Adaptive Control for Linearizable Systems Using On-Policy Reinforcement Learning

Tyler Westenbroek, Eric Mazumdar, David Fridovich-Keil, Valmik Prabhu,
Claire J. Tomlin and S. Shankar Sastry

***Abstract*— This paper proposes a framework for adaptively learning a feedback linearization-based tracking controller for an unknown system using discrete-time model-free policy-gradient parameter update rules. The primary advantage of the scheme over standard model-reference adaptive control techniques is that it does not require the learned inverse model to be invertible at all instances of time. This enables the use of general function approximators to approximate the linearizing controller for the system without having to worry about singularities. However, the discrete-time and stochastic nature of these algorithms precludes the direct application of standard machinery from the adaptive control literature to provide deterministic stability proofs for the system. Nevertheless, we leverage these techniques alongside tools from the stochastic approximation literature to demonstrate that with high probability the tracking and parameter errors concentrate near zero when a certain persistence of excitation condition is satisfied. A simulated example of a double pendulum demonstrates the utility of the proposed theory.** [1]

I. Introduction

Many real-world control systems display nonlinear behaviors which are difficult to model, necessitating the use of control architectures which can adapt to the unknown dynamics online while maintaining certificates of stability. There are many successful model-based strategies for adaptively constructing controllers for uncertain systems [1–3], but these methods often require the presence of a simple, reasonably accurate parametric model of the system dynamics. Recently, however, there has been a resurgence of interest in the use of model-free reinforcement learning techniques to construct feedback controllers without the need for a reliable dynamics model [4–6]. As these methods begin to be deployed in real world settings, a new theory is needed to understand the behavior of these algorithms as they are integrated into safety-critical control loops.

However, the majority of the theory for adaptive control is stated in continuous-time [2], while reinforcement learning algorithms are typically implemented and studied in discrete-time settings [7, 8]. There have been several attempts to define and study policy-gradient algorithms in continuous-time [9, 10], yet many real-world systems have actuators which can only be updated at a fixed maximum sampling frequency. Thus, we find it more natural and practically applicable to unify these methods in the sampled-data setting.

Specifically, this paper addresses the model mismatch issue by combining continuous-time adaptive control techniques with discrete-time model-free reinforcement learning algorithms to learn a feedback linearization-based tracking controller for an unknown system, online. Unfortunately, it is well-known that sampling can destroy the affine relationship between system inputs and outputs which is usually assumed and then exploited in the stability proofs from the adaptive control literature [11]. To overcome this challenge, we first ignore the effects of sampling and design an idealized continuous-time behavior for the system's tracking and parameter error dynamics which employs a least-squares gradient following update rule. In the sampled-data setting, we then use an Euler approximation of the continuous-time reward signal and implement a policy-gradient parameter update rule to produce a noisy approximation to the ideal continuous-time behavior. Our framework is closely related to that of [12]; however, in this paper we address the problem of online adaptation of the learned parameters whereas [12] considers a fully offline setting.

Beyond naturally bridging continuous-time and sampled-data settings, the primary advantage of our approach is that it does not suffer from the "loss of controllability" phenomena which is a core challenge in the model-reference adaptive control literature [1, 13]. This issue arises when the parameterized estimate for the system's decoupling matrix becomes singular, in which case either the learned linearizing control law or associated parameter update scheme may break down. To circumvent this issue, projection-based parameter update rules are used to keep the parameters in a region in which the estimate for the decoupling matrix is known to be invertible. In practice, the construction of these regions requires that a simple parameterization of the system's nonlinearities is available [14]. In contrast, the model-free approach we introduce does not suffer from singularities and can naturally incorporate 'universal' function approximators such as radial bases functions or bases of polynomials.

However, due to the non-deterministic nature of our sampled-data control law and parameter update scheme, the deterministic guarantees usually found in the adaptive control literature do not apply here. Indeed, policy-gradient parameter updates are known to suffer from high variances [15]. Nevertheless, we demonstrate that when a standard persistence of excitation condition is satisfied the tracking and parameter errors of the system concentrate around the

The authors are with the department of Electrical Engineering and Computers Sciences and the University of California, Berkeley.

[1] This version of this preprint corrects a typo in the statement of Theorem 1 in Section III-C which appeared in a previous version. In particular, the right hand side of (51) was originally $C_2 M \sqrt{\frac{\Delta t \ln\left(\frac{\lambda}{2}\right)}{\zeta \sigma^2}}$ but has now been corrected to $C_2 M \sqrt{\frac{\Delta t \ln\left(\frac{2}{\lambda}\right)}{\zeta \sigma^2}}$.

origin with high probability even when the most basic policy-gradient update rule is used. Our analysis technique is derived from the adaptive control literature and the theory of stochastic approximations [8, 16]. After developing the basic theory we discuss how common heuristics from the reinforcement learning literature can be used to reduce the variance of the policy gradient updates. Due to space constraints, we outline the analysis techniques we employ but leave a number of proofs to a technical report [17]. Finally, a simulation of a double pendulum demonstrates the utility of the approach.

A. Related Work

A number of approaches have been proposed to avoid the "loss of controllability" problem discussed above. One approach is to perturb the estimated linearizing control law to avoid singularities [13, 18, 19]. However, this method never learns the exact linearizing controller during operation and hence sacrifices some tracking performance. Other approaches avoid the need to invert the input-output dynamics by driving the system states to a sliding surface [3]. Unfortunately, these methods require high-gain feedback which may lead to undesirable effects such as actuator saturation. Several model-free approaches similar to the one we consider here have been proposed in the literature [20, 21], but these focus on actor-critic methods and, to the best of our knowledge, do not provide any proofs of convergence. Recently, non-parametric function approximators have been been used to learn a linearizing controller [22, 23], but these methods still require structural assumptions to avoid singularities.

While our parameter-update scheme is most closely related to the policy gradient literature, e.g., [7], we believe that recent work in meta-learning [24, 25] is also similar to our own work, at least in spirit. Meta-learning aims to learn priors on the solution to a given machine learning problem, and thereby speed up online fine tuning when presented with a slightly different instance of the problem [26]. Meta-learning is used in practice to apply reinforcement learning algorithms in hardware settings [27, 28].

