Data-Driven Near-Optimal Control of Nonlinear Systems Over Finite Horizon

Vasanth Reddy Baddam VASANTH2608@VT.EDU
Department of Computer Science, Virginia Tech, Arlington, VA, USA.

Hoda Eldardiry HDARDIRY@VT.EDU
Department of Computer Science, Virginia Tech, Blacksburg, VA, USA.

Almuatazbellah Boker BOKER@VT.EDU
Bradley Department of Electrical and Computer Engineering, Virginia Tech, Arlington, VA, USA.

Editors: N. Ozay, L. Balzano, D. Panagou, A. Abate

Abstract

We employ reinforcement learning to address the problem of two-point boundary optimal control for nonlinear systems over a finite time horizon with unknown model dynamics. By leveraging techniques from singular perturbation theory, we decompose the finite-horizon control problem into two sub-problems, each defined over an infinite horizon. This decomposition eliminates the need to solve the time-varying Hamilton-Jacobi-Bellman (HJB) equation, significantly simplifying the process. Using a policy iteration method enabled by this decomposition, we learn the controller gains for each of the two sub-problems. The overall control strategy is then constructed by combining the solutions of these sub-problems. We demonstrate that the performance of the proposed closed-loop system asymptotically approaches the model-based optimal performance as the time horizon becomes large. Finally, we validate our approach through simulation scenarios, which provide strong support for the claims made in this paper.

Keywords: optimal control; nonlinear systems; singular perturbation

1. Introduction

Over the past decade, extensive research has focused on the control and analysis of nonlinear systems Khalil (2002); Vidyasagar (2002), aiming to develop effective control strategies, often grounded in the Hamilton-Jacobi-Bellman (HJB) equation. A key challenge in this domain is controlling nonlinear systems within a finite time horizon, which is considerably more complex than controlling them over an infinite horizon. This complexity arises from the need to solve a time-varying HJB equation in the presence of dynamic system uncertainties. Addressing these challenges requires a range of approaches tailored to the specific problem of finite-horizon control.

One classical approach is Pontryagin's Maximum Principle (PMP), which transforms the problem into a boundary value problem (BVP) that is challenging to solve due to its high dimensionality, often $2n$ dimensions Betts (1998). The indirect nature of PMP requires sophisticated analysis to define boundary conditions, further complicating the solution process. Another method is the discretization of the infinite-dimensional optimal control problem into a finite-dimensional one, which can be solved using nonlinear programming (NLP) techniques. Tools like GPOPS-II leverage this approach Patterson and Rao (2014), but the discretization demands high computational effort, and the resulting NLP problems often fail to reach global optima, especially for nonlinear systems Rao (2009). Moreover, both PMP and discretization methods assume perfect knowledge of system dynamics, a significant limitation in real-world applications where model uncertainties are prevalent.

To address this, model-free methods such as adaptive dynamic programming and actor-critic learning have gained traction, particularly in infinite-horizon settings Jiang and Jiang (2015); Lv et al. (2016); Vamvoudakis (2017); Vamvoudakis and Lewis (2010); Luo et al. (2016); Lin et al. (2017). While these approaches bypass the need for accurate system models, they remain primarily applicable to problems over infinite time horizons. Conversely, few learning algorithms exist for optimal control in finite-horizon settings. Recent efforts have explored model-based Kim et al. (2018) and model-free Zhao et al. (2015); Zhao and Gan (2020); Chen et al. (2022) approaches that learn the time-varying weights of basis function vectors. However, designing these time-varying basis functions is inherently complex. In contrast, our approach avoids this complexity by transforming the HJB equation from a time-varying to a time-invariant form, employing a singular perturbation method. This transformation simplifies the problem significantly, making it more tractable for analysis and solution.

In this paper, we address the challenge of optimal control for uncertain nonlinear systems over finite time horizons. Our approach builds on our previous work Reddy et al. (2022), where we applied singular perturbation theory to solve optimal control problems for *linear time-varying systems*. In that earlier work, we demonstrated how the time-varying Algebraic Riccati Equation (ARE) could be decomposed into two linear time-invariant (LTI) AREs, enabling the use of reinforcement learning to approximate optimal control policies for such systems. While effective for linear systems, this approach left several open questions, particularly regarding its extension to *nonlinear* systems and time-varying Hamilton-Jacobi-Bellman (HJB) equations. In the current work, we focus on addressing these challenges in the context of nonlinear systems. A key insight of our approach is the recognition of a two-time-scale phenomenon that naturally arises in nonlinear systems, even when the original dynamics are not explicitly perturbed. This phenomenon, as described by Wilde and Kokotovic (1972), shows that the closed-loop system's dynamics evolve at a faster rate than the time horizon itself, with this effect becoming more pronounced as the time horizon increases. By leveraging this two-time-scale behavior, we approximate the finite-horizon optimal control problem by decomposing it into two infinite-horizon sub-problems: one focused on stabilizing the system forward in time, and the other stabilizing it backward in time. These sub-problems are solved independently and then combined to achieve near-optimal control, particularly over long time intervals. This extension from linear time-varying systems to nonlinear systems represents a significant advancement, as we now tackle the more complex time-varying HJB equations in uncertain environments, providing a more general and robust framework for solving finite-horizon optimal control problems in practical nonlinear systems.

