2021 60th IEEE Conference on Decision and Control (CDC)
December 13-15, 2021. Austin, Texas

On the Search for Feedback in Reinforcement Learning

Ran Wang, Karthikeya S. Parunandi, Aayushman Sharma, Raman Goyal, Suman Chakravorty

2021 60th IEEE Conference on Decision and Control (CDC) | 978-1-6654-3659-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/CDC45484.2021.9683350

***Abstract*— The problem of Reinforcement Learning (RL) in an unknown nonlinear dynamical system is equivalent to the search for an optimal feedback law utilizing the simulations/ rollouts of the unknown dynamical system. Most RL techniques search over a complex global nonlinear feedback parametrization making them suffer from high training times as well as variance. Instead, we advocate searching over a local feedback representation consisting of an open-loop sequence, and an associated optimal linear feedback law completely determined by the open-loop. We show that this alternate approach results in highly efficient training, the answers obtained are repeatable and hence reliable, and the resulting closed performance is superior to global state of the art RL techniques. Finally, if we replan, whenever required, which is feasible due to the fast and reliable local solution, allows us to recover global optimality of the resulting feedback law.**

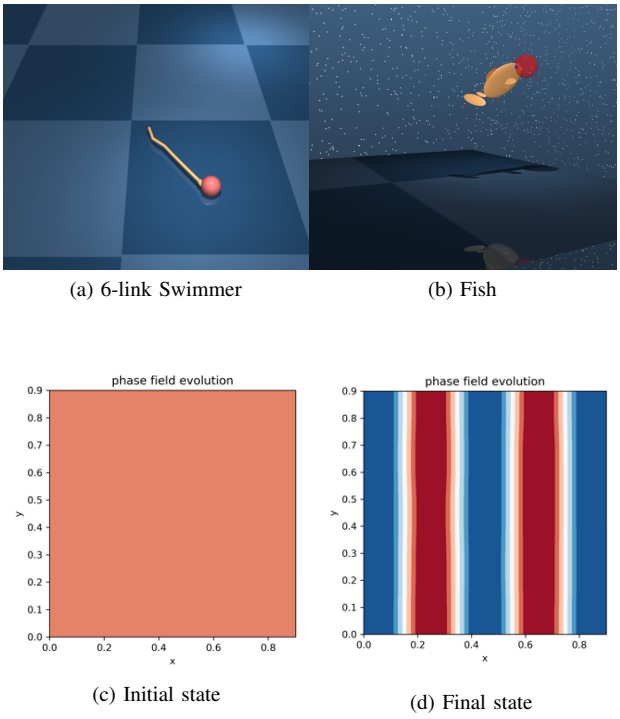
***Index Terms*— RL, Optimal control, Nonlinear systems, Feedback control**

I. Introduction

The control of an unknown (stochastic) dynamical system has a rich history in the control system literature [12], [9]. The stochastic adaptive control literature mostly addresses Linear Time Invariant (LTI) problems. The optimal control of an unknown nonlinear dynamical system with continuous state space and continuous action space is a significantly more challenging problem. The 'curse of dimensionality' associated with Dynamic Programming (DP) makes solving such problems computationally intractable, in general.

The last several years have seen significant progress in deep neural networks based reinforcement learning approaches for controlling unknown dynamical systems, with applications in many areas like playing games [26], locomotion [19] and robotic hand manipulation [14]. A number of new algorithms that show promising performance have been proposed [36], [24], [25] and various improvements and innovations have been continuously developed. However, despite excellent performance on many tasks, reinforcement learning (RL) is still considered very data intensive. The training time for such algorithms is typically really large. Moreover, the techniques suffer from high variance and reproducibility issues [8]. While there have been some attempts to improve the efficiency [6], a systematic approach is still lacking. *The issues with RL can be attributed to the typically complex parametrization of the global feedback policy, and the related fundamental question of what this feedback parametrization ought to be?*

In this paper, we argue that for RL to be a) efficient in training, b) reliable in its result, and c) have robust performance to noise, one needs to use a local feedback parametrization as opposed to a global parametrization. Further, this local feedback parametrization consists of an open-loop control sequence combined with a linear feedback law around the nominal open-loop sequence. Searching over this parametrization is highly efficient when compared to the global RL search, can be shown to reliably converge to the global optimum, while having performance that is superior to the global RL solution. In particular, this search is sufficiently fast and reliable that one can recover the optimal global feedback law by replanning whenever necessary. The sole caveat is that these claims are true for a deterministic, but unknown, system. However, we show that: 1) theoretically, the deterministic optimal feedback law is near optimal to fourth order in a small noise parameter to the optimal stochastic law, and 2) empirically, RL methods have great difficulty in learning on stochastic systems, so much so that most RL algorithms really only find a feedback law for the deterministic system.



(a) 6-link Swimmer (b) Fish

(c) Initial state (d) Final state

Fig. 1: Models controlled in this paper including multi-body systems with fluid-structure interactions, and a material microstructure model governed by the Allen-Cahn phase field partial differential equation.

The authors are with the Department of Aerospace Engineering, Texas A&M University, College Station, TX 77843 USA. { rwang0417, s.parunandi, aayushmansharma, ramaniitrgoyal92, schakrav} @tamu.edu

Related work: The solution approaches to the problem of controlling unknown dynamical systems can be divided into two broad classes, local and global.

The most computationally efficient among these techniques are "local" trajectory-based methods such as differential dynamic programming (DDP), [11], [30], which quadratizes the dynamics and the cost-to-go function around a nominal trajectory, and the iterative linear quadratic regulator (iLQR), [31], [17], which only linearizes the dynamics, and thus, is much more efficient. Our approach, the Decoupled Data based Control (D2C), requires the model-free/ data-based solution of the open-loop optimization along with a local linear feedback, and thus, we use iLQR with an efficient randomized least squares procedure to estimate the linear system parameters, using simulated rollouts of the system. This local approach to control unknown systems has been explored before [28], [15], however, it was never established that this local approach to RL is highly efficient as well as reliable compared to global approaches, while being superior in performance in terms of robustness to noise. Further, we establish that such approaches can recover global optimality when coupled with replanning, whenever necessary, which becomes feasible because of the highly efficient and reliable local search that is guaranteed to converge to a globally optimum open loop, and the associated optimal linear feedback law.