B. Preliminaries

Next, we fix mathematical notation and review some definitions used extensively in the paper. Given a random variable X, if they exist the expectation of X is denoted $\mathbb{E}[X]$ and its variances is denoted by $Var(X)$. Our analysis heavily relies on the notion of a *sub-Gaussian* distribution. We say that a random variable $X \in \mathbb{R}^n$ is sub-Gaussian if there exists a constant $C > 0$ such that for each $t \geq 0$ we have $\mathcal{P}\{|x|_2 \geq t\} \leq 2\exp(-\frac{t^2}{C^2})$. Informally, a distribution is sub-Gaussian if it's tail is dominated by the tail of some Gaussian distribution. We endow the space of sub-Gaussian distributions with the norm $\|\cdot\|_{\psi_2}$ defined by $\|X\|_{\psi_2} = \inf\Big\{t > 0 \colon \mathbb{E}[\exp(\frac{\|X\|_2^2}{t^2})] \leq 2\Big\}$. As an example, if $X = \mathcal{N}(0, \sigma^2 I)$ is a zero-mean Gaussian distribution with variance $\sigma^2 I$ (with I the n-dimensional identity) then $\|X\|_{\psi_2}$ is sub-Gaussian with norm $\|X\|_{\psi_2} \leq C\sigma$, where the constant $C > 0$ does not depend on σ^2.

II. Feedback Linearization

Throughout the paper we will focus on constructing output tracking controllers for systems of the form

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^q$ is the input and $y \in \mathbb{R}^q$ is the output. The mappings $f \colon \mathbb{R}^n \to \mathbb{R}^n$, $g \colon \mathbb{R}^n \to \mathbb{R}^{n \times q}$ and $h \colon \mathbb{R}^n \to \mathbb{R}^q$ are each assumed to be smooth, and we assume without loss of generality that the origin is an equilibrium point of the undriven system, i.e., $f(x) = 0$. Throughout the paper, we will also assume that state x and the output y can both be measured.

A. Single-input single-output systems

We begin by introducing feedback linearization for single-input, single-output (SISO) systems (i.e., $q = 1$). We begin by examining the first time derivative of the output:

$$\dot{y} = L_f h(x) + L_g h(x) \tag{2}$$

Here the terms $L_f h(x) = \frac{d}{dx}h(x) \cdot f(x)$ and $L_g h(x) = \frac{d}{dx}h(x) \cdot g(x)$ are known as *Lie derivatives* [2]. In the case that $L_g h(x) \neq 0$ for each $x \in \mathbb{R}^n$, we can apply

$$u(x, v) = \frac{1}{L_g h(x)}(-L_f h(x) + v) \ , \tag{3}$$

which exactly 'cancels out' the nonlinearities of the system and enforces the linear relationship $\dot{y} = v$ with v some arbitrary, auxiliary input. However if the input does not affect the first time derivative of the output—that is, if $L_g h \equiv 0$—then the control law (3) will be undefined. In general, we can differentiate y multiple times, until the input shows up in one of the higher derivatives of the output. Assuming that the input does not appear the first $\gamma - 1$ times we differentiate the output, the γ-th time derivative of y will be of the form

$$y^{(\gamma)} = L_f^\gamma h(x) + L_g L_f^{\gamma-1} h(x) u \tag{4}$$

Here, $L_f^\gamma h(x)$ and $L_g L_f^{\gamma-1} h(x)$ are higher order Lie derivatives, and we direct the reader to [2, Chapter 9] for further details. If $L_g L_f^{\gamma-1} h(x) \neq 0$ for each $x \in \mathbb{R}^n$ then setting

$$u(x, v) = \frac{1}{L_g L_f^{\gamma-1} h(x)}\big(-L_f^\gamma h(x) + v\big) \tag{5}$$

enforces the trivial linear relationship $y^\gamma = v$. We refer to γ as the *relative degree* of the nonlinear system, which is simply the order of its input-output relationship.

B. Multiple-input multiple-output systems

Next, we consider square multiple-input, multiple-output (MIMO) systems where $q > 1$. As in the SISO case, we differentiate each of the output channels until at least one input appears. Let γ_j be the number of times we need to differentiate y_j (the j-th entry of y) for at least one input to appear. Combining the resulting expressions for each of the outputs yields an input-output relationship of the form

$$y^{(\gamma)} = b(x) + A(x)u \tag{6}$$

where we have adopted the shorthand $y^{(\gamma)} = [y_1^{(\gamma_1)}, \dots, y_q^{(\gamma_q)}]^T$. Here, the matrix $A(x) \in \mathbb{R}^{q\times q}$ is known as the *decoupling matrix* and the vector $b(x) \in \mathbb{R}^q$ is known as the *drift term*. If $A(x)$ is non-singular on for each $x \in \mathbb{R}^n$ then we observe that the control law

$$u(x, v) = A^{-1}(x)(-b(x) + v) \tag{7}$$

where $v \in \mathbb{R}^q$ yields the decoupled linear system

$$[y_1^{(\gamma_1)}, y_2^{(\gamma_2)}, \dots, y_q^{(\gamma_q)}]^T = [v_1, v_2, \dots, v_q]^T, \tag{8}$$

where v_k is the k-th entry of v and $y_j^{\gamma_j}$ is the γ_j-th time derivative of the j-th output. We refer to $\gamma = (\gamma_1, \gamma_2, \dots, \gamma_q)$ as the *vector relative degree* of the system, with $|\gamma| = \sum_i \gamma_i$ the total relative degree of all dimensions. The decoupled dynamics (8) can be compactly represented with the LTI system

$$\dot{\xi}_r = A\xi_r + Bv_r \tag{9}$$

which we will hereafter refer to as the *reference model*. Here, $A \in \mathbb{R}^{|\gamma|\times|\gamma|}$ and $B \in \mathbb{R}^{|\gamma|\times q}$ is constructed so that $B^TB = I_{q\times q}$, where $I_{q\times q}$ is the q-dimensional identity matrix. Note that (9) collects $\xi_r = (y_1, \dot{y}_1, \dots, \dots, y_1^{\gamma_1-1}, \dots, y_q, \dots, y_q^{\gamma_q-1})$. It can be shown [2, Chapter 9] that there exists a change of coordinates $x \to (\xi, \eta)$ such that in the new coordinates and after application of the linearizing control law the dynamics of the system are of the form

$$\begin{aligned} \dot{\xi} &= A\xi + Bv \\ \dot{\eta} &= q(\xi, \eta) + p(\xi, \eta)v. \end{aligned} \tag{10}$$

That is, the $\xi \in \mathbb{R}^{|\gamma|}$ coordinates represent the portion of the system that has been linearized while the $\eta \in \mathbb{R}^{n-|\gamma|}$ coordinates represent the remaining coordinates of the nonlinear system. The undriven dynamics

$$\dot{\eta} = q(\xi, \eta) \tag{11}$$

are referred to as the *zero* dynamics. Conditions which ensure that the η coordinates remain bounded during operation will be discussed below.

