

$$\dot{V} = -e^{\mathrm{T}}(A_m^{\mathrm{T}}P + PA_m)e + 2e^{\mathrm{T}}Pb(\tilde{\theta}^{\mathrm{T}}\omega + \tilde{k}r) - \dot{\tilde{\theta}}^{\mathrm{T}}e_y\omega - \dot{\tilde{k}}e_y r.$$

Since $W_m(s)$ is SPR, the use of the KYL applied to $W_m(s)$ together with the adaptive laws in Equations 32–34 causes the second term to cancel out the third and fourth terms, and hence $\dot{V} = -e^{\mathrm{T}}Qe \leq 0$. The structure of the adaptive controller in Equation 28 guarantees that e_y, θ_c, k, ω, y_p, and u are bounded and that $\lim_{t\to\infty} e_y(t) = 0$. Additions of positive definite gain matrices to Equations 32 and 33 as in Equations 19 and 20 are straightforward. Similar to Section 3.1.2, the action–response–correction sequence is accomplished by u in Equation 28, e_y, and the adaptive laws in Equations 32 and 33.

The choice of the adaptive laws as in Equations 32 and 33 centrally depended on the KYL, which in turn required that $W_m(s)$ be SPR. An SPR transfer function (4) leads to the requirement that the relative degree—the difference between the number of poles and zeros of $W_p(s)$—is unity and has stable zeros (zeros only in $Re[s] < 0$), also defined as hyperminimum phase (31). Qualitatively, it implies that a stable adjustment rule for the parameter should be based on loss functions that do not significantly lag the times at which new data come into the system. For a general case when the relative degree of $W_p(s)$ exceeds unity, it poses a significant stability problem, as it is clear that the same simple adaptive laws as in Equations 32 and 33 will no longer suffice because the corresponding transfer function $W_m(s)$ of the reference model cannot be made SPR.

A final note regarding the assumptions made about the plant model in Equation 2 is in order. For the controller in Equation 29 to allow the closed-loop system to match the reference model in Equation 30 for any reference input $r(t)$, a reference model $W_m(s)$ with the same order and net order as that of $W_p(s)$ must be chosen, which implies that the order and net order of the plant must be known. Determination of a Lyapunov function requires that the sign of k^* be known. Finally, the model matching starting with a nonminimal representation of the plant requires stable pole-zero cancellations, which necessitates that the zeros be stable.

Extensions to a general case with output feedback have been proposed using several novel tools, including an augmented error approach (4), backstepping (8), averaging theory (32), and high-order tuners (33). In all cases, the complexity of the adaptive controller is increased, as the error model in Equation 31 does not permit the realizations of simple loss functions as in $L_i(\theta)$, $i = 1, 2$. Annaswamy & Fradkov (2) provided a concise presentation of these extensions.

3.1.5. Learning and imperfect learning. As is clear from all preceding discussions, the hallmark of all AC problems is the inclusion of a parameter estimation algorithm. In addition to ensuring that the closed-loop system is bounded and that the performance errors are brought to zero, all adaptive systems attempt to learn the underlying parameters, with the goal that the parameter error $\theta - \theta^*$ is reduced, if not brought to zero.

Four important implications of learning and imperfect learning should be kept in mind (and are expanded further in Section 5). The first is the necessary and sufficient condition under which learning occurs.

Definition 1 (4). A bounded function $\phi : [t_0, \infty) \to \mathbb{R}^N$ is persistently exciting if there exist $T > 0$ and $\alpha > 0$ such that

$$\int_t^{t+T} \phi(\tau)\phi^{\mathrm{T}}(\tau)\mathrm{d}\tau \geq \alpha I, \quad \forall t \geq t_0.$$

Morgan & Narendra (12) and Narendra & Annaswamy (4) showed that this condition leads to convergence of the parameter error in algebraic systems, in dynamic systems with states accessible, and in dynamic systems with output feedback. Several books and papers have delineated properties of the exogenous signals in a control system that ensure that the underlying regressor ϕ is persistently exciting (4, 7, 10, 11). It should be noted that this property creates a rank N matrix over an interval even though the integrand is of rank one at any instant τ. This necessary and sufficient condition on the underlying regressor leads to several desirable properties of the adaptive system, including lack of bursting (34–37) and uniform asymptotic stability and robustness to disturbances (38, 39).

The second implication is the important observation that persistent excitation is not required for satisfactory performance of the adaptive system; both output estimation and tracking, which are typical goals in system estimation and control, can be achieved without relying on learning. That is, a guaranteed safe behavior of the controlled system can be assured in real time even without reaching the learning goal, as output matching is an easier task, while parameter matching is task dependent and faces challenges due to spectral properties of a dynamic system. This guarantee in the presence of imperfect learning is essential and suggests that for real-time decision-making, control for learning is the practical goal, in contrast to learning for control. It should also be added that when the excitation level is insufficient or there is simply no persistent excitation, the parameters will not converge to the true values (40).

The third implication is the strong link between learning and robustness. Ensuring that the parameter estimates have converged to their true values opens the door to several attractive properties of a linear system, the foremost of which are exponential stability and robustness to various departures from idealization. In fact, a treatment of bounded behavior in the presence of persistent excitation has been established globally for the case when these departures are due to external disturbances (38) and locally for a larger class of perturbations (32). The foundation for these statements stems from the fact that AC systems are nonlinear, and bounded-input, bounded-output stability is not guaranteed when the unforced system is uniformly asymptotically stable and not exponentially stable (38, 41). Use of regularization and other modifications, such as projection and dead zone modifications, has been suggested to ensure robustness when no persistent excitation properties can be guaranteed (4, 6).

The fourth implication, which rounds off this topic, is this: When there is no persistent excitation and disturbances are present, the closed-loop system can produce large bursts of tracking error (34–36). That is, imperfect learning exhibits a clearly nonrobust property that leads to a significant departure from a tracking or regulation goal, exhibiting an undesirable behavior over short periods when the tracking error becomes significantly large. A specific example that illustrates this behavior is the following (34): Consider a first-order plant with two unknown parameters a and b of the form

$$y_{k+1} = ay_k + bu_k, \tag{35.}$$

whose AC solution is given by (42)

$$u_k = \frac{1}{\widehat{b}_k}\left[-\widehat{a}_k y_k + y^*_{k+1}\right]. \tag{36.}$$

The results of Goodwin et al. (43) can be used to reparameterize Equation 36 as

$$u_k = -\theta_{c_1,k} y_k + \theta_{c_2,k} y^*_{k+1} \tag{37.}$$

and propose parameter adjustment rules for $\theta_{c_1,k}$ and $\theta_{c_2,k}$. These adjustment rules guarantee that (*a*) $\theta_{c_i,k}$ and y_k are bounded (42); (*b*) $\theta_{c_i,k}$ converge to constants $\theta^0_{c_i}$, which may not coincide with the true values (44); and (*c*) y_k approaches y^*_k as $k \to \infty$ (42). In addition, when $\phi_{c,k}$ is persistently

exciting (i.e., satisfies Definition 1), we also have that the estimates $\theta_{c_i,k}$ approach the true values $\theta^*_{c_i}$. However, when such a persistent excitation is not present and when perturbations are present, bursting can occur, which can be explained as follows.

Suppose we consider a simple regulation problem with $y^*_k \equiv 1$. The control input in Equation 37 leads to a closed-loop system of the form

$$y_{k+1} = g(\theta_{c_1,k})y_k + h(\theta_{c_2,k}), \tag{38.}$$

where

$$g(\theta_{c_1,k}) = \left(a - b\theta_{c_1,k}\right), \qquad h(\theta_{c_2,k}) = b\theta_{c_2,k}. \tag{39.}$$

This implies that the closed-loop system is (*a*) unstable if $|g(\theta_{c_1,k})| > 1$ and (*b*) stable if $|g(\theta_{c_1,k})| < 1$. The most troublesome scenario occurs when there is marginal stability—i.e., $\theta_{c_1,k} = \theta^b_{c_1}$, where $g(\theta^b_{c_1}) = -1$. Suppose that that parameters $\theta_{c_i,k}$ become arbitrarily close to $\theta^b_{c_i}$ for some $k = k_0$; at k_0^+, a disturbance pulse is introduced, which can cause the parameters to drift, with $\theta_{c_1,k}$ approaching $\theta^b_{c_1}$, which in turn leads to oscillations y_k, causing $\theta_{c_i,k}$ to readjust and once again approach another set of constant values, $\theta^{0\prime}_{c_i}$. Such a phenomenon has been shown to occur in the absence of persistent excitation (34), including in continuous-time systems (37). This phenomenon is not peculiar to the specific systems in question and can occur in any dynamic system where simultaneous identification and control are attempted.

3.1.6. Numerical illustration of learning and imperfect learning. As an example, consider a continuous-time F-16 model (45), where the nominal dynamics are linearized about level flying at 500 feet/s at an altitude of 15,000 feet to produce a linear time-invariant system similar to the one described by Stevens & Lewis (45), with states, inputs, and parameters defined as follows:

$$x = \begin{bmatrix} \alpha \\ q \end{bmatrix}, \quad u = \delta_e, \quad A_p = \begin{bmatrix} -0.6398 & 0.9378 \\ -1.5679 & -0.8791 \end{bmatrix}, \quad b_p = \begin{bmatrix} -0.0777 \\ -6.5121 \end{bmatrix}, \tag{40.}$$

where α, q, and δ_e are the angle of attack (degrees), pitch rate (degrees per second), and elevator deflection (degrees), respectively. The goal is to control the nominal dynamics using a linear–quadratic regulator with cost matrices

$$Q_{LQR} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad R_{LQR} = 1. \tag{41.}$$

Solving the discrete algebraic Riccati equation provides us with a feedback gain vector

$$\theta_{LQR} = \left[-0.1536,\ 0.8512\right]^T. \tag{42.}$$

Finally, applying the feedback gain above to the dynamics in Equation 40, along with a reference input $r(t)$, gives the following closed-loop system, which we choose as the reference system:

$$\dot{x}_m = A_p x_m + b_p(\theta^T_{LQR} x_m + r) = A_h x_m + b_p r. \tag{43.}$$

Assuming that both states are measurable, the AC goal is to ensure that the plant tracks the reference system for a given reference signal $r(t)$, regardless of differences between the plant and the nominal dynamics.