Global methods, more popularly known as approximate dynamic programming [23], [1] or reinforcement learning (RL) methods [27], seek to improve the control policy by repeated interactions with the environment while observing the system's responses. The repeated interactions, or learning trials, allow these algorithms to compute the solution of the dynamic programming problem (optimal value/Q-value function or optimal policy) either by constructing a model

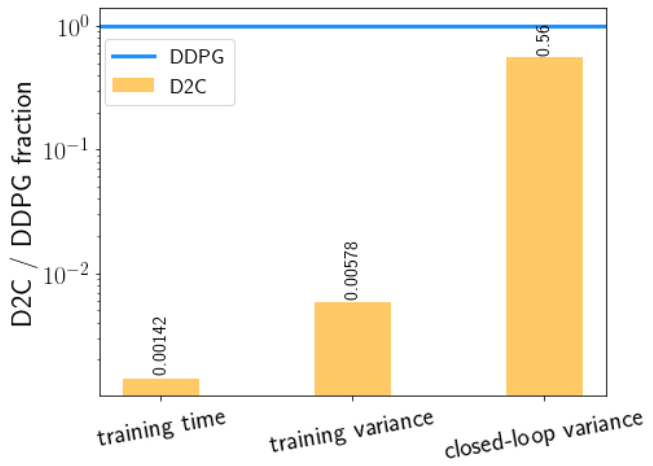


Fig. 2: Local RL approach (D2C) vs global RL approach (DDPG). The statistics shown above, which is found by averaging over all the models simulated, shows that the local approach is highly efficient (training time), reliable (training variance), while still having superior closed loop performance (closed loop variance), when compared to the global approach.

of the dynamics (model-based) [4], [13], [20], or directly estimating the control policy (model-free) [27], [18], [24]. Standard RL algorithms are broadly divided into value-based methods, like Q-learning, and policy-based methods, like policy gradient algorithms. Recently, function approximation using deep neural networks has significantly improved the performance of reinforcement learning algorithms, leading to a growing class of literature on 'deep reinforcement learning' [36], [24], [25]. Despite the success, the training time required by such methods, and their variance, remains prohibitive. Our primary contribution in this paper is to show that performing RL via a local feedback parametrization is highly efficient and reliable when compared to the global approaches, while being superior in terms of robustness to noise, and further that global optimality can be recovered by replanning, which is made feasible via the fast and reliable local planner. As an extension of [35], [34], the open-loop design in this paper uses iLQR instead of gradient descent, which greatly improves the training efficiency, reliability and the closed-loop performance. Further, the fourth order near optimality to the optimal stochastic control law, and the global optimality of the open-loop solution by iLQR is also established. Empirically, we show that local schemes such as D2C are preferable to global RL schemes due to their efficiency, reliability and closed-loop performance.

The rest of the paper is organized as follows. In Section II, the basic problem formulation is outlined. In Section III, the main decoupling results which solve the stochastic optimal control problem in a 'decoupled open loop-closed loop' fashion are briefly summarized. In Section IV, we outline the iLQR based decoupled data based control algorithm. In Section V, we test the proposed approach using typical benchmarking examples with comparisons to a state of the art RL technique.

II. Problem Formulation

Consider the following discrete time nonlinear stochastic dynamical system: $x_{t+1} = h(x_t, u_t, w_t)$,where $x_t \in \mathbb{R}^{n_x}$, $u_t \in \mathbb{R}^{n_u}$ are the state measurement and control vector at time t, respectively. The process noise w_t is assumed as zero-mean, uncorrelated Gaussian white noise, with covariance W. The *optimal stochastic control* problem is to find the the control policy $\pi^o = \{\pi_1^o, \pi_2^o, \cdots, \pi_{N-1}^o\}$ such that the expected cumulative cost is minimized, i.e., $\pi^o = \arg\min_\pi \tilde{J}^\pi(x)$, where, $\tilde{J}^\pi(x) = \mathbb{E}_\pi \left[\sum_{t=1}^{N-1} c(x_t, u_t) + c_N(x_N) | x_1 = x\right]$, $u_t = \pi_t(x_t)$, $c(\cdot,\cdot)$ is the instantaneous cost function, and $c_N(\cdot)$ is the terminal cost function. In the following, we assume that the initial state x_1 is fixed, and denote $\tilde{J}^\pi(x_1)$ simply as $\tilde{J}^\pi$.

III. A Near-Optimal Decoupling Principle

We summarize the key theoretical results for a decoupling principle in stochastic optimal control. All the details can be found in [33].

Let the dynamics be given by:

$$x_t = x_{t-1} + \bar{f}(x_{t-1})\Delta t + \bar{g}(x_{t-1})u_t\Delta t + \epsilon\omega_t\sqrt{\Delta t}, \quad (1)$$

where ω_t is a white noise sequence, and the sampling time Δt is small enough that the $O(\Delta t^\alpha)$ terms are negligible for $\alpha > 1$. The noise term above stems from Brownian motion, and hence the $\sqrt{\Delta t}$ factor. Further, the incremental cost function $c(x,u)$ is given as: $c(x,u) = \bar{l}(x)\Delta t + \frac{1}{2}u'\bar{R}u\Delta t$. Then, we have the following results. Given sufficient regularity, any feedback policy can then be represented as: $\pi_t(x_t) = \bar{u}_t + K_t^1\delta x_t + \delta x_t' K_t^2 \delta x_t + \cdots$, where $\bar{u}_t$ is the nominal action with associated nominal state $\bar{x}_t$, i.e., action under zero noise, and $K_t^1, K_t^2, \cdots$ represent the linear and higher order feedback gains acting on the state deviation from the nominal: $\delta x_t = x_t - \bar{x}_t$, due to the noise.

Proposition 1: The cost function of the optimal stochastic policy, J_t, and the cost function of the "deterministic policy applied to the stochastic system", φ_t, satisfy: $J_t(x) = J_t^0(x) + \epsilon^2 J_t^1(x) + \epsilon^4 J_t^2(x) + \cdots$, and $\varphi_t(x) = \varphi_t^0(x) + \epsilon^2\varphi_t^1(x) + \epsilon^4\varphi_t^2(x) + \cdots$. Furthermore, $J_t^0(x) = \varphi_t^0(x)$, and $J_t^1 = \varphi_t^1(x)$, for all t, x.