C. Inversion & exact tracking for min-phase MIMO systems

Let us assume that we are given a desired reference signal $y_d(\cdot) = \big(y_{1,d}(\cdot), \dots, y_{q,d}(\cdot)\big)$. Our goal is to construct a tracking controller for the nonlinear system using the linearizing controller (7), along with a linear controller designed for the reference model (9) which makes use of both feedback terms. We will assume that the first γ_j derivatives of $y_{j,d}(\cdot)$ are well defined, and assume that the signal $\big(y_{j,d}(\cdot), y_{j,d}^{(1)}(\cdot), \dots, y_{q,d}^{(\gamma_q)}(\cdot)\big)$ can be bounded uniformly.

For compactness of notation, we will collect

$$y_d^{(\gamma)}(\cdot) = \big(y_{1,d}^{(\gamma_1)}(\cdot), y_{2,d}^{(\gamma_2)}(\cdot), \dots, y_{q,d}^{(\gamma_q)}(\cdot)\big)$$

$$\xi_d(\cdot) = \big(y_{1,d}(\cdot), \dots, y_{1,d}^{(\gamma_1-1)}(\cdot), \dots, y_{q,d}(\cdot), \dots, y_{q,d}^{(\gamma_q-1)}(\cdot)\big).$$

Here, $\xi(\cdot)$ is used to capture the desired trajectory of the linear reference model, and $y_d^{(\gamma)}(\cdot)$ will be used in a feed-forward term in the tracking controller. To construct the feedback term, we define the error

$$e(\cdot) = \xi(\cdot) - \xi_d(\cdot) \tag{12}$$

where $\xi(\cdot)$ is the actual trajectory of the linearized coordinates as in (10). Altogether, the tracking controller for the system is then given by

$$u = A^{-1}(x)\big(-b(x) + y_d^{(\gamma)} + Ke\big) \tag{13}$$

where $K \in \mathbb{R}^{q\times|\gamma|}$ is a linear feedback matrix designed so that $(A + BK)$ us Hurwitz. Under the application of this control law the closed loop error dynamics become

$$\dot{e} = (A + BK)e \tag{14}$$

and it becomes apparent that $e \to 0$ exponentially quickly. However, while the tracking error decays exponentially, the η coordinates may be come unbounded during operation, in which case the linearizing control law will break down. One sufficient condition for η to remain bounded is for the zero dynamics to be globally exponentially stable and for $\xi_d(\cdot)$ and $y_d(\cdot)$ to remain bounded [1, Chapter 9]. When the zero dynamics satiecfy this condition we say nonlinear system is *exponentially minimum phase*.

III. Adaptive Control

From here on, we will aim to learn a feedback linearization-based tracking controller for the unknown plant

$$\begin{aligned} \dot{x}_p &= f_p(x_p) + g_p(x_p)u_p \\ y_p &= h_p(x_p) \end{aligned} \tag{15}$$

in an adaptive fashion. We assume that we have access to a an approximate dynamics model for the plant

$$\dot{x}_m = f_m(x_m) + g_m(x_m)u_m \tag{16}$$

$$y_m = h_m(x_m), \tag{17}$$

which incorporates any prior information available about the plant. It is assumed that the state (x_m and x_p) for both systems belongs to $\mathbb{R}^n$, that the inputs and outputs for both systems belong to $\mathbb{R}^q$, and that each of the mappings in (15) and (16) are smooth. We make the following assumption about the model and plant:

Assumption 1: The plant and model have the same well-defined relative degree $\gamma = (\gamma_1, \gamma_2, \dots, \gamma_q)$ on all of $\mathbb{R}^n$.

Assumption 2: The model and plant are both exponentially minimum phase.

With these assumptions in place, we know that there are globally-defined linearizing controllers for the plant and model, which respectively take the following form:

$$u_p(x,v) = \beta_p(x) + \alpha_p(x)v$$
$$u_m(x,v) = \beta_m(x) + \alpha_m(x)v$$

While u_m can be calculated using the model dynamics and the procedures outlined in the previous section, the terms comprising u_p are unknown to us. However, we do know that they may be expressed as

$$\beta_p(x) = \beta_m(x) + \Delta b(x)$$
$$\alpha_p(x) = \alpha_m(x) + \Delta\alpha(x)$$

where $\Delta\beta\colon \mathbb{R}^n \to \mathbb{R}^q$ and $\Delta\alpha\colon \mathbb{R}^n \to \mathbb{R}^{q\times q}$ are unknown but continuous functions. Thus we construct an estimate for u_p of the form

$$\hat{u}(\theta,x,v) = \big(\beta_m(x) + \beta_{\theta_1}\big) + \big(\alpha_m(x) + \alpha_{\theta_2}(x)\big)v$$

where $\beta_{\theta_1}\colon \mathbb{R}^n \to \mathbb{R}^q$ is a parameterized estimate for $\Delta\beta$, and $\alpha_{\theta_2}\colon \mathbb{R}^n \to \mathbb{R}^{q\times q}$ is a parameterized estimate for $\Delta\alpha$. The parameters $\theta_1 = (\theta_1^1, \theta_1^2, \ldots, \theta_1^{K_1}) \in \mathbb{R}^{K_1}$ and $\theta_2 = (\theta_2^1, \theta_2^2, \ldots, \theta_2^{K_2}) \in \mathbb{R}^{K_2}$ are to be learned during online operation of the plant. Our theoretical results will assume that the estimates are of the form

$$\beta_{\theta_1}(x) = \sum_{k=1}^{K_1} \theta_1^k \beta_k(x) \qquad \alpha_{\theta_2}(x) = \sum_{k=1}^{K_2} \theta_2^k \alpha_k(x) \tag{18}$$

where $\{\beta_k\}_{k=1}^{K_1}$ and $\{\alpha_k\}_{k=1}^{K_2}$ are linearly independent bases of functions, such as polynomials or radial basis functions.

A. Idealized continuous-time behavior

We now introduce a continuous-time update rule for the parameters of the learned linearizing controller which assumes that we know the functional form of the nonlinearities of the system. In Section III-B, we demonstrate how to approximate this ideal behavior in the sampled data setting using a policy gradient update rule which requires no information about the structure of the plant's nonlinearities.