We now introduce a parametric perturbation into the plant model such that the true open-loop dynamics are given by

$$\dot{x} = \bar{A}_p x + b_p u, \tag{44.}$$

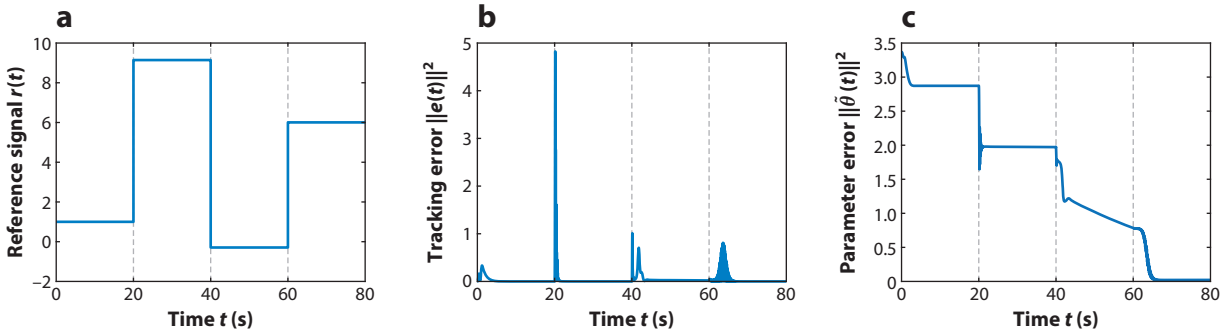

Figure 1

Simulation results of a simple F-16 model with imperfect learning. (*a*) The reference signal $r(t)$ to be followed by the reference model and the plant. Within any given 20-s period, this reference signal does not provide enough excitation for the adaptive system to fully learn. (*b*) The tracking error $\|e(t)\|^2$. The tracking error goes to zero within the first 20-s period, but bursting occurs every time the reference signal subsequently changes. (*c*) The parameter error $\|\tilde{\theta}(t)\|^2$. The adaptive system fully learns the true parameters by the end of the simulation, but while the parameters are not fully learned, bursting occurs.

where

$$\bar{A}_p = \begin{bmatrix} -0.5078 & 0.8839 \\ 9.4950 & -5.3970 \end{bmatrix}. \qquad 45.$$

One can verify that $\bar{A}_p$, b_p, and A_{h} satisfy the matching condition in Assumption 1 for some θ_* and $k^* = 1$. Because the perturbation in the plant's dynamics is unknown to the control designer, we choose an initial parameter estimate of $\theta(0) = \theta_{\mathrm{LQR}}$. We assume $x(0) = 0$. The resulting closed-loop adaptive system with Equations 18, 19, 43, and 44 is simulated. $k(t)$ is set to 1, and Γ_θ is chosen as the identity matrix. The responses are shown in **Figures 1** and **2**. In **Figure 1*b*,*c***, the tracking error $\|x(t) - x_m(t)\|^2$ and the parameter error norm $\|\theta(t) - \theta^*\|^2$ are shown for the reference signal $r(t)$ shown in **Figure 1*a***, where $e(t) = x(t) - x_m(t)$ and $\tilde{\theta}(t) = \theta(t) - \theta^*$. The same is illustrated in **Figure 2*a*–*c*** as well.

Figure 1 is an illustration of imperfect learning: Although the tracking error in **Figure 1*b*** has gone completely to zero at the end of the initial 20-s interval, the parameter error in **Figure 1*c***

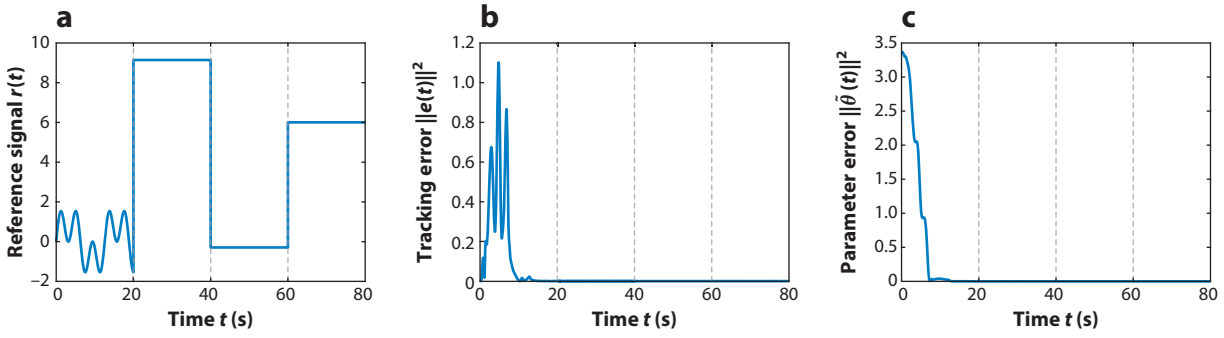

Figure 2

Simulation results of a simple F-16 model with full learning. (*a*) The reference signal $r(t)$. In this simulation, the reference signal is modified during the first 20-s period to provide sufficient excitation for full learning. (*b*) The tracking error $\|e(t)\|^2$. Due to full learning during the first 20 s, the plant tracks the reference model in Equation 43 through subsequent changes in $r(t)$ without bursting. (*c*) The parameter error $\|\tilde{\theta}(t)\|^2$. Due to persistent excitation within the first 20-s time window, the adaptive system quickly learns the true parameters.

remains nonzero due to lack of excitation in the reference input $r(t)$ (**Figure 1*a***). As a result, each time $r(t)$ subsequently changes, which could occur at any instant, the plant and reference model initially move in different directions. The bursting phenomenon is therefore apparent in **Figure 1*b*** because each time $r(t)$ changes, the tracking error explodes to large nonzero values before the adaptive law is able to correct this behavior. Contrast **Figure 1** with **Figure 2**, in which the reference system is set to $r(t) = \sin(0.5t) + \sin(1.5t)$ during the first 20-s interval to provide the plant with persistent excitation and allow the parameter error to go to zero alongside the tracking error, as evidenced by **Figure 2*c***. Learning occurs in this second simulation, and thus no bursting is exhibited in **Figure 2*b***.

This simple example uses a piecewise constant reference input with large jumps to illustrate the danger of imperfect learning. In practice, a control designer might not choose such an adversarial reference input. However, the same phenomenon can be caused by several other potentially adversarial influences on the system that are out of the control designer's hands, such as state-dependent disturbances due to unmodeled dynamics (46).

3.2. Nonlinear Systems

We now return to the original problem shown in Equation 1, where we assume that the parameter θ is unknown. Assuming that the adaptive controller is determined as in Equations 4 and 5, the question is whether one can determine conditions on the overall closed-loop system determined by the plant and the controller under which the control objective shown in Equation 3 can be guaranteed.

An elegant AC solution for a large class of nonlinear systems with guarantees of global stability can be attributed to what is denoted as a backstepping approach (8, 47, 48). The main feature of this class of nonlinearities is a triangular structure, with a typical dynamics of the form (48)

$$\dot{z}_i = \gamma_i^0(z_1, \ldots, z_{i+1}) + \theta^{\mathrm{T}}\gamma_i(z_1, \ldots z_{i+1}), \quad i = 1, \ldots n-1, \qquad 46.$$

$$\dot{z}_n = \gamma_n^0(z) + \theta^{\mathrm{T}}\gamma_n(z) + \left[\beta_0(z) + \theta^{\mathrm{T}}\beta(z)\right]u, \qquad 47.$$

where z_i denotes the ith element of z, and the functions γ_i, β_i, and β are all known, with θ as an unknown parameter, all with suitable dimensions. The backstepping approach involves the construction of a suitable Lyapunov function that allows a stable adaptive controller even though a certainty-equivalence-based approach does not readily lead to a suitable structure and overcomes the fact that the triangular structure does not satisfy a matching condition.

Several approaches to AC of nonlinear systems are based on approximation of nonlinear right-hand sides by linear ones. There are only a few publications with explicit formulations of dynamic properties of the overall system, e.g., a paper by Wen & Hill (49) where the nonlinear model is reduced by standard linearization via finite differences. There are a few results dealing with high-gain linear controllers for nonlinear systems (50, 51). Other tools, such as absolute stability (52, 53), passivity (54), passification (55–57), and immersion and invariance (58), have led to a successful set of approaches for AC of nonlinear systems.

An additional approach that requires special mention is AC of nonlinear systems based on neural networks. The basic principles of using neural networks in control of nonlinear systems have been addressed in a number of papers (59–62), and this approach is one of the common tools employed in both AC and RL due to neural networks' ability to approximate complex nonlinear maps and powerful interpolation properties. Another approach that has been used is an approximation of Lyapunov functions for control using neural networks, so that the stability of the closed loop (63, 64) is guaranteed. The central challenge addressed in all of these works is to come up with a stable approach that addresses the well-known fact that there is an underlying issue of an

approximation error that is a function of the compact set over which the neural network is trained. These problems are difficult to overcome because the goal is to assume that uncertainties can occur even after training has been completed, and the task at hand is the determination of a real-time control input that is guaranteed to be well behaved.

4. REINFORCEMENT LEARNING: PROBLEM STATEMENT AND APPROACH

In contrast to AC, whose evolution has been motivated by stability considerations, RL has been strongly influenced by notions of optimality, finite states, and dynamic programming. To provide a narrative comparable to that in the previous section, we begin with a deterministic, nonlinear, discrete-time dynamic system of the form

$$x_{k+1} = f(x_k, u_k). \tag{48.}$$

It should be noted, however, that a large part of the RL literature has focused on a stochastic treatment wherein principles of optimality as well as the entire approach outlined here have clear analogs.