To address the uncertainty in the system dynamics, we adopt a policy iteration framework using continuous-time Q-learning, as recently developed by Chen and Herrmann (2019). This learning-based approach allows us to estimate control gains for the two sub-problems while accommodating model uncertainties, thus avoiding the need to solve complex time-varying partial differential equations. The combination of these insights forms the primary contribution of our work: we provide a solution to the two-point boundary optimal control problem for a broad class of uncertain nonlinear systems using policy iteration methods. Crucially, our approach circumvents the computationally demanding task of solving time-varying HJB equations, offering a more efficient learning algorithm that scales to real-world scenarios. Our method achieves convergence to solutions comparable to those obtained through traditional numerical solvers, particularly as the control time interval increases. This provides a significant advantage in practical applications where solving for finite-horizon control in uncertain systems is critical.

The remainder of this paper is organized as follows. Section 2 presents the system setup and problem formulation. In Section 3, we describe the two-time-scale reduction of the original problem. Section 4 details the policy iteration learning procedure used to estimate the control gains. Section 5 provides a simulation example, and Section 6 concludes with final remarks.

2. Problem Formulation

We consider a continuous-time nonlinear affine system represented by the dynamics:

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t), \quad x(0) = x_0,\ x(T) = x_T, \tag{1}$$

where $x \in \mathbb{R}^n$ denotes the system states and $u(x) \in \mathbb{R}^m$ is the control input. The goal is to design a control law $u(x)$ that drives the system from the initial state $x(0) = x_0$ to the terminal state $x(T) = x_T$ over a finite time horizon T, while minimizing a predefined cost function.

Assumption 1 *The nonlinear functions $f(x)$ and $g(x)$ are smooth and globally Lipschitz continuous. Furthermore, the function $f(x)$ is unknown, representing uncertainty in the system dynamics.*

Assumption 2 *The system (1) is controllable, meaning it is possible to drive the system from any initial state to the origin and from the origin to any prescribed final state within a finite time interval.*

The objective is to design a control law $u(x)$ that not only transitions the system from x_0 to x_T over time T, but also minimizes the following cost function:

$$J = \int_0^T \mathcal{R}(x(t), u(t))\, dt, \tag{2}$$

where $\mathcal{R}(x(t), u(t)) = \mathcal{S}(x(t)) + u^\top(t)Ru(t)$, $\mathcal{S}(x) \succ 0$ is a positive definite state cost matrix, and $R \succ 0$ is a positive definite control cost matrix for all $t \in [0, T]$.

To approach the optimal control problem, we define the value function, which represents the minimum cost-to-go, with initial condition $x(0)$, which is given by:

$$V^*(x(0)) = \min_{u(t)} J(x, u, 0, T), \tag{3}$$

where $V^*(x)$ is the optimal value function, and $u^*(t)$ is the corresponding optimal control law. The optimal control problem is solved by finding the solution to the Hamilton-Jacobi-Bellman (HJB) equation Athans and Falb (2013):

$$-\frac{\partial V^*}{\partial t} = \min_u \left[\mathcal{R}(x, u) + \frac{\partial V^*}{\partial x} \left(f(x) + g(x)u \right) \right]. \tag{4}$$

The HJB equation (4) is a nonlinear partial differential equation (PDE) that characterizes the optimal control policy. However, solving it analytically for general nonlinear systems is often intractable due to the curse of dimensionality and the complexity of the PDE. Thus, in the subsequent section, we employ a singular perturbation approach to simplify the HJB equation, transforming it into a more manageable form. We will then introduce a data-driven method for learning the control law without requiring a full solution to the time-varying HJB equation leading to a near optimal performance.