Remark 1: The result above shows that the cost due to the nominal action, $J_t^0(x)$ and the cost due to the linear feedback action, $J_t^1(x)$, are the same for the optimal deterministic and optimal stochastic policies, when acting on the stochastic system, given they both start at state x at time t. This essentially means that the optimal deterministic policy and the optimal stochastic policy agree locally up to order $O(\epsilon^4)$. An important practical consequence of Proposition 1 is that we can get $O(\epsilon^4)$ near-optimal performance, by wrapping the optimal linear feedback law around the nominal control sequence ($u_t = \bar{u}_t + K_t^1\delta x_t$),where δx_t is the state deviation from the nominal $\bar{x}_t$ state, and replanning the nominal sequence when the deviation is sufficiently large. This is similar to the event driven MPC philosophy of [7], [16]. We note that the open-loop ($\bar{u}_t$) design is independent of the closed loop design (K_t^1) which suggests the following "decoupled" procedure to find the optimal feedback law (locally).

Open Loop Design. First, we design an optimal (open-loop) control sequence $\bar{u}_t^*$ for the noiseless system by solving $(\bar{u}_t^*)_{t=1}^{N-1} = \arg\min_{(\bar{u}_t)_{t=1}^{N-1}} \sum_{t=1}^{N-1} c(\bar{x}_t, \bar{u}_t) + c_N(\bar{x}_N)$,with $\bar{x}_{t+1} = f(\bar{x}_t) + g(\bar{x}_t)\tilde{u}_t$, where $\mathcal{F}(x) = x + \bar{f}(x)\Delta t$ and $\mathcal{G}(x) = \bar{g}(x)\Delta t$ with reference to Eq. 1. The global optimum for this open-loop problem can be solved by satisfying the necessary conditions of optimality, and we propose doing so using the iLQR algorithm in the next Section.

Closed Loop Design. The linear feedback gain K_t^1 is calculated in a slightly different fashion and may be done as shown in the following result. In the following, $A_t = \frac{\partial \mathcal{F}}{\partial x}|_{\bar{x}_t} + \frac{\partial \mathcal{G}\bar{u}_t}{\partial x}|_{\bar{x}_t}$, $B_t = \mathcal{G}(\bar{x}_t)$, $L_t^x = \frac{\partial l}{\partial x}|'_{\bar{x}_t}$ and $L_N^{xx} = \nabla^2_{xx} l|_{\bar{x}_t}$. Let $\phi_t(x_t)$ denote the optimal cost-to-go of the detrministic problem, i.e., Eq 1 with $\epsilon = 0$.

Proposition 2: Given an optimal nominal trajectory $(\bar{x}_t, \bar{u}_t)$, the backward evolutions of the first and second derivatives, $G_t = \frac{\partial \phi_t}{\partial x}|'_{\bar{x}_t}$ and $P_t = \nabla^2_{xx}\phi_t|_{\bar{x}_t}$, of the optimal cost-to-go function $\phi_t(x_t)$, initiated with the terminal boundary conditions $G_N = \frac{\partial c_N(x_N)}{\partial x_N}|'_{\bar{x}_N}$ and $P_N = \nabla^2_x c_N|_{\bar{x}_N}$ respectively, are as follows:

$$G_t = L_t^x + G_{t+1}A_t, \tag{2}$$

$$P_t = L_t^{xx} + A_t' P_{t+1} A_t - K_t' S_t K_t + G_{t+1} \otimes \tilde{R}_{t,xx} \tag{3}$$

for $t = \{0, 1, ..., N-1\}$, where,

$$S_t = (R_t + B_t' P_{t+1} B_t), \tag{4}$$

$$K_t^1 = -S_t^{-1}(B_t' P_{t+1} A_t + (G_{t+1} \otimes \tilde{R}_{t_{xu}})'), \tag{5}$$

$\tilde{R}_{t,xx} = \nabla^2_{xx}\mathcal{F}(x_t)|_{\bar{x}_t} + \nabla^2_{xx}\mathcal{G}(x_t)|_{\bar{x}_t,\bar{u}_t}$,
$\tilde{R}_{t,xu} = \nabla^2_{xu}(\mathcal{F}(x_t) + \mathcal{G}(x_t)u_t)|_{\bar{x}_t,\bar{u}_t}$, where ∇^2_{xx} represents the Hessian of a vector-valued function w.r.t x and $\otimes$ denotes the tensor product.

IV. Decoupled Data based Control (D2C)

This section presents our decoupled data-based control (D2C) algorithm. We outline the open-loop and closed-loop design components of D2C below.

A. Open-Loop Trajectory Design

We present an iLQR [17] based method to solve the open-loop optimization problem. iLQR typically requires the availability of analytical system Jacobian, and thus, cannot be directly applied when such analytical gradient information is unavailable (much like Nonlinear Programming software whose efficiency depends on the availability of analytical gradients and Hessians). In order to make it an (analytical) model-free algorithm, it is sufficient to obtain estimates of the system Jacobians from simulations, and a sample-efficient randomized way of doing so is described in the following subsection. Since iLQR is a well-established framework, we skip the details and instead present pseudocode in algorithm 1. *Please refer to [21] [33] to see why the iLQR scheme is particularly attractive and can be guaranteed to converge to a global minimum for the open-loop problem even though the problem is non-convex. We also note that any (analytical) model-free open-loop design technique can be swapped for iLQR in this step.*