The goal is to design a control input of the form

$$u_k = \pi(x_k) \tag{49.}$$

so that an underlying cost $J_\pi(x_0)$ is minimized, where

$$J_\pi(x_0) = \lim_{k\to\infty} \frac{1}{k} \sum_{i=0}^{k} c(x_i, \pi(x_i)), \qquad x_0 \in X, \tag{50.}$$

and π is a deterministic policy that maps $x \in X$ to $u \in U(x)$. $X \subset \mathbb{R}^n$ and $U \subset \mathbb{R}^m$ are desired sets for the state x and the control input u, respectively. It is assumed that policies $\pi(x_k)$ can be found such that

$$0 \le c(x_k, u_k) \le \infty, \qquad \forall x_k \in X, u_k \in U(x_k). \tag{51.}$$

With the above problem statement, optimal control results provide a framework for determining the solutions of Equation 50. The foundation for this framework comes from the well-known Bellman equation:

$$J^*(x) = \min_{u\in U(x)} \{c(x,u) + J^*(f(x,u))\}, \qquad \forall x \in X, \tag{52.}$$

in which the fixed point solution J^* is the optimal cost. The solution leads to an optimal control input $u^*(x_k)$ that satisfies

$$u^*(x) = \arg\min_{u\in U(x)} \{c(x,u) + J^*(f(x,u))\}, \qquad \forall x \in X. \tag{53.}$$

The dynamic programming approach utilizes the principle of optimality in which the following cost-to-go problem is solved:

$$J^*(x_k) = \min_{u_k\in U_k} \{c(x_k, u_k) + J^*(x_{k+1})\}, \tag{54.}$$

with a boundary condition $J^*(x_\infty)$ specified. The challenge in solving the Bellman equation comes from its computational burden, especially when the underlying dimensions of x and u as well as the sets U and X are large and when f is unknown.

Motivated by these concerns of computational burden and overall complexity in determining the optimal control input when the model is uncertain, an approximation is employed to solve the

Bellman equation and forms the subject matter of the field of RL, often denoted as approximate dynamic (or sometimes neuro-dynamic) programming. The evolution of this field centrally involves the determination of a suitable approximate cost and a corresponding approximately optimal control policy. Various iterative approaches, including policy iteration, *Q*-learning, and value iteration, have been proposed in the literature to determine the best approximation. Bertsekas (65, 66) and Watkins & Dayan (67) have published excellent expositions on *Q*-learning and value iteration. A brief discussion of policy iteration is given below.

Here, policy iteration is illustrated using the concept of a *Q*-function. The optimal *Q*-function corresponds to the cost stemming from the current control action, assuming that all future costs will be optimized using an optimal policy. Such a function is defined as the solution to

$$Q^*(x_k, u_k) = c(x_k, u_k) + \min_{u \in U}, Q^*(f(x_k, u_k), u). \quad 55.$$

An attractive property of a *Q*-function is that it provides a model-free method for determining the optimal control action, in contrast to Equation 53, which requires f. That is,

$$u^*(x) = \arg\min_{u \in U(x)} Q^*(x, u). \quad 56.$$

To construct an approximation $\hat{Q}_i$ to Q^*, the following iterative algorithm is used:

$$\widehat{Q}_{i+1}(x, u) = c(x, u) + \min_{a \in U} \widehat{Q}_i(f(x, u), a). \quad 57.$$

An implicit assumption here is that the cost $c(x, u)$ can be determined in Equation 57, even when the model is unknown, using the concept of an oracle (68). As $\widehat{Q}_i \to Q^*$, it follows that the corresponding input from Equation 56 will yield the optimal u^*.

To determine an efficient approximation, a parametric approach is often used. Denoting this parameter as $\theta \in \mathbb{R}^p$, we estimate the *Q*-function as

$$\widehat{Q}_\theta(x_k, u_k) = \sum_{i=1}^{p} \phi_i(x_k, u_k)\theta_i, \quad 58.$$

where $\phi(x, u)$ is a basis function in $\mathbb{R}^p$ for the *Q*-function. One can then associate a parameter θ_π for a particular policy $u_k = \pi(x_k)$ by defining

$$\widehat{Q}_{\theta_\pi}(x_k, u_k) = \phi^{\mathrm{T}}(x_k, u_k)\theta_\pi. \quad 59.$$

A particularly successful approximation involves deep neural networks of the form

$$\widehat{Q}_{\theta_\pi}(x_k, u_k) = g(x_k, u_k, \theta_\pi), \quad 60.$$

where g is a nonlinear function of x_k and u_k as well as parameters θ_π, which represent the weights of the neural network. This ability to approximate even complex functions has been successfully leveraged in AC as well (59, 60, 62, 69). The structure of the network g often permits a powerful approximation and leads to a desired approximation $\hat{Q}^*_\theta$ with a minimal approximation error. By collecting several samples $\{x_{k_j}, u_{k_j}, c_{k_j}\}, j = 1, \ldots, N$ subject to the policy $u_{k_j} = \pi(x_{k_j})$, one can compute a least-squares-based solution (70, 71) to compute θ_π. One can also use a recursive approach to determine these parameters, based on a gradient descent rule that minimizes a loss function of the approximation error, an example of which is the well-known back-propagation approach (72, 73). A large number of variations occur in the type of adjustments used in determining θ_π (74–76), motivated by performance, efficiency, and robustness, an exposition of which is beyond the scope of this article. Once θ_π is determined, an approximate optimal policy is determined as

$$\hat{u}(x) = \arg\min_{u \in U(x)} \widehat{Q}(x, u) = \arg\min_{u \in U(x)} g(x_k, u_k, \theta_\pi). \quad 61.$$

It should be noted that the same action–response–correction sequence introduced in Section 3 occurs in RL. This follows from Equation 61, which determines the action; $\hat{Q}_{\theta_\pi}$ and ϕ, which constitute the response; and Equation 57 together with the underlying gradient-descent-based neural network update, which represents the correction.

This iterative approach for optimizing the policy and Q-function is predicated on the ability to interrogate the system through simulations and collect the responses and costs. As these computations are often carried out offline, real-time performance metrics, such as stability, are not of concern. The focus of the dynamic programming approach, approximate or otherwise, is on optimality and not stability.

When the problem at hand shifts to that of real-time control, and when the dynamic system under consideration has uncertainties, the RL approach begins to get tested. Over the past several years, the scope of RL has been increasingly used to not only learn the optimal policy through an approximate structure but also carry out this learning when the dynamic system is uncertain. And it is in the context of the latter problem that the commonality between AC and RL begins to emerge. As the RL approach is predicated on access to a simulation engine or dataset that enables repeated exploration of various policies, one of the major uncertainties that must be contended with is the oft-mentioned challenge of a sim-to-real gap (77, 78), which describes the difficulty of generalizing a trained policy to a new environment. This challenge takes even more of a center stage in the context of real-time control.

Three points should be noted in particular. First, RL algorithms, whether based on nonrecursive or recursive approaches, are geared toward ensuring that the function approximations (e.g., $\hat{Q}$), and not the parameter estimates of θ_π, converge to their true values. It should be noted that when the dimensions of θ_π are large, as in deep networks, the focus is exclusively on the optimization of the underlying Q-function, value function, or policy. Any such overparameterization often negates identifiability and can lead to imperfect learning, as there can be infinite $\hat{\theta}$ that solves

$$\hat{Q}_{\hat{\theta}_{\hat{\pi}}}(x, u) = Q_{\theta_\pi}(x, u).$$

Second, the parametric updates, whether using least squares or recursive counterparts, are exclusively an offline exercise. As we move the focus of the RL methods toward an online solution, a huge set of obstacles that were mentioned in the previous section will have to be addressed here. Imperfect learning can often occur because of a lack of identifiability or lack of convergence. It is not clear that robustness properties will always be satisfied or that bursting can be avoided. Third, the approximation error in Q is a function of the input u and the state x. This in turn implies once again that in real time, any perturbations that occur due to departures from the simulation environment, because of unforeseen anomalies, environmental changes, or modeling errors, may lead to fundamental questions of stability and robustness.

5. ILLUSTRATIONS OF ADAPTIVE CONTROL AND REINFORCEMENT LEARNING

To better elucidate the two approaches of AC and RL, this section describes two specific examples—a linear and a nonlinear dynamic system—and delineates the two approaches.

5.1. Example 1: Control of a Linear Discrete-Time Plant

A typical problem formulation in this class is of the form (79)

$$x_{k+1} = Ax_k + Bu_k, \qquad 62.$$

where A and B are unknown matrices. The control objective is to determine u_k such that the cost function

$$J(A,B) \stackrel{\text{def}}{=} \limsup_{T\to\infty} \frac{1}{T} \sum_{i=1}^{T} \left[x_i^{\mathrm{T}} Q x_i + u_i^{\mathrm{T}} R u_i\right], \qquad 63.$$

where $Q = Q^{\mathrm{T}} > 0$ and $R = R^{\mathrm{T}} > 0$, is minimized. To address this objective, the following first describes the AC approach (both direct and indirect) and then the RL approach.

5.1.1. Adaptive control approach. In what follows, we outline two different approaches, indirect and direct, where in the former, the plant parameters are first estimated and then the control parameters are determined, whereas in the latter, the control parameters are directly estimated.