3. Control Design using Singular Perturbation Method

For control problems over a finite time horizon, it is well known that as the time period T increases, the system trajectories begin to exhibit two time-scale behavior Wilde and Kokotovic (1972). In such cases, most of the transient dynamics occur near the boundary points $t = 0$ and $t = T$, while the system remains in a quasi-steady state in the interior. This phenomenon of two-time scale allows us to simplify the control problem by focusing on the behavior of the system at the boundary points, significantly reducing the complexity of the problem.

We begin by setting up the singular perturbation model for the system. To this end, we introduce the scaled time τ and a small parameter ε such that

$$\tau = \frac{t}{T}, \quad \varepsilon = \frac{1}{T}. \tag{5}$$

This allows for rewriting the system dynamics (1), the cost function (2), and the HJB equation (4) as follows:

$$\varepsilon \frac{dx(\tau)}{d\tau} = f(x(\tau)) + g(x(\tau))u(\tau), \quad x(0) = x_0,\, x(1) = x_T, \tag{6}$$

$$-\varepsilon \frac{\partial V^*}{\partial \tau} = \min_{u^*(\tau)} \left[\mathcal{R}(x,u) + \frac{\partial V^*}{\partial x} \left(f(x) + g(x)u^*(\tau) \right) \right]. \tag{7}$$

Taking the limit as $\varepsilon \to 0$ in (7) simplifies the HJB equation to:

$$0 = \min_{u^*(\tau)} \left[\mathcal{R}(x,u^*) + \frac{\partial V^*}{\partial x} \left(f(x) + g(x)u^*(\tau) \right) \right]. \tag{8}$$

Following the singular perturbation approach, it can be shown that the solution to (8) can be decomposed into two boundary layer solutions: a *forward controller* $u_a(x)$ for stabilizing the system forward in time, and a *backward controller* $u_b(x)$ for stabilizing the system in reverse time Kokotović et al. (1999). Both of these solutions are relatively easier to find compared to the solution of (4). This leads to two controllers that handle the dynamics at the boundaries $t = 0$ and $t = T$, respectively. Next, we describe the design of these two controllers.

3.1. Forward Controller

In the limit as $\varepsilon \to 0$ in (7), the value function associated with the system in forward time, with initial condition, $x(0)$ is given by:

$$V_a(x(0)) = \min_{u_a(.)} \int_0^\infty \mathcal{R}(x(\tau), u_a(\tau))\, d\tau, \tag{9}$$

where $u_a(\tau)$ is the forward controller. The corresponding HJB optimality condition for the forward value function is given by (8).

Assuming the minimum exists and is unique, the optimal control law for the forward controller is given by:

$$u_a^*(\tau) = -\frac{1}{2} R^{-1} g^\top(x) \frac{\partial V_a^*}{\partial x}. \tag{10}$$

3.2. Backward Controller

For the backward controller, we consider reverse time $s = -\tau$. The corresponding value function in reverse time, and in the limit as $\varepsilon \to 0$, is given by:

$$V_b(x(T)) = -\int_{-\infty}^{T} \mathcal{R}(x(s), u_b(s))\, ds, \tag{11}$$

where $u_b(\tau)$ is the backward controller. Moreover, the HJB optimality condition for the backward value function is given by:

$$\min_{u_b(.)} \left[\mathcal{R}(x, u_b) - \frac{\partial V_b^*}{\partial x} \left(f(x) + g(x) u_b^*(s) \right) \right] = 0. \tag{12}$$

As with the forward controller, if the minimum exists and is unique, the optimal control law for the backward controller is:

$$u_b^*(s) = \frac{1}{2} R^{-1} g^\top(x) \frac{\partial V_b^*}{\partial x}. \tag{13}$$

3.3. Near-Optimal Performance

It has been shown in Anderson and Kokotovic (1987); Kokotović et al. (1999) that the combination of these two infinite-horizon subproblems provides an approximation to the original finite-horizon value function $V^*(x(0), x(T))$ for sufficiently small ε. The following theorem summarizes this result:

Theorem 1 *Let Assumptions 1-2 hold. Suppose further that $\|x(0)\| \leq 1$ and $\|x(T)\| \leq 1$. As defined in (5), $\epsilon = \frac{1}{T}$ as the small perturbation parameter inversely proportional to the time horizon. Then, there exists $\varepsilon_1 > 0$ such that for all $\varepsilon \in (0, \varepsilon_1]$,*

$$V^*(x(0), x(T)) \leq V_a^*(x(0)) - V_b^*(x(T)) + k_1(\varepsilon), \tag{14}$$

$$V_a^*(x(0)) - V_b^*(x(T)) \leq V^*(x(0), x(T)) + k_2(\varepsilon), \tag{15}$$

where $k_1(\varepsilon)$ and $k_2(\varepsilon)$ are monotonic functions of ε with $\lim_{\varepsilon \to 0} k_1(\varepsilon) = 0$ and $\lim_{\varepsilon \to 0} k_2(\varepsilon) = 0$.