We begin by assuming that there exists a set of "true" parameters $\theta^* = (\theta_1^*, \theta_2^*) \in \mathbb{R}^{K_1+K_2}$ for the plant so that for each $x \in \mathbb{R}^n$ and $v \in \mathbb{R}^q$ we have $\hat{u}(\theta^*, x, v) \equiv u_p(x,v)$. In this case, we can write our parameter estimation error as $\phi = (\theta_1 - \theta_1^*, \theta_2 - \theta_2^*)$ so that $\theta = \phi + \theta^*$.

With the gain matrix K constructed as in Section II-C, an estimate for the feedback linearization-based tracking controller is of the form

$$u = \hat{u}(\theta, x, y_d^\gamma + Ke). \tag{19}$$

When this control law is applied to the system the closed-loop error dynamics take the form

$$\dot{e} = (A + BK)e + BW(x, y_d^\gamma, e)\phi \tag{20}$$

where W is a complicated function of x, y_d^γ and e which contains terms involving $b_p(x), A_p(x), \beta_m(x), \alpha_m(x), \beta_p(x)$ and $\alpha_p(x)$. The exact form of this function can be found in the technical report. The term $BW\phi$ captures the effects that the parameter estimation error ϕ has on the closed loop error dynamics. As we have done here, we will frequently drop the arguments of W to simplify notation. We will also write $W(t)$ for $W(x(t), y_d^\gamma(t), e(t))$ when we wish to emphasize the dependence of the function on time.

Ideally, we would like to drive $BW\phi \to 0$ as $t \to \infty$ so that we obtain the desired closed-loop error dynamics (14). Recalling from Section II-B that the reference model is designed such that $B^TB = I$, this suggests applying the least-squares cost signal

$$R(t) = \|BW\phi\|_2^2 = \|W\phi\|_2^2 \tag{21}$$

and following the negative gradient of the cost with the following update rule:

$$\dot{\phi} = -W^TW\phi. \tag{22}$$

Least-squares gradient-following algorithms of this sort are well studied in the adaptive control literature [1, Chapter 2]. Since we have $\dot{\theta} = \dot{\phi}$, this suggests that the parameters should also be updated according to $\dot{\theta} = -W^TW\phi$. Altogether, we can represent the tracking and parameter error dynamics with the linear time-varying system

$$\begin{bmatrix}\dot{e}\\ \dot{\phi}\end{bmatrix} = \underbrace{\begin{bmatrix}A+BK & BW(t)\\ 0 & -W^T(t)W(t)\end{bmatrix}}_{A(t)}\begin{bmatrix}e\\ \phi\end{bmatrix}. \tag{23}$$

Letting $X = (e^T, \phi^T)^T$, the solution to this system is given by

$$X(t) = \Phi(t,0)X(0) \tag{24}$$

where for each $t_1, t_2 \in \mathbb{R}^n$ the state transition matrix $\Phi(t_1,t_2)$ is the solution to the matrix differential equation $\frac{d}{dt}\Phi(t,t_2) = A(t)\Phi(t,t_2)$ with intial condition $\Phi(t_2,t_2) = I$, where I is the identity matrix of appropriate dimension. From the adaptive control literature, it is well known that if $W(t)^TW(T)$ is "persistently exciting" in the sense that there exists $\delta > 0$ such that for each $t_0 \geq 0$

$$c_1 I > \int_{t_0}^{t_0+\delta} W^T(t)W(t)dt > c_2 I \tag{25}$$

for some $c_1, c_2 > 0$, then the time varying system (23) is exponentially stable, if $W(t)$ also remains bounded. Intuitively, this condition simply ensures that the regressor term W^TW is "rich enough" during the learning process to drive $\phi \to 0$ exponentially quickly. Observing (20) we also see that if $\phi \to 0$ exponentially quickly then $e \to 0$ exponentially as well. We formalize this point with the following Lemma:

Lemma 1: Let the persistence of excitation condition (25) hold and assume that there exists $C > 0$ such that $\|W(t)\| < C$ for each $t \in \mathbb{R}$. Then there exists $M > 0$ and $\zeta > 0$ such that for each $t_1, t_2 \in \mathbb{R}$

$$\|\Phi(t_1,t_2)\| \leq Me^{-\zeta(t_1-t_2)} \tag{26}$$

with $\Phi(t_1,t_2)$ defined as above.

Proof of this result can be found in the technical report, but variations of this result can be found in standard adaptive control texts [1]. Unfortunately, we do not know the terms in (22) since we don't know ϕ or W so this update rule cannot be directly implemented. In the next section we introduce a model-free update rule for the parameters of the learned controller which approximates the continuous update (22) without requiring direct knowledge of W or ϕ.

B. Sampled-data parameter updates with policy gradients

Hereafter, we will assume that the control supplied to the plant can only be updated every Δt seconds. While this setting provides a more realistic model for many robotic systems, sampling has the unfortunate effect of destroying the affine relationship between the plant's inputs and outputs [11] which was key to the continuous-time design techniques discussed above. Nevertheless, we now introduce a framework for approximately matching the ideal tracking and parameter error dynamics introduced in the previous section in the sampled-data setting using an Euler discretization of the continuous-time reward (21) and a policy-gradient based parameter update rule.

Before introducing our sampled-data control law and adaptation scheme, we first fix notation and discuss a few key assumptions our analysis will employ. To begin we let $t_k = k\Delta t$ for each $k \in \mathbb{N}$ denote the sampling times for the system. Letting $x(\cdot)$ denote the trajectory of the plant, we let $x_k = x(t_k) \in \mathbb{R}^n$ denote the state of the plant at the k-th sample. Similarly, we let $\xi(\cdot)$ denote the trajectory of the outputs and their derivatives as in (10), and we set $\xi_k = \xi(t_k) \in \mathbb{R}^{|\gamma|}$ (not to be confused with the k-th entry of ξ). Next we let $u_k \in \mathbb{R}^m$ denote the input applied to the plant on the interval $[t_k, t_{k+1})$. The parameters for our learned controller will be updated only at the sampling times, and we let $\theta_k \in \mathbb{R}^K$ denote the value of the parameters on $[t_k, t_{k+1})$. We again let $y_d(\cdot)$, $\xi_d(\cdot)$ and $y_d^{(\gamma)}(\cdot)$ denote the desired trajectory for the outputs and their appropriate derivatives, and let $\xi_{d,k} = \xi_d(t_k) \in \mathbb{R}^{|\gamma|}$ and $y_{d,k}^{(\gamma)} = y_d^{(\gamma)}(t_k) \in \mathbb{R}^q$, and $e_k = (\xi_k - \xi_{d,k}) \in \mathbb{R}^{|\gamma|}$. We make the following assumption about the desired output signals and their derivatives:

Assumption 3: The signal $y_d(\cdot)$ is continuous and uniformly bounded. Furthermore, for each $j = 1, \dots, q$ the derivatives $\{\dot{y}_{j,d}(\cdot), \ddot{y}_{j,d}(\cdot), \dots, y_{j,d}^{(\gamma_j)}(\cdot)\}$ are also continuous and uniformly bounded.