1) Estimation of Jacobians: Linear Least Squares by Central Difference (LLS-CD): Using Taylor's expansions of 'h' (for generality, h is the non-linear model of Section 2) about the nominal trajectory $(\bar{x}_t, \bar{u}_t)$ on both the positive and the negative sides, we obtain the following central difference equation: $h(\bar{x}_t + \delta x_t, \bar{u}_t + \delta u_t) - h(\bar{x}_t - \delta x_t, \bar{u}_t - \delta u_t)$ $= 2\begin{bmatrix} h_{x_t} & h_{u_t} \end{bmatrix}\begin{bmatrix} \delta x_t \\ \delta u_t \end{bmatrix} + O(\|\delta x_t\|^3 + \|\delta u_t\|^3)$. Multiplying by $\begin{bmatrix} \delta {x_t}^T & \delta {u_t}^T \end{bmatrix}$ on both sides to the above equation and apply standard Least Square method:

$$\begin{bmatrix} h_{x_t} & h_{u_t} \end{bmatrix} = H\delta Y_t^T(\delta Y_t \delta Y_t^T)^{-1}$$

$$H = \begin{bmatrix} h(\bar{x}_t + \delta x_t^1, \bar{u}_t + \delta u_t^1) - h(\bar{x}_t - \delta x_t^1, \bar{u}_t - \delta u_t^1) \\ h(\bar{x}_t + \delta x_t^2, \bar{u}_t + \delta u_t^2) - h(\bar{x}_t - \delta x_t^2, \bar{u}_t - \delta u_t^2) \\ \vdots \\ h(\bar{x}_t + \delta x_t^{n_s}, \bar{u}_t + \delta u_t^{n_s}) - h(\bar{x}_t - \delta x_t^{n_s}, \bar{u}_t - \delta u_t^{n_s}) \end{bmatrix}$$

where 'n_s' be the number of samples for each of the random variables, δx_t and δu_t. Denote the random samples as $\delta X_t =$

$\begin{bmatrix}\delta x_t^1 & \delta x_t^2 & \dots & \delta x_t^{n_s}\end{bmatrix}$, $\delta U_t = \begin{bmatrix}\delta u_t^1 & \delta u_t^2 & \dots & \delta u_t^{n_s}\end{bmatrix}$ and $\delta Y_t = \begin{bmatrix}\delta X_t & \delta U_t\end{bmatrix}$.

We are free to choose the distribution of δx_t and δu_t. We assume both are i.i.d. Gaussian distributed random variables with zero mean and a standard deviation of σ. This ensures that $\delta Y_t \delta Y_t^T$ is invertible.

Let us consider the terms in the matrix $\delta Y_t \delta Y_t^T = \begin{bmatrix}\delta X_t \delta {X_t}^T & \delta X_t \delta {U_t}^T \\ \delta U_t \delta {X_t}^T & \delta U_t \delta {U_t}^T\end{bmatrix}$. $\delta X_t \delta {X_t}^T = \sum_{i=1}^{n_s} \delta {x_t}^i \delta {x_t}^{iT}$. Similarly, $\delta U_t \delta {U_t}^T = \sum_{i=1}^{n_s} \delta {u_t}^i \delta {u_t}^{iT}$, $\delta U_t \delta {X_t}^T = \sum_{i=1}^{n_s} \delta {u_t}^i \delta {x_t}^{iT}$ and $\delta X_t \delta {U_t}^T = \sum_{i=1}^{n_s} \delta {x_t}^i \delta {u_t}^{iT}$. From the definition of sample variance, for a large enough n_s, we can write the above matrix as

$$\begin{aligned}\delta Y_t \delta Y_t^T &= \begin{bmatrix}\sum_{i=1}^{n_s} \delta {x_t}^i \delta {x_t}^{iT} & \sum_{i=1}^{n_s} \delta {x_t}^i \delta {u_t}^{iT} \\ \sum_{i=1}^{n_s} \delta {u_t}^i \delta {x_t}^{iT} & \sum_{i=1}^{n_s} \delta {u_t}^i \delta {u_t}^{iT}\end{bmatrix} \\ &\approx \begin{bmatrix}\sigma^2(n_s-1)\mathrm{I}_{n_x} & 0_{n_x \times n_u} \\ 0_{n_u \times n_x} & \sigma^2(n_s-1)\mathrm{I}_{n_u}\end{bmatrix} \\ &= \sigma^2(n_s-1)\mathrm{I}_{(n_x+n_u)\times(n_x+n_u)}\end{aligned}$$

Typically for $n_s \sim O(n_x + n_u)$, the above approximation holds good. The reason is as follows. Note that the above least squares procedure converges when the matrix $\delta Y_t \delta Y_t^T$ converges to the identity matrix. This is entirely equivalent to estimation of the covariance of the random vector $\delta Y_t = [\delta x_t \ \delta u_t]$ where δx_t, and δu_t are Gaussian i.i.d. samples. Thus, it follows that the number of samples is $O(n_x + n_u)$, given $n_x + n_u$ is large enough (see [32]).
This has important ramifications since the overwhelming bulk of the computations in the D2C iLQR implementation consists of the estimation of these system dynamics. Moreover, these calculations are highly parallelizable.

Henceforth, we will refer to this method as 'Linear Least Squares by Central Difference (LLS-CD)'.

B. Closed Loop Design

The iLQR design in the open-loop part also furnishes a linear feedback law, however, this is not the linear feedback corresponding to the optimal feedback law. In order to accomplish this, we need to use the feedback gain equations (3). This can be done in a data based fashion analogous to the LLS-CD procedure above [33], but in practice, the iLQR feedback gain offers very comparable performance to the optimal feedback gain. The entire algorithm is summarized together in Algorithm 1.

Algorithm 1: Decoupled Data-based Control (D2C) Algorithm

⇒ ***Open-loop trajectory optimization***
Initialization: Set state $x = x_0$, initial guess $u^0_{0:N-1}$, line search parameter $\alpha = 0.3$, regularization $\mu = 10^{-6}$, iteration counter $k = 0$, convergence coefficient $\epsilon = 0.001$.
while $cost_k/cost_{k-1} < 1 + \epsilon$ **do**
  `/* backward pass */`
  $\{k^k_{0:N-1}, K^k_{0:N-1}\}$ = backward_pass()
  `/* forward pass */`
  while *cost descent not acceptable* **do**
    Reduce α
    $u^{k+1}_{0:N-1}$, $cost_k$ = forward_pass($u^k_{0:N-1}, \{k^k_{0:N-1}, K^k_{0:N-1}\}$)
  end while
  $k \leftarrow k + 1$
end while
$\bar{u}_{0:N-1} \leftarrow u^{k+1}_{0:N-1}$
⇒ ***The closed-loop feedback design***
1. $A_t, B_t \leftarrow LLS - CD(\bar{u}_{0:N-1}, \bar{x}_{0:N-1})$
2. Calculate feedback gain $K_{0:N-1}$ from equ.5.
3. Full closed-loop control policy:
$u_t = \bar{u}_t + K_t \delta x_t$,
where δx_t is the state deviation from the nominal trajectory.