Remark 2 *Together, (14) and (15) show that $V^*(x(0), x(T))$ and $V_a^*(x(0)) - V_b^*(x(T))$ can be made arbitrarily close by selecting T large enough. It will also be shown later in the simulation results that the optimal controller can be approximated by the composite addition of the forward and backward controller, that is*

$$u^*(t) = u_a^*(\tau) + u_b^*(s) + \mathcal{O}(\epsilon). \tag{16}$$

Remark 3 *In Theorem 1, it is assumed that $\|x(0)\| \leq 1$ and $\|x(T)\| \leq 1$. However, this assumption can be relaxed to any compact set. The primary effect of changing the initial and terminal sets is that the required time interval T for achieving a given closeness between $V^*(x(0), x(T))$ and $V_a^*(x(0)) - V_b^*(x(T))$ may change.*

In the following section, we present a reinforcement learning algorithm to compute the optimal value functions V_a and V_b, and thereby derive the controllers $u_a^*(\tau)$ and $u_b^*(s)$, without requiring explicit knowledge of $f(x)$.

4. Main Results

4.1. Learning-Based Design

In this section, we employ a policy iteration method, as outlined in Chen and Herrmann (2019), to learn the value functions for both the forward and backward regulators. The key idea is to iteratively update the control policies and their corresponding value functions until convergence to optimal policies is achieved. In this section, we follow a learning approach to learn the forward and terminal backward problems separately.

4.1.1. LEARNING FOR THE FORWARD REGULATOR

The core of our learning approach is Policy Iteration, which alternates between policy evaluation and policy improvement to converge to an optimal control policy. Below, we describe this procedure for the forward regulator problem.

Policy Iteration:

1. For a given control policy, $u_a^k(\tau)$, solve for the value function $V_a^k(x)$ using:

$$V_a^{k-1}(x(t-T)) = \int_{t-T}^{t} \mathcal{R}(x(\tau), u_a(\tau))\,\mathrm{d}\tau + V_a^k(x(t)), \tag{17}$$

   where the value function approximates the cumulative cost over time.

2. Update the control policy using the value function:

$$u_a^{k+1} = -\frac{1}{2} R^{-1} g^\top(x) \nabla V_a^k(x), \tag{18}$$

   where $\nabla V_a^k(x)$ represents the gradient of the value function with respect to the system state.

Value Function Approximation:

Next, we approximate the value function $V_a(x)$ using a neural network-based adaptive critic. This approximation is given by:

$$V_a = W_a^\top \phi(x) + \delta(x), \tag{19}$$

where $\phi(x) : \mathbb{R}^n \to \mathbb{R}^N$ represents the activation function vector with N neurons in the hidden layer, $W_a \in \mathbb{R}^N$ is the weight vector, and $\delta(x)$ is the approximation error.

The activation functions are chosen such that the neural network can approximate any function within a compact set Ω. Substituting (19) into (17), we obtain the temporal difference:

$$-\delta_B = \int_{t-T}^{t} \mathcal{R}(x(\tau), u_a(\tau))\,\mathrm{d}\tau + W_a^\top \Delta\phi(t), \tag{20}$$

where $\delta_B = \delta(x(T)) - \delta(x(t-T))$ represents the Bellman residual, and $\Delta\phi(t) = \phi(x(t)) - \phi(x(t-T))$.

To ensure the stability and convergence of the critic learning process, we introduce the auxiliary variables $\xi \in \mathbb{R}^{N\times N}$ and $\psi \in \mathbb{R}^N$. These variables act as low-pass filters, smoothing the updates to the critic weights and preventing instability in the weight adjustment process. Specifically, ξ accumulates the contributions of the activation function $\Delta\phi(t)$, while ψ accumulates the contributions of

the reward integral $\rho_a(x,u)$. The evolution of these auxiliary variables is governed by the following equations:

$$\dot{\xi} = -\ell\xi + \Delta\phi(t)\Delta\phi(t)^\top, \quad \xi(0) = 0, \tag{21}$$

$$\dot{\psi} = -\ell\psi + \Delta\phi(t)\rho_a(x,u), \quad \psi(0) = 0, \tag{22}$$

where $\ell > 0$ is a positive constant that ensures boundedness of the variables and stability of the weight update process Na et al. (2015) and $\rho_a(x,u)$ is defined as:

$$\rho_a(x,u) = \int_{t-T}^{t} \mathcal{R}(x(\tau), u_a(\tau))\,\mathrm{d}\tau. \tag{23}$$

These auxiliary variables accumulate information over time and are used in the critic weight update, ensuring that the learning process converges smoothly.