Remark 1: Typical convergence proofs in the continuous-time adaptive control literature generally only require that $(y_{j,d}(\cdot), \dot{y}_{1,d}(\cdot), \dots, y_{j,d}^{\gamma_j - 1}(\cdot))$ be continuous and bounded, but these methods also assume that the input to the plant can be updated continuously. In the sampled data setting, we require the continuity of $y_{j,d}^{\gamma_j}(\cdot)$ to ensure that it does not vary too much within a given sampling period.

After sampling the discrete-time tracking error dynamics obey a difference equation of the form

$$e_{k+1} = H_k(x_k, e_k, u_k) \tag{27}$$

where $H_k \colon \mathbb{R}^n \times \mathbb{R}^{|\gamma|} \times \mathbb{R}^q \to \mathbb{R}^{|\gamma|}$ is obtained by integrating the dynamics of the nonlinear system and reference trajectory over $[t_k, t_{k+1})$. Generally, H_k will no longer be affine in the input. However, the relationship is approximately affine for small values of Δt. Indeed, with Assumptions 3 and 5 in place, if we apply the control law

$$u_k = u(\theta_k, x_k, y_{d,k}^{\gamma} + Ke_k), \tag{28}$$

then an Euler discretization of the continuous time error dynamics (20) yields

$$e_{k+1} = e_k + \Delta t(A + BK)e_k + \Delta t B W_k \phi_k + O(\Delta t^2) \tag{29}$$

where we have set $W_k = W(x_k, \xi_k, y_{d,k}^{\gamma} + Ke_k)$. Thus, letting $\bar{A} = (I + \Delta t(A + BK))$, for small $\Delta t > 0$ the continuous-time cost is well approximated by

$$R(t_k) = \|W_k\phi_k\|_2^2 \approx \left\| \frac{e_{k+1} - \bar{A}e_k}{\Delta t} \right\|_2^2 :\, = R_k(x_k, e_k, u_k), \tag{30}$$

where we note that e_k and e_{k+1} are both quantities which can be measured by numerically differentiating the outputs from the plant. Intuitively, the sampled-data cost R_k provides a measure for how well the control u_k matches the desired change in the tracking error (20) over the interval $[t_k, t_{k+1})$.

Next, we add probing noise to the control law (28) to ensure that the input is sufficiently exciting and to enable the use of policy-gradient methods for estimating the gradient of the discrete-time cost signal. In particular, we will draw the input according as $u_k \sim \pi_k(\cdot|\theta_k, x_k, e_k)$, where

$$\pi_k(\cdot|\theta_k, x_k, e_k) = \hat{u}\Big(\theta_k, x_k, y_{d,k}^{\gamma} + K(\xi_{d,k} - \xi_k)\Big) + \mathcal{W}_k \tag{31}$$

and $\mathcal{W}_k = \mathcal{N}(0, \sigma^2 I)$ is additive zero-mean Gaussian noise. Methods for selecting the variance-scaling term σ^2 will be discussed below, however for now it is sufficient to assume that σ^2 is bounded.

With the addition of the random noise we now define

$$J_k(\theta_k) = \mathbb{E}_{u_k \sim \pi_k(\theta_k, x_k, e_k)} R_k(x_k, e_k, u_k), \tag{32}$$

noting that it is also common for policy gradient methods to use an expected "cost-to-go" as the objective. Regardless, using the policy-gradient theorem [29], the gradient of J_k can be written as

$$\nabla_{\theta_k} J_k(\theta_k) = \mathbb{E}_{\pi_k} R(x_k, \xi_k, u_k) \cdot \nabla_{\theta_k} \log \mathbb{P}\{\pi_k(u_k|\theta_k, x_k, e_k)\}$$

where the expectation accounts for randomness due to the input $u_k = \pi_k(u_k|\theta_k, x_k, e_k)$.

Moreover, a noisy, unbiased estimate of ∇J_k is given by

$$\hat{J}_k = R(x_k, \xi_k, u_k) \nabla_{\theta_k} \log\left(\mathbb{P}\{\pi(u_k|\theta_k, \theta_k, x_k, e_k)\}\right) \tag{33}$$

where $u_k = \pi_k$ and is the actual input applied to the plant over the k-th time interval. Recall that $R_k(x_k, e_k, u_k)$ can be directly calculated using e_k, e_{k+1} and (30), and $\nabla_{\theta_k} \mathbb{P}\{\log(\pi(u_k|\theta_k, s_k))\}$ can also be computed since the derivatives of $\hat{u}$ (and thus of $\log \mathbb{P}\{\pi_k\}$) are known to us. Thus, $\hat{J}_k$ can be computed using values that we have assumed

we can measure. However, since the input u_k is random, the gradient estimate is drawn according to

$$\hat{J}_k \sim \Delta\hat{J}_k(\cdot|\theta_k, x_k, e_k) \tag{34}$$

where the random variable is constructed using the relationship (33). Using our estimate of the gradient for the discrete-time reward we propose the following noisy update rule for the parameters of our learned controller:

$$\theta_{k+1} = \theta_k - \Delta t\hat{J}_k \tag{35}$$

Putting it all together, the sampled-data stochastic version of our error dynamics becomes

$$\begin{aligned} e_{k+1} &= e_k + H_k(x_k, e_k, u_k) \\ \phi_{k+1} &= \phi_k - \Delta t\hat{J}_k \end{aligned} \tag{36}$$

where $u_k = \pi_k$ and $\hat{J}_k$ is calculated as in (33). We make the following Assumptions about this stochastic process:

Assumption 4: There exists a constant $C > 0$ such that $\sup_{k\geq 0}\|w_k\| < C$ almost surely.

Assumption 5: There exists a constant $C > 0$ such $\sup_{k\geq 0}\|x_k\| < C$ and $\sup_{k\geq 0}\|\theta_k\| < C$ almost surely.