5.1.1.1. Indirect approach. It is well known that for this linear–quadratic system, the following control input is optimal:

$$u_k = K(A,B)x_k, \qquad 64.$$

where

$$K(A,B) = -[B^{\mathrm{T}} P B + R]^{-1} B^{\mathrm{T}} P A$$

and P solves the discrete-time algebraic Riccati equation

$$P = A^{\mathrm{T}} P A - A^{\mathrm{T}} P B (B^{\mathrm{T}} P B + R)^{-1} B^{\mathrm{T}} P A + Q.$$

The results of Campi & Kumar (79) show that the problem becomes significantly more complex when A and B are unknown, and the control gain in Equation 64 must be replaced with one that depends on parameter estimates of A and B. Suppose we define $(A_k^{\mathrm{LS}}, B_k^{\mathrm{LS}})$ as the least-squares estimate of $[A, B]$, i.e.,

$$(A_k^{\mathrm{LS}}, B_k^{\mathrm{LS}}) \stackrel{\text{def}}{=} \operatorname{argmin}_{(A,B)\in\Theta} \sum_{s=1}^{k} ||x_s - A x_{s-1} - B u_{s-1}||^2. \qquad 65.$$

Becker et al. (44) showed that for autoregressive–moving-average with exogenous inputs (ARMAX) systems, which are equivalent representations of the plant in Equation 62, the parameter estimates can converge to false values with positive probabilities when measurement noise is present in Equation 62; Borkar & Varaiya (80) provided an example of the above statement for general Markov chains. Campi & Kumar (79, 81) reported an interesting fix for this problem that enabled a suboptimal solution. The following assumptions are made.

Assumption 2. The true value (A^0, B^0) lies in the interior of a known compact set Θ_0, and (A, B) is stabilizable at all points in this compact set Θ_0.

The approach used by Campi & Kumar (79) consists of adding a bias term to the cost J in Equation 63 so as to lead to estimates of the form

$$(A_k^{\mathrm{LS}}, B_k^{\mathrm{LS}}) = \operatorname{argmin}_{(A,B)\in\Theta} \sum_{s=1}^{k} ||x_s - A x_{s-1} - B u_{s-1}||^2 + \mu_k J(A,B) \qquad \text{if } k \text{ is even} \qquad 66.$$

$$= (\widehat{A}_{k-1}, \widehat{B}_{k-1}) \qquad \text{if } k \text{ is odd.} \qquad 67.$$

In the above, μ_k is a deterministic sequence that tends to infinity as $o(\log k)$. The points to note here are that (*a*) the estimation is based on a nonrecursive approach, (*b*) the problem is strictly focused on a stabilization task, and (*c*) the solution exploits the linear dynamics and the quadratic structure

of the cost. The typical procedure in the indirect approach is to utilize the updated estimates of A and B to determine a controller using Equation 64 so as to render the control input optimal. This often requires sufficient persistent excitation to precede the control computation, which may not always be possible.

In addition to the above, Guo and colleagues (82–84) used diminished persistent excitation with time to lead to adaptive optimal control in stochastic systems. Elements of the approach used by Campi & Kumar (79) have also been employed by Dean et al. (85) and Abbasi-Yadkori & Szepesvári (86) to derive nonasymptotic bounds with the requirement that the estimation error in A become arbitrarily small for optimal control to be guaranteed, with persistent excitation introduced through the injection of noise. Here, too, the optimization cost centers around a stabilization task rather than tracking; the latter can make all associated derivations significantly more challenging, as there is a potential to lead to imperfect learning and therefore bursting.

5.1.1.2. Direct approach. Unlike the previous case, in a direct approach the control parameters themselves are directly estimated. To describe this method, we start with a single-input version of Equation 62, rewritten as

$$x_{k+1} = Ax_k + bu_k, \tag{68.}$$

where for ease of exposition only A is assumed to be unknown. Suppose that a nominal value of A is given by A_m, the reference system is designed as in Section 3.1.2 as

$$x_{m(k+1)} = A_m x_{mk} + bu_{mk}, \tag{69.}$$

and u_{mk} is designed to optimize a cost function

$$J(A_m, b) \stackrel{\text{def}}{=} \limsup_{T\to\infty} \frac{1}{T} \sum_{i=1}^{T} \left[x_{mi}^{\mathrm{T}} Q x_{mi} + u_{mi}^{\mathrm{T}} R u_{mi}\right]. \tag{70.}$$

Noting that the cost function is quadratic and the dynamics in Equation 69 is linear, the optimal control input is given by $u_{mk} = k_{\mathrm{opt}}^{\mathrm{T}} x_{mk}$, which leads to a reference system $x_{m(k+1)} = A_{\mathrm{h}} x_{mk}$, where A_{h} is Schur-stable and is such that it leads to an optimal cost of J. We assume that (A, b) satisfies Assumption 1, with Equation 16 satisfied for $k^* = 1$. The structure of the optimal input u_{mk} suggests that as θ^* is not known, an AC input is chosen as

$$u_k = \theta_k^{\mathrm{T}} x_{pk}, \tag{71.}$$

which leads to closed-loop adaptive system

$$x_{k+1} = (A_{\mathrm{h}} + b\tilde{\theta}_k^{\mathrm{T}}) x_k, \tag{72.}$$

where $\tilde{\theta}_k = \theta_k - \theta^*$. To drive parameter error $\tilde{\theta}_k$, we attempt to drive e_k to zero, where $e_k = x_{pk} - x_{mk}$.

As in Section 3.1.2, we derive an error model that relates e_k to $\tilde{\theta}_k$, which takes the form

$$e_{(k+1)} = A_{\mathrm{h}} e_k + b\tilde{\theta}_k^{\mathrm{T}} x_k. \tag{73.}$$

The question then is whether it is possible to adjust the parameter estimate as

$$\theta_{k+1} = \theta_k - \gamma g(e_k, \theta_k) \tag{74.}$$

using a suitable gradient $g(e_k, \theta_k)$ that ensures stability, convergence of e_k to zero, and optimality.

The following constants, both scalars and matrices, are first defined:

$$P = P^{\mathrm{T}} > 0 \text{ such that } A_{\mathrm{h}}^{\mathrm{T}} P A_{\mathrm{h}} - P = -Q, \qquad Q = Q^{\mathrm{T}} > 0, \tag{75.}$$

$$c = 2A_{\mathrm{h}}^{\mathrm{T}} P b, \qquad d > b^{\mathrm{T}} P b, \qquad d_0 = \frac{d}{b^{\mathrm{T}} P b}. \tag{76.}$$

Next, we define a few variables:

$$N(x) = \frac{1}{1 + \alpha\gamma d x^{\mathrm{T}} x}, \tag{77.}$$

$$\omega_{k+1} = A_{\mathrm{h}}\omega_k + b\,\alpha\gamma\, x_k^{\mathrm{T}} x_k \epsilon_{yk}, \tag{78.}$$

$$\epsilon_k = e_k - \omega_k, \tag{79.}$$

$$\epsilon_{yk} = N(x_k)\left(c^{\mathrm{T}}\epsilon_k + d_0 b^{\mathrm{T}} P(e_{k+1} - A_{\mathrm{h}} e_k)\right). \tag{80.}$$

In the above, Equation 75 is the Lyapunov equation for discrete linear time-invariant systems, yielding a positive definite solution P; d_0 is a positive constant that exceeds unity and is useful in developing the update law for θ_k; c determines a unique combination of a vector of state errors and plays a central role that will become clear below; and ω_k and ϵ_k (and therefore ϵ_{yk}) are augmented state and output errors, respectively, that lead to a provably correct adaptive law that is in lieu of that in Equation 73, as will be seen below.

With the above variables and constants, we choose the adaptive law (87)

$$\theta_{k+1} = \theta_k - \gamma \epsilon_{yk} x_k. \tag{81.}$$

What is remarkable about the choice of the gradient $g(\cdot, \cdot)$ as $\epsilon_{yk} x_k$ is that it causes the positive definite function

$$V_k = 2\epsilon_k^{\mathrm{T}} P \epsilon_k + \frac{1}{\gamma}\tilde{\theta}_k^{\mathrm{T}}\tilde{\theta}_k \tag{82.}$$

to become a Lyapunov function, i.e., $\Delta V_k \leq 0$. To provide a rationale for the choice of V in Equation 82, the following theorem is needed.

Theorem 1 (87). A dynamic system given by

$$\epsilon_{k+1} = A_{\mathrm{h}}\epsilon_k + b\nu_k, \qquad \epsilon_{yk} = c^{\mathrm{T}}\epsilon_k + d\nu_k, \tag{83.}$$

$$\nu_k = \tilde{\theta}_k^{\mathrm{T}} x_k - \alpha\gamma x_k^{\mathrm{T}} x_k \epsilon_{yk}, \tag{84.}$$

together with the adaptive law in Equation 81, permits the Lyapunov function in Equation 82 with a nonpositive decrease

$$\Delta V_k = -2\epsilon_k^{\mathrm{T}} Q \epsilon_k - 2(d - b^{\mathrm{T}} P b)\nu_k^2 - (2\alpha - 1)\gamma \epsilon_{yk}^2 x_k^{\mathrm{T}} x_k. \tag{85.}$$

The crucial point to note here is that the augmented state error ϵ_k and ϵ_{yk} in Equations 79 and 80 can be shown to satisfy the dynamic relations in Equation 83. Thanks to Theorem 1 (87), we have therefore identified a provably correct law (Equation 81) for the control parameter θ_k in Equation 71. This leads to the complete solution using the direct AC approach, given by Equations 71 and 81.

Several properties follow from Theorem 1. First, θ_k and ϵ_k are bounded. Second, as $k \to \infty$, ϵ_k, ν_k, and $\epsilon_{yk} x_k$ all approach zero. Third, because $\nu_k = \tilde{\theta}_k^{\mathrm{T}} x_k - \alpha\gamma x_k^{\mathrm{T}} x_k \epsilon_{yk}$, it follows that $\tilde{\theta}_k^{\mathrm{T}} x_k$ approaches zero asymptotically. Fourth, the true state error e_k approaches zero asymptotically, as A_{h} is Schur-stable. And fifth, all variables in the closed-loop system are bounded. The dynamic model in Equations 83 and 84 can be viewed as an SPR transfer function between ν_k and ϵ_{yk}. The normalization $N(x)$ as in Equation 77 and the choice of ϵ_{yk} as in Equation 80 were necessary to create such an SPR operator, which clearly adds to the complexity of the underlying solution.