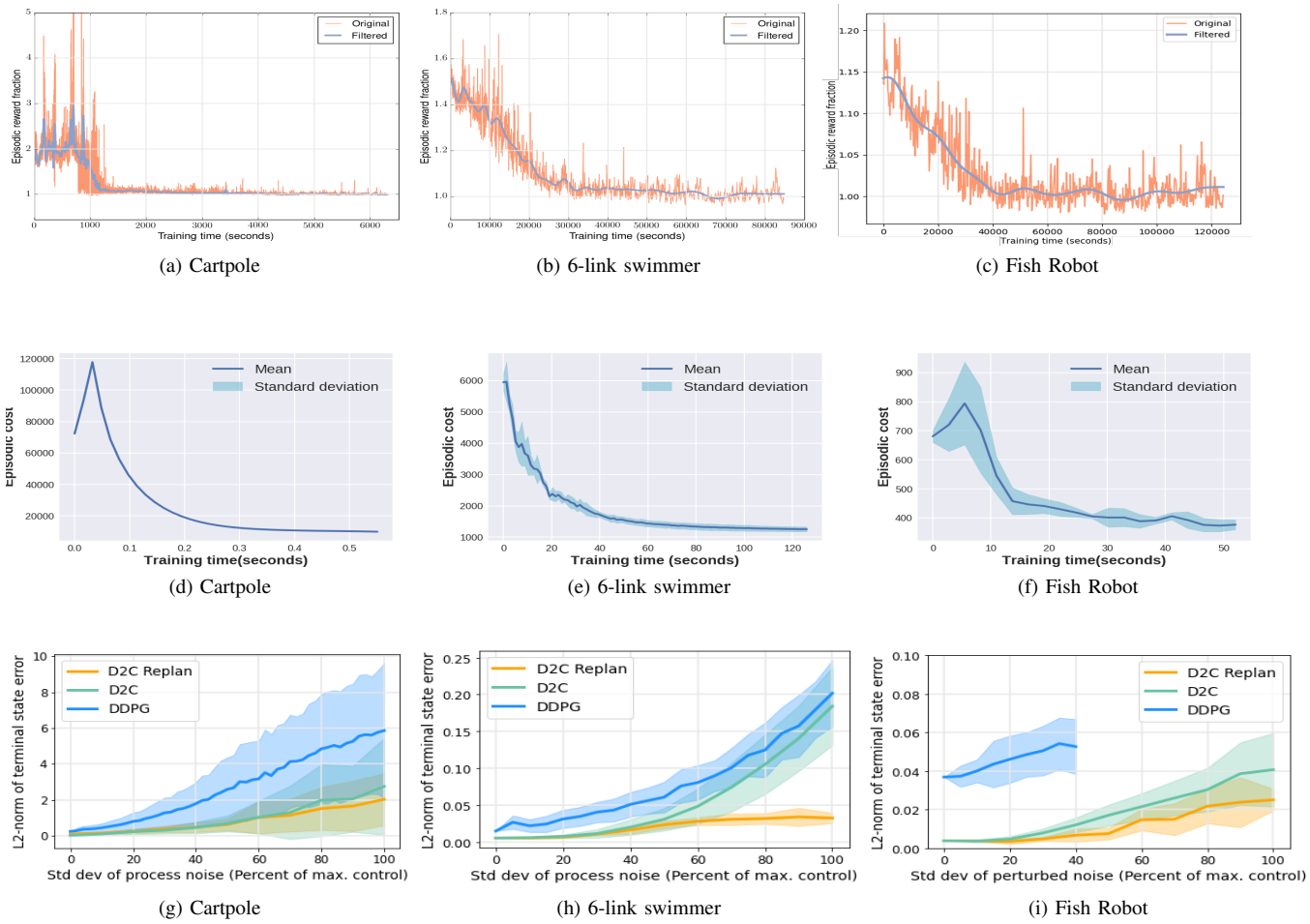
V. Empirical Results

This section reports the result of training and performance of D2C on several benchmark examples and its comparison to DDPG [22]. The physical models of the system are deployed in the simulation platform 'MuJoCo-2.0' [5] as a surrogate to their analytical models. The models are imported from the OpenAI gym [2] and Deepmind's control suite [29]. In addition, to further illustrate scalability, we test the D2C algorithm on a Material Microstructure Control problem (state dimension of 400) which is governed by a Partial Differential Equation (PDE) called the Allen-Cahn Equation. Please see the supplementary document for more details about the results as well as more experiments. All simulations are done on a machine with the following specifications: 4X Intel Xeon CPU@2.4GHz, with a 16 GB RAM, with no multi-threading.

We test the algorithms on four fronts: 1) the *training efficiency*, where we study the speed of training, 2) the *reliability of the training* studied using the variance of the resulting answers, 3) the *robustness of the learned controllers* to differing levels of noise, and hence, a test of the "globality" of the feedback, and 4) the effect of *learning in stochastic systems*.

A. Training Efficiency

We measure training efficiency by comparing the times taken for the episodic cost (or reward) to converge during training. Plots in Figure. 3 show the training process with both methods on the systems considered. Table I delineates the times taken for training respectively. The total time comparison in Table I shows that D2C learns the optimal policy orders of magnitude faster than DDPG. The primary reason for this disparity is the feedback parametrization of the two methods: the DDPG deep neural nets are complex parametrizations that are difficult to search over, when compared to the highly compact open-loop + linear feedback parametrization of D2C, i.e. the number of parameters optimized during D2C training is the number of actuators times the number of timesteps while the DDPG parameter