Assumption 4 ensures that the additive noise does not drive the state to be unbounded during a single sampling interval, while Assumption 5 ensures that the gradient estimate does not become undefined during the learning process. These important technical assumptions are common in the theory of stochastic approximations [8], and allow us to characterize the estimator for the gradient as follows:

Lemma 2: Let Assumptions 3-5 hold. Then $\Delta\hat{J}_k(\cdot|\theta_k, x_k, e_k)$ is a sub-Gaussian distribution where

$$\mathbb{E}[\hat{J}_k] = W_k^T W_k\phi_k + O(\Delta t(1+\sigma+\sigma^2)) \tag{37}$$

and

$$\left\|\hat{J}_k(\cdot|\theta_k, x_k, e_k)\right\|_{\psi_2} = O\left(\frac{1}{\sigma}\right). \tag{38}$$

The Lemma demonstrates a trade-off between the bias and variance of the gradient estimate that has been observed in the reinforcement learning literature [15, 30]. Specifically, the bias of the gradient estimate decreases as $\sigma^2 \to 0$ but this causes the gradient of the estimator to blow up, as indicated by the increasing sub-Gaussian norm. However, the bias of the gradient estimate has a term which is $O(\Delta t)$ which does not depend on the amount of noise added to the system. This term comes from the fact that we have resorted to using a finite difference approximation (30) to approximate the gradient of the continuous-time reward in the sampled data setting. Due to this inherent bias, little is gained by decreasing σ^2 past a certain point. Next, we analyze the overall behavior of (36).

C. Convergence analysis

The main idea behind our analysis is to model our sampled-data error dynamics (36) as a perturbation to the idealized continuous-time error dynamics (23), as is commonly done in the stochastic approximation literature [8]. Under the assumption that W^TW is persistently exciting, the nominal continuous time dynamics are exponentially stable and we observe that the total perturbation accumulated over each sampling interval decays exponentially as time goes on. Due to space constraints, we outline the main points of the analysis here but leave the details to the technichal report.

Our analysis makes use of the piecewise-linear curve $\bar{\phi}\colon \mathbb{R} \to \mathbb{R}^K$ which is constructed by interpolating between ϕ_k and ϕ_{k+1} along the interval $[t_k, t_{k+1})$. That is, we define

$$\bar{\phi}(t) = \left(\frac{t_{k+1}-t}{\Delta t}\right)\phi_k + \left(\frac{t-t_k}{\Delta t}\right)\phi_{k+1} \quad \text{if } t \in [t_k, t_{k+1}).$$

Combining the tracking and interpolated tracking error into the state $X = (e^T, \phi^T)^T$ we may write

$$\frac{d}{dt}X(t) = A(t)X(t) + \delta(t) \tag{39}$$

where for each $t \in \mathbb{R}$ the dynamics matrix $A(t)$ constructed as in (23) and the disturbance $\delta\colon \mathbb{R} \to \mathbb{R}^{|\gamma|+K}$ captures the deviation from the idealized continuous dynamics caused at each instance of time due the sampling, additive noise, and the process of interpolating the parameter error. Again letting $\Phi(t,\tau)$ denote the solution to $\frac{d}{dt}\Phi(t,\tau) = A(t)\Phi(t,\tau)$ with initial condition $\Phi(s,s) = I$, for each $t, s \in \mathbb{R}$ we have that

$$X(t) = \Phi(t,0)X(0) + \int_0^t \Phi(t,\tau)\delta(\tau)d\tau \tag{40}$$

Now, if we let $X_k = X(t_k)$ for each $k \in \mathbb{N}$ we can instead write

$$X_k = \Phi(t_k,0)X_0 + \sum_{i=1}^{k-1}\Phi(t_k,t_{i+1})\underbrace{\int_{t_i}^{t_{i+1}}\Phi(t_{i+1},\tau)\delta(\tau)d\tau}_{\delta_k}, \tag{41}$$

where the term $\delta_k \in \mathbb{R}^{|\gamma|+K}$ is the total disturbance accumulated over the interval $[t_k, t_{k+1})$. We separate the effects the distubance has on the tracking and error dynamics by letting $\delta_k^e \in \mathbb{R}^{|\gamma|}$ denote the first $|\gamma|$ elements of δ_k and letting $\delta_k^\phi \in \mathbb{R}^K$ denote the remaining entries. On the interval $[t_k, t_{k+1})$ the disturbance $\delta(t)$ can be written as a function of u_k, x_k and e_k. Since u_k is a random function of x_k, for fixed x_k, e_k and θ_k, the two elements of δ_k are distributed according to

$$\delta_k^e \sim \Delta_k^e(\cdot|\theta_k, x_k, e_k) \quad \text{and} \quad \delta_k^\phi \sim \Delta_k^\phi(\cdot|\theta_k.x_k, e_k). \tag{42}$$

These random variables are constructed by integrating the distrubance over $[t_k, t_{k+1})$ and an explicit representation of these variable can be found in the technical report, where proof of the following result can also be found.

Lemma 3: Let Assumptions 3-5 hold. Then $\Delta_k^e(\cdot|\theta_k, x_k, e_k)$ and $\Delta_k^\phi(\cdot|\theta_k, x_k, e_k)$ are sub-Gaussian

random variables where

$$\|E[\Delta_k^e(\cdot|\theta_k, x_k, e_k)]\|_2 = O(\Delta t^2(1+\sigma)) \tag{43}$$

$$\|E[\Delta_k^\phi(\cdot|\theta_k, x_k, e_k)]\|_2 = O(\Delta t^2(1+\sigma+\sigma^2)) \tag{44}$$

$$\|\Delta_k^e(\cdot|\theta_k, x_k, e_k)\|_{\psi_2} = O(\Delta t\sigma) \tag{45}$$

$$\|\Delta_k^\phi(\cdot|\theta_k, x_k, e_k)\|_{\psi_2} = O\bigg(\frac{\Delta t}{\sigma}\bigg). \tag{46}$$

Next, for each $k \in \mathbb{N}$ we put $\varepsilon_k^e = E[\Delta_k^e(\cdot|\theta_k, x_k, e_k)] \in \mathbb{R}^{|\gamma|}$, $\varepsilon_k^\phi = E[\Delta_k^\phi(\cdot|\theta_k, x_k, e_k)] \in \mathbb{R}^K$ and then define the zero-mean random variables $\mathcal{M}_k^e = \Delta_k^e(\cdot|\theta_k, x_k, e_k) - \varepsilon_k^e$ and $\mathcal{M}_k^\phi = \Delta_k^\phi(\cdot|\theta_k, x_k, e_k) - \varepsilon_k^\phi$. Our overall discrete-time process can then be written as

$$X_k = \Phi(t_k, 0)X_0 + \sum_{i=0}^{k-1} \Phi(t_k, t_{i+1})(\varepsilon_i + \mathcal{M}_i). \tag{47}$$