5.1.2. A comparison of direct and indirect approaches. A few obvious distinctions are apparent from the discussions in the preceding sections. While the indirect approach is motivated by optimality and the direct approach is motivated by stability, the assumptions made—Assumption 2 in the indirect approach and Assumption 1 in the direct approach—make the class of systems in question quite comparable. Assumption 2 requires controllability in the entire compact set, which may not be satisfied by systems where Assumption 1 does not hold. For ease of exposition, noise has not been included, and only a single-input case has been considered in the discussions in Section 5.1.1.2. Extensions were reported by Annaswamy & Fradkov (2 and references therein).

The difference between the two approaches, however, becomes more pronounced as one moves toward the tracking problem. It is not easy to ensure a stable controller with the indirect approach unless the parameter estimation error becomes arbitrarily small, which in turn makes the dependence on persistent excitation a strong one; in this regard, a direct approach is more robust, as imperfect learning is implicitly accounted for in its formulation. Its development, however, entails more complexity because the algorithm requires an appropriate gradient function $g(\cdot)$ that leverages notions of SPR transfer functions.

5.1.3. Reinforcement learning approach. Note that the RL approach outlined in Section 4, particularly in Equations 59 and 61, is directly applicable to an optimal choice of u_k in Equation 62 by replacing the right-hand side of Equation 50 with that of Equation 63. Several variations of this approach have been suggested in the past few decades, as mentioned in Section 4.

5.2. Example 2: Control of a Continuous-Time Nonlinear Plant

The problem that we consider is a nonlinear plant of the form

$$\dot{x} = f(x) + g(x)u, \tag{86.}$$

where there are some uncertainties, which may be either in f or in both f and g. The goal is to choose u so as to minimize a cost function

$$J(x_0, u) = \int_0^\infty c(x(t), u(t))\mathrm{d}t, \quad x(0) = x_0. \tag{87.}$$

5.2.1. Adaptive control approach. We start with the case when the dynamic system in Equation 86 is affine and controllable, where $f(x)$ is assumed to be unknown, and $g(x) = B$. Without loss of generality, it is assumed that $f(0) = 0$. With this assumption, we rewrite Equation 86 as

$$\dot{x} = Ax + B[u + f_1(x)], \tag{88.}$$

where A and B denote the Jacobian evaluated at the equilibrium point $x = 0$, and (A, B) is a controllable pair. The uncertainty in f and g in Equation 86 can be assumed to pertain to A, B, and $f_1(\cdot)$.

Similar to Example 1, we propose a reference system

$$\dot{x}_\mathrm{r} = A_\mathrm{r}x_\mathrm{r} + B_\mathrm{r}[u_\mathrm{r} + f_{1\mathrm{r}}(x_\mathrm{r})] \tag{89.}$$

where $A_\mathrm{r}, B_\mathrm{r}$, and $f_{1\mathrm{r}}(x)$ can be viewed as nominal values of the matrices A and B and the nonlinearity $f_1(x)$. Suppose that a nominal controller is designed as

$$u_\mathrm{r} = -f_{1\mathrm{r}}(x_\mathrm{r}) + \Theta_{l,\mathrm{r}}x_\mathrm{r} + u_\mathrm{com}, \tag{90.}$$

where $\Theta_{l,\mathrm{r}}$ is such that $A_\mathrm{r} + B_\mathrm{r}\Theta_{l,\mathrm{r}} = A_\mathrm{H}$, with A_H a Hurwitz matrix, and u_com is such that $x_\mathrm{r}(t)$ tracks a desired command signal $x_\mathrm{com}(t)$ as closely as possible and such that the cost function in Equation 87 corresponding to x_r and u_com is minimized.

Suppose that the plant dynamics in Equation 88 had uncertainties that satisfy the following assumptions.

Assumption 3. The nonlinearity $f_l(x)$ is equal to $\Theta_n' \Phi_n(x)$, where $\Theta_n' \in \mathbb{R}^{m \times l}$ is unknown, while $\Phi_n(x) \in \mathbb{R}^l$ is a known nonlinearity.

Assumption 4. The unknown linear parameters A and B are such that a matrix $\Theta^* \in \mathbb{R}^{m \times m}$ exists such that

$$A + B\Theta^* = A_{\mathrm{H}} \tag{91.}$$

and a diagonal matrix Λ exists, with known signs for all diagonal entries, such that

$$B = B_{\mathrm{r}} \Lambda. \tag{92.}$$

The plant equation in Equation 88 then becomes

$$\dot{x} = Ax + B_{\mathrm{r}} \Lambda \left[u + \Lambda^{-1} \Theta_n' \Phi_n(x) \right], \tag{93.}$$

where the uncertainties in the plant to be controlled are lumped into the matrices $\Lambda \in \mathbb{R}^{m \times m}$ and $\Theta_n' \in \mathbb{R}^{m \times l}$. The structure of the uncertainty Λ in Equation 92 often occurs in many practical applications in the form of a loss of control effectiveness. This is typically due to unforeseen anomalies that may occur in real time, such as accidents or aging in system components, especially in actuators. Parametric uncertainty Θ_n' in the nonlinearity $f_l(x)$ may be due to modeling errors. As nonlinearities are always more difficult to model even with system identification, it may not always be possible to accurately identify the parameters of nonlinear effects even if the underlying mechanisms may be known; this provides the rationale for Assumption 3. The problem here is therefore control of Equation 93, where B_{r} and $\Phi_n(x)$ are known but A, Λ, and Θ_n' are unknown. Overall, the model structure in Equation 93 is utilized to develop the AC solution.

The adaptive controller is chosen as follows:

$$u = \widehat{\Theta}(t) \Phi(t), \tag{94.}$$

$$\dot{\widehat{\Theta}} = -\gamma B^{\mathrm{T}} P e \Phi^{\mathrm{T}}, \qquad e = x - x_{\mathrm{r}}, \tag{95.}$$

where $\widehat{\Theta}$ is a parameter estimate of Θ,

$$\Theta := \left[\Lambda^{-1},\ -\Lambda^{-1} \Theta_n',\ \Theta^* \right], \qquad \Phi := \begin{bmatrix} u_{\mathrm{com}} \\ \Phi_n(x) \\ x \end{bmatrix}. \tag{96.}$$

The efforts in AC guarantee that the closed-loop system determined by Equations 93–95 is globally stable for any initial conditions in $x(0)$, $x_{\mathrm{r}}(0)$, and $\widehat{\Theta}(0)$.

This follows by deriving an error model that relates e and the parameter error $\widetilde{\Theta} = \widehat{\Theta} - \Theta$, which takes the form

$$\dot{e} = A_{\mathrm{H}} e + B_{\mathrm{r}} \Lambda [u - \Theta \Phi]. \tag{97.}$$

That is, the key component that connects the uncertain parameter Θ to the performance error e is a regressor Φ. Together with the AC input as in Equation 94, we get a fundamental error model of the form

$$\dot{e} = A_{\mathrm{H}} e + B_{\mathrm{r}} \Lambda [\widetilde{\Theta} \Phi]. \tag{98.}$$

The following comments can be made regarding the choice of the adaptive controller and the behavior of the closed-loop system.

1. Three different elements are employed in the regressor Φ: $u_{\text{com}}(t)$ in Equation 90, $\Phi_n(x(t))$, and the state x. These are utilized to address the three different sources of parametric uncertainties, Λ, Θ_n, and Θ^*. The first regressor component, $u_{\text{com}}(t)$, comes predominantly from the reference system. It is assumed that sufficient information is available about the nominal system to be controlled and the desired command $x_{\text{com}}(t)$ so as to generate $u_{\text{com}}(t)$ at each t. The second and third regressors are determined by the linear and nonlinear aspects of the plant dynamics, which is assumed to be available based on the physics of the system. Together, these regressors lead to an error model structure as shown in Equation 98. Both this error model and a real-time performance metric such as a loss function or a Lyapunov function are used to determine the parameter learning algorithm in Equation 95.
2. It can be shown that $V = e^{\mathrm{T}}Pe + \mathrm{Tr}(\widetilde{\Theta}^{\mathrm{T}}(\Lambda^{\mathrm{T}}S)\widetilde{\Theta})$ is a Lyapunov function for the error system specified by Equations 95 and 98, where $S = \Gamma^{-1}$, with the symmetric part of ΛS positive definite.
3. There are several extensions of the AC approach to broader nonlinear systems, as in Equations 46 and 47 (2, 8).

5.2.2. Reinforcement learning approach. One particular application of RL to control the system in Equation 86 begins with the following assumption (88, 89).

Assumption 5. There exist a function $V_0 \in \mathcal{P}$ and a feedback control policy u_1 such that

$$\mathcal{L}(V_0(x), u_1(x)) \geq 0, \quad \forall x \in \mathbb{R}^n, \tag{99.}$$

where, for any $V \in \mathcal{C}^1$ and $u \in \mathbb{R}^m$,

$$\mathcal{L}(V, u) = -\nabla V^{\mathrm{T}}(x)(f(x) + g(x)u) + c(x, u), \tag{100.}$$

where V_0 is denoted as the value function.

It is clear that Assumption 5 assumes that despite the uncertainty in Equation 86, a stabilizing policy $u_1(x)$ can be found for some V_0. It should also be noted that V_0 serves as a Lyapunov function for this system. It then follows that V_0 could be used as an upper bound for the cost incurred using this stabilizing policy—that is,

$$J(x_0, u_1) \leq V_0(x_0), \quad \forall x_0 \in \mathbb{R}^n. \tag{101.}$$

This stability assumption is then connected with optimality through the Hamilton–Jacobi–Bellman equation, necessitating the following assumption.