(a) Cartpole (b) 6-link swimmer (c) Fish Robot

(d) Cartpole (e) 6-link swimmer (f) Fish Robot

(g) Cartpole (h) 6-link swimmer (i) Fish Robot

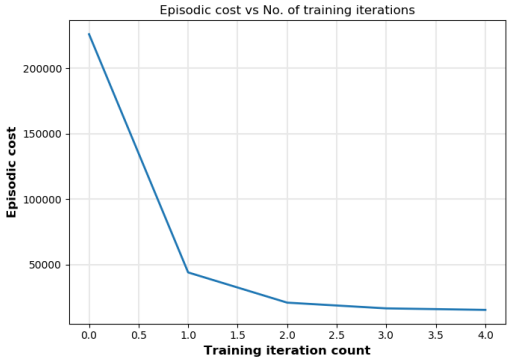
Fig. 3: Top row: Convergence of Episodic cost in DDPG. Middle row: Convergence of Episodic cost D2C. Bottom row: L2-norm of terminal state error during testing in D2C vs DDPG. The solid line in the plots indicates the mean and the shade indicates the standard deviation of the corresponding metric.

size equals the size of the neural networks, which is much larger. Due to the much larger network size, the computation done per rollout is much higher for DDPG. From Figure. 4, on the material microstructure problem (a 400 dimensional state and 100 dimensional control), we observe that D2C converges very quickly, even for a very high dimensional system ($d = 400$), whereas DDPG failed to converge to the correct goal state.

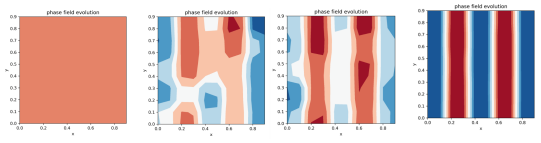
We also note the benefit of iLQR here: due to its quadratic convergence properties, the convergence is very fast, when allied with the randomized LLS-CD procedure for Jacobian estimation. We refer the reader to the Arxiv document [21] to see why we can expect it to converge to the 'global' optimum in a quadratic fashion even though the open-loop problem is non-convex. Under large noise levels, the local feedback policy may not give a good closed-loop performance, thus we introduce the replanning procedure which resolves the open-loop design from the current state of the system and wraps another local feedback policy along the new optimal trajectory. During the replanning, we take the current nominal policy as the initial guess. With this warm start, the time and iteration taken in each replanning step are less than solving the open-loop optimization with zero initial guess in D2C. Under 100% U_{max} noise, the fish needs 25 seconds and 13 iterations, the 6-link swimmer needs 90 seconds and 51 iterations in average for each replanning. As the cart-pole fails under high noise levels, it is tested with 40% U_{max} noise and needs 12 iterations and 0.5 seconds in average. Thus, by replanning, the closed-loop performance can be improved with affordable training time increase. Finally, we note that the estimation of the feedback gain takes a very small fraction of the training time when compared to the open-loop, even though it is a much bigger parameter: this is a by-product of the decoupling result.

B. Closed-loop Performance

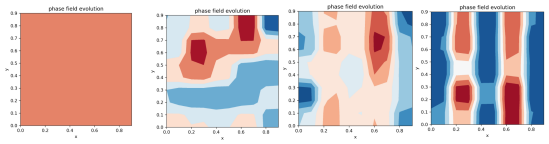
It may be expected that DDPG provides a global feedback law while D2C, by design, only a local one, and thus, the performance of DDPG might be better globally. To test this hypothesis, we apply noise to the system via the ϵ parameter,



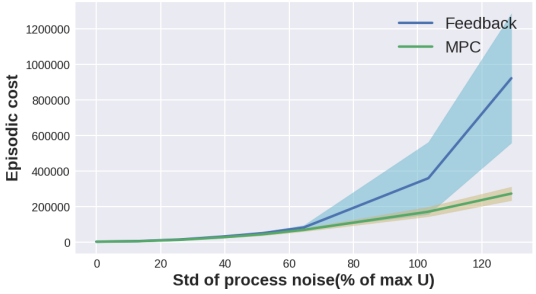
(a) Material Microstructure



(b) Initial (c) t=0.50s (d) t=1.00s (e) t=1.25s



(f) Initial (g) t=0.50s (h) t=1.00s (i) t=1.25s



(j) Closed-loop Performance

Fig. 4: Top: Episodic cost vs. training iteration number in D2C for the Material Microstructure. Middle: Closed-loop trajectories showing the temporal evolution of the spatial microstructure from the initial configuration on the left to the desired configuration on the extreme right. Figs. (b)-(e) No input noise, and (f)-(i) Gaussian input noise at std 50% U_{MAX}. Bottom: Closed-loop performance comparison between D2C with LQR feedback and D2C with replanning.

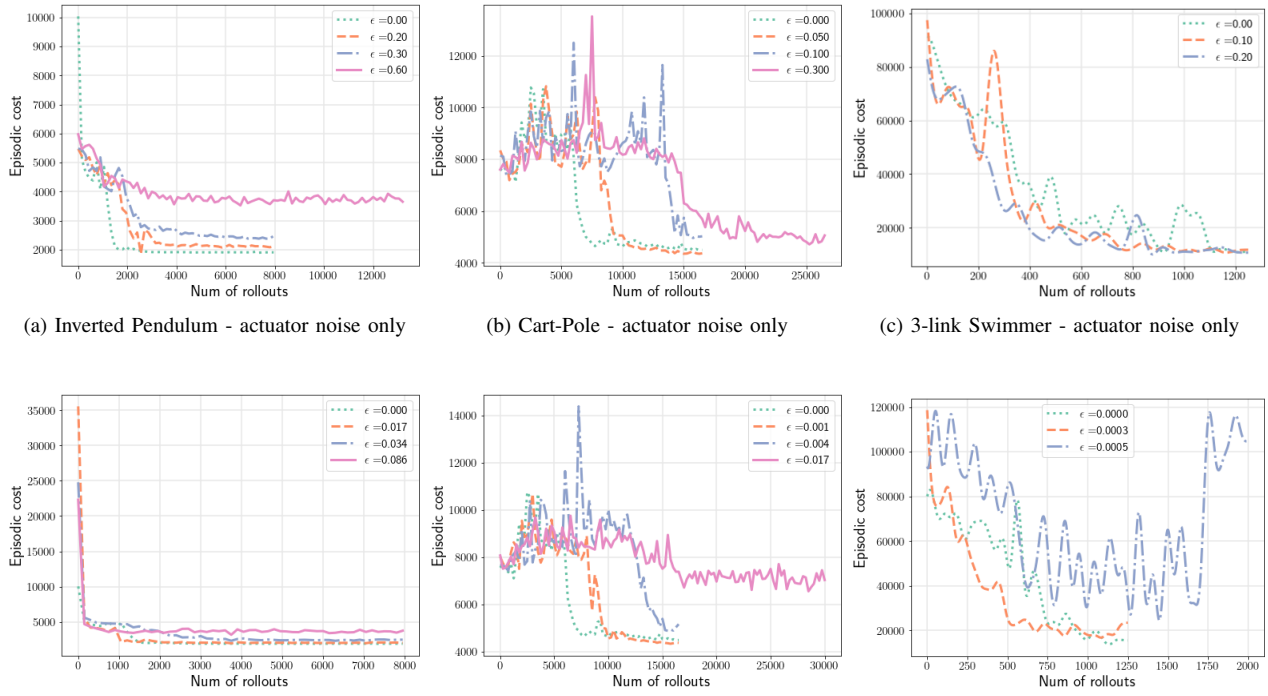
and find the average performance of the two methods at each noise level. This has the effect of perturbing the state from its nominal path, and thus, can be used to test the efficacy of the controllers far from a nominal path, i.e., their global behavior. It can be seen from Figure. 3 that the performance of D2C is actually better than DDPG at all noise levels. This, in turn implies that albeit DDPG is theoretically global in nature, in practice, it is reliable only locally, and moreover, its performance is inferior to the local D2C approach. We also report the effect of replanning on the D2C scheme, and it can be seen from these plots that the performance is far better than both D2C and DDPG, thereby regaining globally optimal behavior. We also note that the performance of D2C is similar in the high dimensional material microstructure control problem while DDPG fails to converge in this problem.