where $\varepsilon_k \in \mathbb{R}^{|\gamma|+K}$ is constructed by stacking ε_k^e on top of ε_k^ϕ and $\mathcal{M}_k$ is constructed by stacking $\mathcal{M}_k^e$ on top of $\mathcal{M}_k^\phi$. Now if we assume that W^TW is persistently exciting, then for each $k_1, k_2 \in \mathbb{N}$ we have

$$\|\Phi(t_{k_1}, t_{k_2})\| \leq Me^{-\zeta\Delta t(k_1-k_2)} = M\rho^{k_1-k_2} \tag{48}$$

where $M > 0$ and $\zeta > 0$ are as in Lemma 1 and we have put $\rho = e^{-\zeta\Delta t} < 1$. Thus, under this assumption we may use the triangle inequality to bound

$$|X_k| \leq M\bigg(\rho^k|X_0| + \sum_{i=0}^{k-1}\rho^{k-i}|\varepsilon_k| + |\sum_{i=0}^{k-1}\rho^{k-i}\mathcal{M}_k|\bigg). \tag{49}$$

Thus, when W^TW is persistently exciting we see that the effects of the disturbance accumulated at each time step decays exponentially as time goes on, along with the effects of the initial tracking and parameter error. A full proof for the following Theorem is given in the technichal report, but the main idea is to use properties of geometric series to bound $\sum_{i=0}^{k-1}\rho^{k-i}|\varepsilon_k|$ over time and to use the concentration inequality from [16, Theorem 2.6.3] to bound the deviation of $|\sum_{i=0}^{k-1}\rho^{k-i}\mathcal{M}_k|$.

Theorem 1: Let Assumptions 3-5 hold. Further assume that W^TW is persistently exciting and let $M > 0$ and $\zeta > 0$ be defined as in Lemma 1. Then there exists numerical constants $C_1 > 0$ and $C_2 > 0$ such that

$$|\mathbb{E}[X_k]| \leq M\rho^k|X_0| + MC_1\frac{\Delta t(1+\sigma+\sigma^2)}{\zeta} \tag{50}$$

and for each $\lambda > 0$ with probability $1-\lambda$ we have

$$|X_k - \mathbb{E}[X_k]| \leq C_2M\sqrt{\frac{\Delta t\ln\big(\frac{2}{\lambda}\big)}{\zeta\sigma^2}} \tag{51}$$

Despite the high variance of the simple policy gradient parameter update analyzed so far, the Theorem demonstrates that with high probability our tracking and parameter errors concentrate around the origin. As Δt decreases, the bias introduced by the sampling and additive noise diminish, as does the radius of our high-probability bound. These bounds also become tighter as the exponential rate of decay for the idealized continuous time dynamics increases. The Theorem again displays the trade-off between the bias and variance of the learning scheme observed in Section 3. However, here we still observe in equation (50) that the bias introduced by the noise is relatively small, meaning σ^2 does not have to be made prohibitively small so as to degrade the bound in (51).

D. Variance Reduction via Baslines

It is common for policy gradients to be implemented with a *baseline* [31]. In this case, the gradient estimator in (33) may become biased, though it often has lower variance [7, 32]. The expression with a baseline is

$$\hat{J}_k = \big(R_k(x_k, \xi_k, u_k) - S_k(x_k, \xi_k, u_k)\big) \cdot \nabla_{\theta_k}\log\big(\mathbb{P}\{\pi(u_k|\theta_k, \theta_k, x_k, e_k)\}\big), \tag{52}$$

where $S_k(x_k, \xi_k, u_k)$ is an estimate of $R(x_k, \xi_k, u_k)$. If S_k does not depend on u_k then the addition of the baseline does not add any bias to the gradient estimate [7]. For example, in our numerical example below we use a simple sum-of-past-rewards baseline by setting $S_k = \sum_{i=0}^{k-1} R_i$, where R_i is the i-th reward recorded. We consider it a matter of future work to rigorously study the effects of this an other common baselines from the reinforcement learned literature within the theoretical framework we have developed.

IV. Numerical Example

Our numerical example examines the application of our method to the double pendulum depicted in Figure 1 (a), whose dynamics can be found in [33]. With a slight abuse of notation, the system has generalized coordinates $q = (\theta_1, \theta_2)$ which represent the angles the two arms make with the vertical. Letting $x = (x_1, x_2, x_3, x_4) = (q, \dot{q})$, the system can be represented with a state-space model of the form (1) where the angles of the two joints are chosen as outputs. It can be shown that the vector relative degree is $(2, 2)$, so the system can be completely linearized by state feedback.

The dynamics of the system depend on the parameters m_1, m_2, l_1, l_2 where m_i is the mass of the i-th link and l_i its length. For the purposes of our simulation, we set the true parameters for the plant to be $m_1 = m_2 = l_1 = l_2 = 1$. However, to set-up the learning problem, we assume that we have inaccurate measurements for each of these parameters, namely, $\hat{m}_1 = \hat{m}_2 = \hat{l}_1 = \hat{l}_2 = 1.3$. That is, each estimated parameter is scales to 1.3 times its true value. Our nominal model-based linearizing controller u_m is constructed by computing the linearizing controller for the dynamics model which corresponds to the inaccurate parameter estimates. The learned component of the controller is then constructed by using radial basis functions to populate the entries of $\{\beta_k\}_{k=1}^{K_1}$ and $\{\alpha_k\}_{k=1}^{k_2}$. In total, 250 radial basis functions were used.

For the online leaning problem we set the sampling interval to be $\Delta t = 0.05$ seconds and set the level of

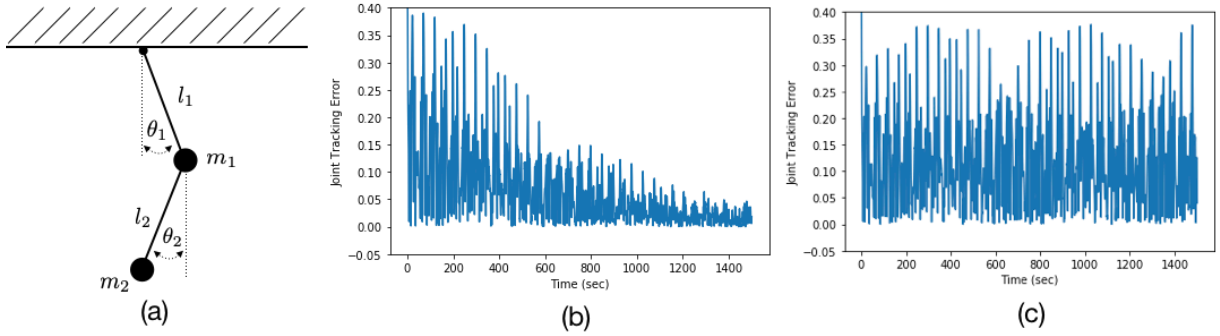

Fig. 1: (a) Schematic representation of the double pendulum model used in the simulations study. (b) The norm of the tracking error for the adaptive learning scheme (c) The tracking error for the nominal model-based controller with no learning.