Assumption 6. There exists a proper, positive definite, and continuously differentiable function $V^*(x)$ such that the Hamilton–Jacobi–Bellman equation holds:

$$\mathcal{H}(V^*) = 0, \tag{102.}$$

where

$$\mathcal{H}(V) = \nabla V^{\mathrm{T}}(x)f(x) + q(x) - \frac{1}{4}\nabla V^{\mathrm{T}}(x)g(x)R^{-1}(x)g^{\mathrm{T}}(x)\nabla V(x).$$

Such a V^* can be easily shown to be a Lyapunov function such that

$$V^*(x_0) = \min_u J(x_0, u) = J(x_0, u^*), \quad \forall x_0 \in \mathbb{R}^n, \tag{103.}$$

corresponds to the optimal value function and also yields the optimal control input

$$u^*(x) = -\frac{1}{2}R^{-1}(x)g^{\mathrm{T}}(x)\nabla V^*(x). \tag{104.}$$

Finding a V^* that solves Equation 102 is too difficult. In addition, it is easy to see that it requires knowledge of f and g. Policy iteration is often used to find V^*, where the control input u is iterated, and with each new input u, the corresponding Lyapunov function is computed. The following procedure is often utilized:

1. For $i = 1, 2, \ldots$, solve for the cost function $V_i(x) \in \mathcal{C}^1$, with $V_i(0) = 0$, from the partial differential equation

$$\mathcal{L}(V_i(x), u_i(x)) = 0. \tag{105.}$$

   It can be seen that solving for Equation 105 requires the immediate cost $c(x, u_i)$, which may be available through an oracle when the plant model is not known.
2. Update the control policy using u_i and the value function estimate V_i as

$$u_{i+1}(x) = -\frac{1}{2}R^{-1}(x)g^{\mathrm{T}}(x)\nabla V_i(x). \tag{106.}$$

That is, instead of solving Equation 102, the approach seeks to find a sequence of V_i that satisfies Equation 105. Step 1 is referred to as policy evaluation, and step 2 is referred to as policy improvement. Then convergence of the stabilizing policy u^1 to an optimal policy u^* can be achieved, stated in the following theorem.

Theorem 2. Suppose Assumptions 5 and 6 hold, and the solution $V_i(x) \in \mathcal{C}^1$ satisfying Equation 105 exists, for $i = 1, 2, \ldots$. Let $V_i(x)$ and $u_{i+1}(x)$ be the functions generated from Equations 105 and 106. Then the following properties hold, $\forall i = 0, 1, \ldots$:

1. $V^*(x) \leq V_{i+1}(x) \leq V_i(x), \forall x \in \mathbb{R}^n$.
2. u_{i+1} is globally stabilizing.
3. Suppose there exist $V^o \in \mathcal{C}^1$ and u^o such that $\forall x_0 \in \mathbb{R}^n$, we have $\lim_{i\to\infty} V_i(x_0) = V^o(x_0)$ and $\lim_{i\to\infty} u_i(x_0) = u^o(x_0)$. Then, $V^* = V^o$ and $u^* = u^o$.

The problem that still remains is the following. The solution of Equation 105 that determines a global solution $V_i(x)$ for all x and policies $u_i(x)$, especially when the precise knowledge of f or g is not available, is still nontrivial. Also, as mentioned earlier, any approximation-based approaches, such as RL, pose problems of stability and robustness. As stated succinctly by Jiang & Jiang (89, p. 2919), "although approximation methods can give acceptable results on some compact set in the state space, they cannot be used to achieve global stabilization." Any efforts to reduce the approximation error, including neural networks and sum of squares, carry with them a large computational burden in addition to issues of robustness. By contrast, stability and robustness properties are clearly delineated with the use of AC. However, these guarantees are predicated on specific structures of dynamic systems such as Equation 88 or Equations 46 and 47.

6. COMPARISONS, COMBINATIONS, AND CONCLUSIONS

We begin with comparisons of the AC and RL approaches using Examples 1 and 2. The first point to note from Example 1 is that in AC, both the direct and indirect approaches explicitly used the structure of the system model. The parameters A and B of the linear model are tied intimately to the knowledge of the system model and its order, input, and dimensions. The indirect approach relied further on persistent excitation and therefore on learning the parameters accurately; the direct approach, by contrast, did not require persistent excitation, allowed imperfect learning, and still ensured stable control. Optimality followed learning in the indirect approach and followed stability in the direct approach. Unlike AC, the RL approach is agnostic to the model structure. It is assumed that an oracle or a simulation engine is available that allows offline experimentation,

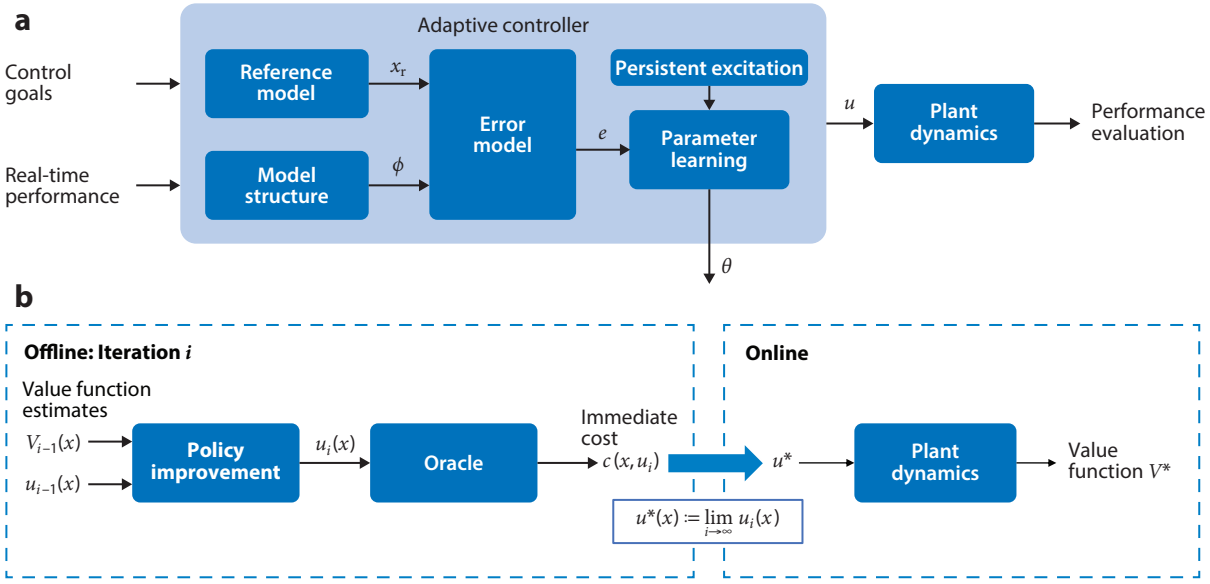

Figure 3

Schematics of (*a*) an adaptive controller and (*b*) reinforcement learning. The adaptive controller is an online solution that monitors the performance in real time (such as the loss function L_1 in Equation 11 or L_2 in Equation 23) and suitably designs the control input u. The adaptive controller uses the plant model structure to identify an underlying regressor ϕ and a reference model to determine both the input and a parameter estimate θ. The regressor and the reference model lead to an error model that relates the parameter error to the real-time performance. This error model is utilized to determine the rule by which the parameter estimate θ is adjusted. Such an adaptive system is always guaranteed to achieve the desired real-time performance, even with imperfect learning. In addition, if the regressor ϕ satisfies persistent excitation properties, then parameter learning takes place as well. Reinforcement learning is an offline approach where the oracle is the entity used to generate the response to a policy choice and can be viewed as the plant model. An immediate cost $c(x, u_i)$ is computed based on the system response generated by the oracle and is used to update the estimated value function $V_{i-1}(x)$. An iterative procedure is used to update both the policy as u_i and estimates of value function, using which an optimal policy $u^*(x)$ is obtained after $u_i(x)$ converges. If the oracle is identical to the plant dynamics, then applying $u^*(x)$ to the plant online achieves the optimal value function $V^*(x)$.

and that one can collect a large amount of data pertaining to (x, u) pairs, often enough to permit the learning of the optimal policy. When the underlying Bellman equation becomes computationally infeasible to solve, approximations are deployed. The optimality of the resulting policy improves as the approximation error becomes small.

All of the above statements apply to Example 2 as well. The AC approach relied completely on a model structure, including the order as well as the nature of the nonlinearities present. The specific result outlined required the system to be feedback linearizable. Here, too, imperfect learning was possible by leveraging the model structure, including that of the nonlinearities. The focus once again was on stability, and as before, optimality follows once learning takes place. The specific RL approach that was discussed proposed an optimal solution for a class of nonlinear systems under some assumptions.

Schematics of both the AC and RL approaches are shown in **Figure 3**. The RL approach indicated in the figure illustrates popular methods, including the *Q*-learning approach described in Section 4 and methods based on temporal-difference learning (90). It should be noted that there are online methods based on RL applied to control of nonlinear systems control (89, 91, 92) that provide analytical guarantees based on assumptions such as knowledge of an initial stable control law u_0 and a sufficiently accurate oracle to reduce the sim-to-real gap.

Both methods require assumptions and restrictions. The AC approach is predicated on a model structure, with the rationale that the underlying problem is grounded in physical mechanisms and is therefore amenable to a model structure that could be determined from an understanding of physical laws, conservation equations, and constitutive relations. Assumptions that equilibrium points, order, model structure, and feedback linearizability are all known are not always valid. These assumptions are stress tested as the scope of the systems being addressed increases in complexity, size, and scale. Not all systems can be modeled as in the above examples. Complex systems and stringent performance specifications of efficiency, reliability, and resilience pose tremendous challenges, as unmodeled components introduce uncertainties of various kinds, and in real time. The RL approach outlined here is model agnostic and is therefore applicable to a wider class of systems. However, restrictions arise because it is a data-intensive approach—often requiring training on an offline dataset or through the use of an environment simulator. This implies in turn that there is sufficient information available about the true system to replicate in the form of a simulation engine or collected dataset. That is, the sim-to-real gap is a huge challenge that must be addressed to render RL applicable in real time for safety-critical systems. As there are always unknown uncertainties that can occur and cannot be anticipated or incorporated in the simulation engine, the desirable properties of robustness, stability, reliability, and resilience all need to be addressed. In all cases, these decisions must be synthesized with inaccurate, partial, and limited information in real time, which in turn imposes challenges for both approaches, albeit different ones.