C. Reliability of Training

For any algorithm that has a training step, it is important that the training result is stable and reproducible, and thus reliable. However, reproducibility is a major challenge that the field of reinforcement learning (RL) is yet to overcome, a manifestation of the extremely high variance of RL training results. Thus, we test the training variance of D2C by conducting multiple training sessions with the same hyperparameters but different random seeds. The middle row of Figure. 3 shows the mean and the standard deviation of the episodic cost data during 16 repeated D2C training runs each. For the cart-pole model, the results of all the training experiments are almost the same. Even for more complex models like the 6-link swimmer and the fish, the training is stable and the variance is small. Further investigation into the training results shows that given the set of hyperparameters, D2C always results in the same policy (with a very small variance) unlike the DDPG results which have high variance even after convergence, which was reported in [10]. We show this in Fig. 6, where the final distance to target of the nominal trajectories (i.e., nominal control sequence of D2C and DDPG) generated from 4 different instances of converged training of D2C and DDPG with identical hyper-parameters. It can be noted that the D2C results almost overlap with each other with very small variance while the DDPG results have a wider spread. The high variance of training results makes it questionable whether DDPG indeed converges to an optimal solution or the seeming convergence is the result of shrinking exploration noise as the training progresses. On the other hand, D2C can always guarantee the same solution from a converged training. The advantage of a local approach like D2C in training stability and reproducibility makes it far more reliable for solving data-based optimal control problems.

D. Learning on Stochastic Systems

A noteworthy facet of the D2C design is that it is agnostic to the uncertainty, encapsulated by ϵ, and the near-optimality stems from the local optimality (identical nominal control and linear feedback gain) of the deterministic feedback law when applied to the stochastic system. One may then question the fact that the design is not for the true stochastic system, and thus, one may expect RL techniques to perform better since they are applicable to the stochastic system. However, in practice, most RL algorithms only consider the deterministic system, in the sense that the only noise in the training simulations is the exploration noise in the control, and not from a persistent process noise. We now show the effect of adding a persistent process noise with a small to moderate



(a) Inverted Pendulum - actuator noise only (b) Cart-Pole - actuator noise only (c) 3-link Swimmer - actuator noise only

(d) Inverted Pendulum - state and actuator noise (e) Cart-Pole - state and actuator noise (f) 3-link Swimmer - state and actuator noise

Fig. 5: Episodic cost vs number of rollouts taken during training with process noise for DDPG.

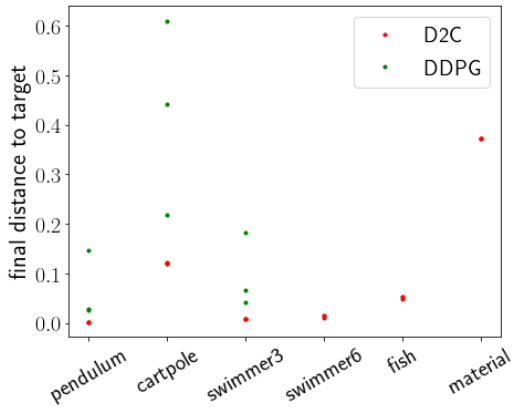


Fig. 6: Training variance comparison D2C vs. DDPG. The DDPG results for swimmer6, fish and material examples are lacking because the training time taken is too long.

value of ϵ to the training of DDPG, in the control as well as the state.

We trained the DDPG policy on the pendulum, cart-pole and 3-link swimmer examples. To simulate the stochastic environment, Gaussian i.i.d. random noise is added to all the input channels as process noise. As usual, the noise level ϵ is the noise standard deviation divided by the maximum control value of the open-loop optimal control sequence. Figure. 5 shows the DDPG training curve under different levels of process noise. As the process noise increases, the episodic cost converges slower and to a worse policy. When the process noise is larger than a threshold, the algorithm may altogether fail to converge for a given time budget. The problem is greatly exacerbated in the presence of state noise as seen from Figure. 5 that results in non-convergence or bad policies in the different examples for even small levels of noise. Hence, although theoretically, RL algorithms such as DDPG can train on the stochastic system, in practice, the process noise level ϵ must be limited to a small value for training convergence and/or good policies. Thus, this begs the question as to whether we should train on the stochastic system rather than appeal to the decoupling result that the deterministic policy is locally identical to the optimal stochastic policy, and thus train on the deterministic system. A theoretical exploration of this topic, in particular, the variance inherent in RL, is the subject of the companion manuscript [3].

TABLE I: Comparison of the training outcomes of D2C with DDPG.