probing noise at $\sigma^2 = 0.1$. The reward was regularized using an average sum-of-rewards baseline as described in III-D. The reference trajectory for each of the output channels were constructed by summing together sinusoidal functions whose frequencies are non-integer multiples of each other to ensure that the entire region of operation was explored. The feedback gain matrix $K \in \mathbb{R}^{2\times 4}$ was designed so that each of the eigenvalues of $(A + BK)$ are equal to -1.5, where $A \in \mathbb{R}^{4\times 4}$ and $B \in \mathbb{R}^{4\times 2}$ are the appropriate matricies in the reference model for the system.

Figure 1 (b) shows the norm of the tracking error of the learning scheme over time while Figure 1 (c) shows the norm of the tracking error for the nominal model-based controller with no learning. Note that the learning-based approach is able to steadily reduce the tracking error over time while keeping the system stable.

V. Conclusion

This paper developed an adaptive framework which employs model-free policy-gradient parameter update rules to construct a feedback-linearization based tracking controller for systems with unknown dynamics. We combined analysis techniques from the adaptive control literature and theory of stochastic approximations to provide high-confidence tracking guarantees for the closed loops system, and demonstrated the utility of the framework through a simulation experiment. Beyond the immediate utility of the proposed framework, we believe the analysis tools we developed provide a foundation for studying the use of reinforcement learning algorithms for online adaptation.

References

[1] S. S. Sastry and A. Isidori. "Adaptive control of linearizable systems". *IEEE Transactions on Automatic Control* 34.11 (1989).

[2] S. Sastry. *Nonlinear systems: analysis, stability, and control*. Vol. 10. Springer Science & Business Media, 1999.

[3] J.-J. E. Slotine and W. Li. "On the adaptive control of robot manipulators". *The international journal of robotics research* 6.3 (1987).

[4] J. Schulman et al. "Trust region policy optimization". *International conference on machine learning*. 2015.

[5] J. Schulman et al. "Proximal policy optimization algorithms". *arXiv preprint arXiv:1707.06347* (2017).

[6] T. P. Lillicrap et al. "Continuous control with deep reinforcement learning". *arXiv preprint arXiv:1509.02971* (2015).

[7] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. 2018.

[8] V. S. Borkar. *Stochastic approximation: a dynamical systems viewpoint*. Vol. 48. Springer, 2009.

[9] R. Munos. "Policy gradient in continuous time". *Journal of Machine Learning Research* 7.May (2006).

[10] K. Doya. "Reinforcement learning in continuous time and space". *Neural computation* 12.1 (2000).

[11] J. Grizzle and P. Kokotovic. "Feedback linearization of sampled-data systems". *IEEE Transactions on Automatic Control* 33.9 (1988).

[12] T. Westenbroek et al. "Feedback Linearization for Unknown Systems via Reinforcement Learning". *arXiv preprint arXiv:1910.13272* (2019).

[13] E. B. Kosmatopoulos and P. A. Ioannou. "A switching adaptive controller for feedback linearizable systems". *IEEE Transactions on automatic control* 44.4 (1999).

[14] J. J. Craig, P. Hsu, and S. S. Sastry. "Adaptive control of mechanical manipulators". *The International Journal of Robotics Research* 6.2 (1987).

[15] T. Zhao et al. "Analysis and improvement of policy gradient estimation". *Advances in Neural Information Processing Systems*. 2011.

[16] R. Vershynin. *High-dimensional probability: An introduction with applications in data science*. Vol. 47. Cambridge university press, 2018.

[17] T. Westenbroek et al. "Thechnical Report: Adaptive Control for Linearizable Systems Using On-Policy Reinforcement Learning". *arXiv preprint* (2020).

[18] E. B. Kosmatopoulos and P. A. Ioannou. "Robust switching adaptive control of multi-input nonlinear systems". *IEEE transactions on automatic control* 47.4 (2002).

[19] C. P. Bechlioulis and G. A. Rovithakis. "Robust adaptive control of feedback linearizable MIMO nonlinear systems with prescribed performance". *IEEE Transactions on Automatic Control* 53.9 (2008).

[20] K.-S. Hwang, S.-W. Tan, and M.-C. Tsai. "Reinforcement learning to adaptive control of nonlinear systems". *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 33.3 (2003).

[21] A. Y. Zomaya. "Reinforcement learning for the adaptive control of nonlinear systems". *IEEE transactions on systems, man, and cybernetics* 24.2 (1994).

[22] J. Umlauft et al. "Feedback linearization using Gaussian processes". *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE. 2017.

[23] J. Umlauft and S. Hirche. "Feedback Linearization based on Gaussian Processes with event-triggered Online Learning". *IEEE Transactions on Automatic Control* (2019).

[24] C. Finn, P. Abbeel, and S. Levine. "Model-agnostic meta-learning for fast adaptation of deep networks". *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017.

[25] A. Santoro et al. "Meta-learning with memory-augmented neural networks". *International conference on machine learning*. 2016.

[26] R. Vilalta and Y. Drissi. "A perspective view and survey of meta-learning". *Artificial intelligence review* 18.2 (2002).

[27] A. Nagabandi et al. "Learning to adapt in dynamic, real-world environments through meta-reinforcement learning". *arXiv preprint arXiv:1803.11347* (2018).

[28] M. Andrychowicz et al. "Learning dexterous in-hand manipulation". *The International Journal of Robotics Research* 39.1 (2020).

[29] R. S. Sutton et al. "Policy gradient methods for reinforcement learning with function approximation". *Advances in neural information processing systems*. 2000.

[30] D. Silver et al. “Deterministic policy gradient algorithms”. 2014.
[31] R. J. Williams. “Simple statistical gradient-following algorithms for connectionist reinforcement learning”. *Machine learning* 8.3-4 (1992).
[32] E. Greensmith, P. L. Bartlett, and J. Baxter. “Variance reduction techniques for gradient estimates in reinforcement learning”. *Journal of Machine Learning Research* 5.Nov (2004).
[33] T. Shinbrot et al. “Chaos in a double pendulum”. *American Journal of Physics* 60.6 (1992).