While the challenges seem formidable and pose roadblocks for both approaches, they also present tremendous opportunities. The fact that the two approaches are different suggests that there are ways in which they could be integrated so as to realize their combined advantages. The focus of AC on stability and RL on optimality suggests that one such candidate is a multiloop approach, with an inner loop focused on AC methods that are capable of delivering real-time performance with stability guarantees and an outer loop focused on RL methods that can anticipate an optimal policy when the sim-to-real gap is small (93–97).

The problem of controlling a large-scale dynamic system in real time is exceedingly complex. The two solutions that have been delineated in this article, AC and RL, are huge subfields of control that have been researched over the past several years. AC has been synthesized through the lens of stability, parameter learning, online control, and continuous-time systems, and RL has been synthesized through an optimality-based and data-driven viewpoint. It is clear that the concept of learning is common to both. Stability is followed by learning and optimality in AC, while RL attempts to achieve optimality through learning and simulation. While analytical rigor and provable correctness in real time are hallmarks of AC, they are also plagued with several restrictions and difficulty in extending the approach to complex dynamic systems. Comparatively, RL has achieved enormous success in difficult problems related to games and pattern recognition, although the lack of guarantees of stability and robustness is a deficiency that remains to be addressed. Both approaches have learning as a fundamental tenet and employ an iterative procedure that consists of an action–response–correction sequence. Despite these rich intersections and commonalities, little effort has been expended in comparing the two approaches or in combining their philosophies and methodologies. This article takes a first step in this direction.

There are several directions that were not explored here owing to space limitations. Each field is vast, with several subtopics that have deep and varying insights and rich results. The intent here is not to provide a comprehensive exposition of these topics but rather to expose the reader to these distinct inquiries into an extremely challenging problem. Several societal-scale challenges, including sustainability, quality of life, and resilient infrastructure, have in their core the need to analyze and synthesize complex systems. AC and RL are fundamental building blocks that need to be refined to meet this need.

DISCLOSURE STATEMENT

The author is not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

ACKNOWLEDGMENTS

I gratefully acknowledge support from the Boeing Strategic University Initiative. I would also like to acknowledge several useful discussions with Peter Fisher, Anubhav Guha, and Sunbochen Tang as well as the comments of an anonymous reviewer, which led to a significant improvement in the readability of the article.

LITERATURE CITED

1. Qu Z, Thomsen B, Annaswamy AM. 2020. Adaptive control for a class of multi-input multi-output plants with arbitrary relative degree. *IEEE Trans. Autom. Control* 65:3023–38
2. Annaswamy AM, Fradkov AL. 2021. A historical perspective of adaptive control and learning. *Annu. Rev. Control* 52:18–41
3. Krstić M. 2021. *Control has met learning: aspirational lessons from adaptive control theory*. Control Meets Learning Seminar, virtual, June 9
4. Narendra KS, Annaswamy AM. 2005. *Stable Adaptive Systems*. Mineola, NY: Dover (original publication by Prentice Hall, 1989)
5. Åström KJ, Wittenmark B. 1995. *Adaptive Control*. Reading, MA: Addison-Wesley. 2nd ed.
6. Ioannou PA, Sun J. 1996. *Robust Adaptive Control*. Upper Saddle River, NJ: Prentice Hall
7. Sastry S, Bodson M. 1989. *Adaptive Control: Stability, Convergence and Robustness*. Upper Saddle River, NJ: Prentice Hall
8. Krstić M, Kanellakopoulos I, Kokotović P. 1995. *Nonlinear and Adaptive Control Design*. New York: Wiley
9. Tao G. 2003. *Adaptive Control Design and Analysis*. New York: Wiley
10. Narendra KS, Annaswamy AM. 1987. Persistent excitation in adaptive systems. *Int. J. Control* 45:127–60
11. Boyd S, Sastry S. 1983. On parameter convergence in adaptive control. *Syst. Control Lett.* 3:311–19
12. Morgan AP, Narendra KS. 1977. On the uniform asymptotic stability of certain linear nonautonomous differential equations. *SIAM J. Control Optim.* 15:5–24
13. Anderson BDO, Johnson CR Jr. 1982. Exponential convergence of adaptive identification and control algorithms. *Automatica* 18:1–13
14. Jenkins B, Krupadanam A, Annaswamy AM. 2019. Fast adaptive observers for battery management systems. *IEEE Trans. Control Syst. Technol.* 28:776–89
15. Gaudio JE, Annaswamy AM, Bolender MA, Lavretsky E, Gibson TE. 2021. A class of high order tuners for adaptive systems. *IEEE Control Syst. Lett.* 5:391–96
16. Lavretsky E, Wise KA. 2013. *Robust and Adaptive Control with Aerospace Applications*. London: Springer
17. Luders G, Narendra KS. 1974. Stable adaptive schemes for state estimation and identification of linear systems. *IEEE Trans. Autom. Control* 19:841–47
18. Lion PM. 1967. Rapid identification of linear and nonlinear systems. *AIAA J.* 5:1835–42
19. Kreisselmeier G. 1977. Adaptive observers with exponential rate of convergence. *IEEE Trans. Autom. Control* 22:2–8
20. Aranovskiy S, Belov A, Ortega R, Barabanov N, Bobtsov A. 2019. Parameter identification of linear time-invariant systems using dynamic regressor extension and mixing. *Int. J. Adapt. Control Signal Process.* 33:1016–30
21. Ortega R, Aranovskiy S, Pyrkin A, Astolfi A, Bobtsov A. 2020. New results on parameter estimation via dynamic regressor extension and mixing: continuous and discrete-time cases. *IEEE Trans. Autom. Control* 66:2265–72
22. Gaudio JE, Annaswamy AM, Lavretsky E, Bolender MA. 2020. Fast parameter convergence in adaptive flight control. In *AIAA Scitech 2020 Forum*, pap. 2020-0594. Reston, VA: Am. Inst. Aeronaut. Astronaut.
23. Kailath T. 1980. *Linear Systems*. Englewood Cliffs, NJ: Prentice Hall

24. Chen CT. 1984. *Linear System Theory and Design*. New York: Holt, Rinehart & Winston
25. Feldbaum A. 1960. Dual control theory. I. *Avtom. Telemekhanika* 21:1240–49
26. Yakubovich VA. 1962. The solution of certain matrix inequalities in automatic control theory. *Dokl. Akad. Nauk* 143:1304–7
27. Kalman RE. 1963. Lyapunov functions for the problem of Lur'e in automatic control. *PNAS* 49:201–5
28. Meyer K. 1965. On the existence of Lyapunov function for the problem of Lur'e. *J. Soc. Ind. Appl. Math. A* 3:373–83
29. Lefschetz S. 1965. *Stability of Nonlinear Control Systems*. New York: Academic
30. Narendra KS, Taylor JH. 1973. *Frequency Domain Criteria for Absolute Stability*. New York: Academic
31. Fradkov A. 1974. Synthesis of adaptive system of stabilization for linear dynamic plants. *Autom. Remote Control* 35:1960–66
32. Anderson BDO, Bitmead RR, Johnson CR Jr., Kokotović PV, Kosut RL, et al. 1986. *Stability of Adaptive Systems: Passivity and Averaging Analysis*. Cambridge, MA: MIT Press
33. Evesque S, Annaswamy AM, Niculescu S, Dowling AP. 2003. Adaptive control of a class of time-delay systems. *J. Dyn. Syst. Meas. Control* 125:186–93
34. Anderson BDO. 1985. Adaptive systems, lack of persistency of excitation and bursting phenomena. *Automatica* 21:247–58
35. Morris A, Fenton T, Nazer Y. 1977. Application of self-tuning regulators to the control of chemical processes. *IFAC Proc. Vol.* 10(16):447–55
36. Fortescue T, Kershenbaum LS, Ydstie BE. 1981. Implementation of self-tuning regulators with variable forgetting factors. *Automatica* 17:831–35
37. Narendra KS, Annaswamy AM. 1987. A new adaptive law for robust adaptation without persistent excitation. *IEEE Trans. Autom. Control* 32:134–45
38. Narendra KS, Annaswamy AM. 1986. Robust adaptive control in the presence of bounded disturbances. *IEEE Trans. Autom. Control* 31:306–15
39. Jenkins BM, Annaswamy AM, Lavretsky E, Gibson TE. 2018. Convergence properties of adaptive systems and the definition of exponential stability. *SIAM J. Control Optim.* 56:2463–84
40. Kumar PR. 1983. Optimal adaptive control of linear-quadratic-Gaussian systems. *SIAM J. Control Optim.* 21:163–78
41. Desoer C, Liu R, Auth L. 1965. Linearity versus nonlinearity and asymptotic stability in the large. *IEEE Trans. Circuit Theory* 12:117–18
42. Goodwin GC, Ramadge PJ, Caines PE. 1981. Discrete time stochastic adaptive control. *SIAM J. Control Optim.* 19:829–53
43. Goodwin GC, Ramadge PJ, Caines PE. 1980. Discrete-time multivariable adaptive control. *IEEE Trans. Autom. Control* 25:449–56
44. Becker A, Kumar PR, Wei CZ. 1985. Adaptive control with the stochastic approximation algorithm: geometry and convergence. *IEEE Trans. Autom. Control* 30:330–38
45. Stevens BL, Lewis FL. 2003. *Aircraft Control and Simulation*. Hoboken, NJ: Wiley. 2nd ed.
46. Rohrs C, Valavani L, Athans M, Stein G. 1985. Robustness of continuous-time adaptive control algorithms in the presence of unmodeled dynamics. *IEEE Trans. Autom. Control* 30:881–89
47. Marino R, Tomei P. 1993. Global adaptive output-feedback control of nonlinear systems. II. Nonlinear parameterization. *IEEE Trans. Autom. Control* 38:33–48
48. Seto D, Annaswamy AM, Baillieul J. 1994. Adaptive control of nonlinear systems with a triangular structure. *IEEE Trans. Autom. Control* 39:1411–28
49. Wen C, Hill DJ. 1990. Adaptive linear control of nonlinear systems. *IEEE Trans. Autom. Control* 35:1253–57
50. Gusev S. 1988. Linear stabilization of nonlinear systems program motion. *Syst. Control Lett.* 11:409–12
51. Marino R. 1985. High-gain feedback in non-linear control systems. *Int. J. Control* 42:1369–85
52. Haddad WM, Chellaboina V, Hayakawa T. 2001. Robust adaptive control for nonlinear uncertain systems. In *Proceedings of the 40th IEEE Conference on Decision and Control*, Vol. 2, pp. 1615–20. Piscataway, NJ: IEEE
53. Fradkov A, Lipkovich M. 2015. Adaptive absolute stability. *IFAC-PapersOnLine* 48(11):258–63