System	Training time (in sec.)		Training variance	
	D2C	DDPG	D2C	DDPG
Inverted Pendulum	0.33	2261.15	6.7×10^{-5}	0.08
Cart pole	0.55	6306.7	0.0004	0.16
3-link Swimmer	186.2	38833.64	0.0007	0.05
6-link Swimmer	127.2	88160	0.0023	*
Fish	54.8	124367.6	0.0016	*

* DDPG training variance is not tested for 6-link swimmer and fish because the training time taken is too long.

VI. Conclusion

The D2C policy is not global, i.e., it does not claim to be valid over the entire state space, however, seemingly global deep RL methods do not offer better performance as can be seen from our experiments. Further, owing to the fast and reliable open-loop solver, D2C could offer a real time solution even for high dimensional problems when allied with high performance computing. In such cases, one could replan whenever necessary, and this replanning procedure will make the D2C approach global in scope, as we have shown in this paper albeit not in real time. There might be a sentiment that the comparison with DDPG is unfair due to the wide chasm in the training times, however, the primary point of our paper is to show theoretically, as well as empirically, that the local parametrization and search procedure, is a highly efficient and reliable (almost zero variance) alternative that is still superior in terms of closed-loop performance when compared to typical global RL algorithms like DDPG. Thus, for data-based optimal control problems that need efficient training, reliable near optimal solution and robust closed-loop performance, such local RL techniques, coupled with replanning, should be the preferred method over typical global RL methods.

References

[1] D. P. Bertsekas. *Dynamic Programming and Optimal Control, Two Volume Set*. Athena Scientific, 2nd edition, 1995.

[2] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[3] S. Chakravorty, R. Wang, and M. N. G. Mohamed. On the convergence of reinforcement learning. *arXiv: 2011.10829*, 2020.

[4] M. Deisenroth and C. Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *International Conference on Machine Learning (ICML)*, 2011.

[5] T. Emanuel, E. Tom, and Y. Tassa. Mujoco: A physics engine for model-based control. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012.

[6] S. Gu, T. Lillicrap, Z. Ghahramani, R. E. Turner, and S. Levine. Q-prop: Sample-efficient policy gradient with an off-policy critic. *arXiv preprint arXiv:1611.02247*, 2016.

[7] W. Heemels, K. Johansson, and P. Tabuada. An introduction to event triggered and self triggered control. In *Proc. IEEE Int. CDC*, 2012.

[8] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger. Deep reinforcement learning that matters. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[9] P. A. Ioannou and J. Sun. *Robust adaptive control*. Courier Corporation, 2012.

[10] R. Islam, P. Henderson, M. Gomrokchi, and D. Precup. Reproducibility of benchmarked deep reinforcement learning tasks for continuous control. *Reproducibility in Machine Learning Workshop, ICML'17*, 2017.

[11] D. Jacobsen and D. Mayne. *Differential Dynamic Programming*. Elsevier, 1970.

[12] P. R. Kumar and P. Varaiya. *Stochastic systems: Estimation, identification, and adaptive control*, volume 75. SIAM, 2015.

[13] V. Kumar, E. Todorov, and S. Levine. Optimal Control with Learned Local Models: Application to Dexterous Manipulation. In *International Conference for Robotics and Automation (ICRA)*, 2016.

[14] S. Levine, C. Finn, T. Darrell, and P. Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

[15] S. Levine and K. Vladlen. Learning Complex Neural Network Policies with Trajectory Optimization. In *Proceedings of the International Conference on Machine Learning*, 2014.

[16] H. Li, Y. She, W. Yan, and K. Johansson. Periodic event-triggered distributed receding horizon control of dynamically decoupled linear systems. In *Proc. IFAC World Congress*, 2014.

[17] W. Li and E. Todorov. Iterative linearization methods for approximately optimal control and estimation of non-linear stochastic system. *International Journal of Control*, 80(9):1439–1453, 2007.

[18] T. Lillicrap et al. Continuous control with deep reinforcement learning. In *Proc. ICLR*, 2016.

[19] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[20] D. Mitrovic, S. Klanke, and S. Vijayakumar. *Adaptive Optimal Feedback Control with Learned Internal Dynamics Models, in From Motor Learning to Interaction Learning in Robots. Studies in Computational Intelligence, vol 264*. Springer, Berlin, 2010.

[21] M. N. G. Mohamed, S. Chakravorty, and R. Wang. Optimality and Tractability in Stochastic Nonlinear Control. *arXiv: 2004.01041*, 2020.

[22] M. Plappert. keras-rl. https://github.com/keras-rl/keras-rl, 2016.

[23] W. B. Powell. *Approximate Dynamic Programming: Solving the curses of dimensionality*. John Wiley & Sons, 2007.

[24] J. Schulman, S. Levine, P. Moritz, M. Jordan, and P. Abbeel. Trust region policy optimization. *arXiv:1502.05477*, 2017.

[25] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv:1707.06347*, 2017.

[26] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

[27] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[28] Y. Tassa, T. Erez, and E. Todorov. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4906–4913. IEEE, 2012.

[29] Y. Tassa et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690v1*, 2018.

[30] E. Theoddorou, Y. Tassa, and E. Todorov. Stochastic Differential Dynamic Programming. In *Proceedings of American Control Conference*, 2010.

[31] E. Todorov and W. Li. A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems. In *Proceedings of American Control Conference*, pages 300 – 306, 2005.

[32] R. Versyhnin. *High Dimensional Probability: An Introduction with Application to Data Science*. Cambridge University Press, Cambridge, UK, 2018.

[33] R. Wang, K. S. Parunandi, A. Sharma, S. Chakravorty, and D. Kalathil. On the search for feedback in reinforcement learning. *arXiv: 2002.09478*, 2020.

[34] R. Wang, K. S. Parunandi, D. Yu, D. M. Kalathil, and S. Chakravorty. Decoupled data based approach for learning to control nonlinear dynamical systems. *arXiv, also IEEE Transactions on Automatic Control, accepted*, abs/1904.08361, 2019.

[35] D. Yu, M. Rafieisakhaei, and S. Chakravorty. Stochastic Feedback Control of Systems with Unknown Nonlinear Dynamics. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 4309–4314, 2017.

[36] W. Yuhuai, M. Elman, L. Shun, G. Roger, and B. Jimmy. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. *arXiv:1708.05144*, 2017.