54. Astolfi A, Karagiannis D, Ortega R. 2007. *Nonlinear and Adaptive Control with Applications*. London: Springer
55. Fomin V, Fradkov AL, Yakubovich V. 1981. *Adaptive Control of Dynamical Systems*. Moscow: Nauka
56. Seron MM, Hill DJ, Fradkov AL. 1995. Nonlinear adaptive control of feedback passive systems. *Automatica* 31:1053–60
57. Andrievsky B, Selivanov A. 2020. Historical overview of the passification method and its applications to nonlinear and adaptive control problems. In *2020 European Control Conference*, pp. 791–94. Piscataway, NJ: IEEE
58. Astolfi A, Ortega R. 2003. Immersion and invariance: a new tool for stabilization and adaptive control of nonlinear systems. *IEEE Trans. Autom. Control* 48:590–606
59. Narendra KS, Parthasarathy K. 1990. Identification and control of dynamical systems using neural networks. *IEEE Trans. Neural Netw.* 1:4–27
60. Sanner RM, Slotine JJE. 1992. Gaussian networks for direct adaptive control. *IEEE Trans. Neural Netw.* 3:837–63
61. Polycarpou MM. 1996. Stable adaptive neural control scheme for nonlinear systems. *IEEE Trans. Autom. Control* 41:447–51
62. Lewis FL, Yesildirek A, Liu K. 1996. Multilayer neural-net robot controller with guaranteed tracking performance. *IEEE Trans. Neural Netw.* 7:388–99
63. Chang YC, Roohi N, Gao S. 2019. Neural Lyapunov control. In *Advances in Neural Information Processing Systems 32*, ed. H Wallach, H Larochelle, A Beygelzimer, F d'Alché-Buc, E Fox, R Garnett, pp. 3245–54. Red Hook, NY: Curran
64. Yu SH, Annaswamy AM. 1998. Stable neural controllers for nonlinear dynamic systems. *Automatica* 34:641–50
65. Bertsekas DP. 2015. Value and policy iterations in optimal control and adaptive dynamic programming. *IEEE Trans. Neural Netw. Learn. Syst.* 28:500–9
66. Bertsekas DP. 2017. *Dynamic Programming and Optimal Control*, Vol. 1. Belmont, MA: Athena Sci.
67. Watkins CJ, Dayan P. 1992. *Q*-learning. *Mach. Learn.* 8:279–92
68. Recht B. 2019. A tour of reinforcement learning: the view from continuous control. *Annu. Rev. Control Robot. Auton. Syst.* 2:253–79
69. Yu SH, Annaswamy AM. 1996. Neural control for nonlinear dynamic systems. In *Advances in Neural Information Processing Systems 8*, ed. D Touretzky, MC Mozer, ME Hasselmo, pp. 1010–16, Cambridge, MA: MIT Press
70. Lagoudakis MG, Parr R. 2003. Least-squares policy iteration. *J. Mach. Learn. Res.* 4:1107–49
71. Bradtke SJ, Barto AG. 1996. Linear least-squares algorithms for temporal difference learning. *Mach. Learn.* 22:33–57
72. Narendra KS, Parthasarathy K. 1991. Gradient methods for the optimization of dynamical systems containing neural networks. *IEEE Trans. Neural Netw.* 2:252–62
73. Werbos PJ. 1990. Backpropagation through time: what it does and how to do it. *Proc. IEEE* 78:1550–60
74. Finn C, Abbeel P, Levine S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, ed. Doina Precup, YW Teh, pp. 1126–35. Proc. Mach. Learn. Res. 70. N.p.: PMLR
75. Schulman J, Levine S, Abbeel P, Jordan M, Moritz P. 2015. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, ed. F Bach, D Blei, pp. 1889–97. Proc. Mach. Learn. Res. 37. N.p.: PMLR
76. Silver D, Lever G, Heess N, Degris T, Wierstra D, Riedmiller M. 2014. Deterministic policy gradient algorithms. In *Proceedings of the 31st International Conference on Machine Learning*, ed. EP Xing, T Jebara, pp. 387–395. Proc. Mach. Learn. Res. 32. N.p.: PMLR
77. Zhao W, Queralta JP, Westerlund T. 2020. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE Symposium Series on Computational Intelligence*, pp. 737–44. Piscataway, NJ: IEEE
78. Chebotar Y, Handa A, Makoviychuk V, Macklin M, Issac J, et al. 2019. Closing the sim-to-real loop: adapting simulation randomization with real world experience. In *2019 International Conference on Robotics and Automation*, pp. 8973–79. Piscataway, NJ: IEEE

79. Campi MC, Kumar PR. 1998. Adaptive linear quadratic Gaussian control: the cost-biased approach revisited. *SIAM J. Control Optim.* 36:1890–907
80. Borkar V, Varaiya P. 1979. Adaptive control of Markov chains, I: finite parameter set. *IEEE Trans. Autom. Control* 24:953–57
81. Campi MC, Kumar PR. 1996. Optimal adaptive control of an LQG system. In *Proceedings of 35th IEEE Conference on Decision and Control*, Vol. 1, pp. 349–53. Piscataway, NJ: IEEE
82. Guo L, Chen HF. 1991. The Åström–Wittenmark self-tuning regulator revisited and ELS-based adaptive trackers. *IEEE Trans. Autom. Control* 36:802–12
83. Guo L. 1995. Convergence and logarithm laws of self-tuning regulators. *Automatica* 31:435–50
84. Duncan T, Guo L, Pasik-Duncan B. 1999. Adaptive continuous-time linear quadratic Gaussian control. *IEEE Trans. Autom. Control* 44:1653–62
85. Dean S, Tu S, Matni N, Recht B. 2018. Safely learning to control the constrained linear quadratic regulator. arXiv:1809.10121 [math.OC]
86. Abbasi-Yadkori Y, Szepesvári C. 2011. Regret bounds for the adaptive control of linear quadratic systems. In *Proceedings of the 24th Annual Conference on Learning Theory*, ed. SM Kakade, U Von Luxburg, pp. 1–26. Proc. Mach. Learn. Res. 19. N.p.: PMLR
87. Lin YH, Narendra K. 1980. A new error model for adaptive systems. *IEEE Trans. Autom. Control* 25:585–87
88. Lewis FL, Vrabie D. 2009. Reinforcement learning and adaptive dynamic programming for feedback control. *IEEE Circuits Syst. Mag.* 9(3):32–50
89. Jiang Y, Jiang ZP. 2015. Global adaptive dynamic programming for continuous-time nonlinear systems. *IEEE Trans. Autom. Control* 60:2917–29
90. Tsitsiklis J, Van Roy B. 1996. Analysis of temporal-diffference learning with function approximation. In *Advances in Neural Information Processing Systems 9*, ed. MC Mozer, M Jordan, T Petsche, pp. 1075–81. Cambridge, MA: MIT Press
91. Vrabie D, Pastravanu O, Abu-Khalaf M, Lewis FL. 2009. Adaptive optimal control for continuous-time linear systems based on policy iteration. *Automatica* 45:477–84
92. Berkenkamp F, Turchetta M, Schoellig A, Krause A. 2017. Safe model-based reinforcement learning with stability guarantees. In *Advances in Neural Information Processing Systems 30*, ed. I Guyon, U Von Luxburg, S Bengio, H Wallach, R Fergus, et al., pp. 908–18. Red Hook, NY: Curran
93. Annaswamy AM, Guha A, Cui Y, Tang S, Gaudio JE. 2022. Integration of adaptive control and reinforcement learning approaches for real-time control and learning. arXiv:2105.06577 [cs.LG]
94. Matni N, Proutiere A, Rantzer A, Tu S. 2019. From self-tuning regulators to reinforcement learning and back again. In *2019 IEEE 58th Conference on Decision and Control*, pp. 3724–40. Piscataway, NJ: IEEE
95. Westenbroek T, Mazumdar E, Fridovich-Keil D, Prabhu V, Tomlin CJ, Sastry SS. 2020. Adaptive control for linearizable systems using on-policy reinforcement learning. In *2020 59th IEEE Conference on Decision and Control*, pp. 118–25. Piscataway, NJ: IEEE
96. Sun R, Greene ML, Le DM, Bell ZI, Chowdhary G, Dixon WE. 2021. Lyapunov-based real-time and iterative adjustment of deep neural networks. *IEEE Control Syst. Lett.* 6:193–98
97. Richards SM, Azizan N, Slotine JJ, Pavone M. 2021. Adaptive-control-oriented meta-learning for nonlinear systems. In *Robotics: Science and Systems XVII*, ed. D Shell, M Toussaint, MA Hsieh, pap. 56. N.p.: Robot. Sci. Syst. Found.