The weight update for the critic network is given by:

$$\hat{W}_a = -\Gamma\xi\frac{G}{\|G\|}, \tag{24}$$

where $G = \xi\hat{W}_a + \psi$, and $\Gamma > 0$ is the learning gain parameter. The resulting estimated value function is:

$$\hat{V}_a = \hat{W}_a^\top\phi(x). \tag{25}$$

Policy Improvement:

Once the adaptive critic has been updated, the control policy can be improved using the following update rule:

$$u_a = -\frac{1}{2}R^{-1}g(x)^\top\nabla\phi^\top\hat{W}_a. \tag{26}$$

4.1.2. BACKWARD REGULATOR

We follow a similar approach to design the backward regulator, with the integral term for policy evaluation modified as:

$$\rho_b(x_b,u) = -\int_{t-T}^{t} \mathcal{R}(x_b(\tau), u_b(s))\,\mathrm{d}\tau. \tag{27}$$

The control policy for the backward regulator is updated as:

$$u_b(s) = \frac{1}{2}R^{-1}g(x_b)^\top\nabla\phi^\top\hat{W}_b. \tag{28}$$

The pseudocode for the policy iteration algorithm for both regulators is presented in Algorithm 1.

4.2. Convergence Guarantees

The Persistent Excitation (PE) condition Na et al. (2015) ensures convergence of the adaptive critic by requiring the signal $\Delta\phi(x)$ to satisfy $\int_{\tau-1}^{\tau}\Delta\phi(\bar{\tau})\Delta\phi(\bar{\tau})^\top\,\mathrm{d}\tau \geq \sigma_1 I$. where $\sigma_1 > 0$. Lemma Na et al. (2015) guarantees that this condition ensures positive definiteness of the auxiliary matrix ξ, monitored online via its minimum eigenvalue.

Finally, we define the estimation error for the value function as $\tilde{V}_a = V_a^* - \hat{V}_a$. If the system state $x(\tau)$ is bounded and $x(\tau)$, $\Delta\phi(x)$ are persistently excited, we have the following lemmas and corollaries:

Algorithm 1 Policy Iteration on Two Boundary Value Problems

Result: Control policies for forward and backward regulator problems
Initialization: W_a, W_b, $\delta > 0, k = 0$, exploration noise e
while $\Delta W_a \geq \delta$ **do**
  Get the state x from the system (1) using the control policy (10), $u_a^k = u_a^k + e$
  Evaluate the value function $V_a^k = (\hat{W}_a^k)^\top \phi(x)$
  Policy evaluation: $u_a^{k+1} = -\frac{1}{2} R^{-1} g^\top(x) \nabla V_a^k(x)$
  Update weights as in (24)
  Compute weight change: $\Delta W_a = W_a^k - W_a^{k+1}$; $k = k + 1$;
end
$u_a = -\frac{1}{2} R^{-1} g^\top(x) \nabla \phi^\top W_a^{k+1}$
while $\Delta W_b \geq \delta$ **do**
  Get the state x from the system (1) using the control policy (13), $u_b^k = u_b^k + e$
  Evaluate the value function $V_b^k = (\hat{W}_b^k)^\top \phi(x)$
  Policy evaluation: $u_b^{k+1} = \frac{1}{2} R^{-1} g^\top(x) \nabla V_b^k(x)$
  Update weights as in (24)
  Compute weight change: $\Delta W_b = W_b^k - W_b^{k+1}$; $k = k + 1$;
end
$u_b = \frac{1}{2} R^{-1} g^\top(x) \nabla \phi^\top W_b^{k+1}$

Lemma 4 *Na et al. (2015) Let $\hat{V}_a$ and $\hat{V}_b$ denote the estimated value functions obtained using neural network-based approximations for the forward and backward subproblems (9), (10), (11), (13), respectively. Let V_a^* and V_b^* represent the corresponding optimal value functions. Then, for a sufficiently small error parameter δ, the estimates $\hat{V}_a$ and $\hat{V}_b$ approximate V_a^* and V_b^* within an error bound such that:*

$$\|\hat{V}_a - V_a^*\| \leq \delta \quad \textit{and} \quad \|\hat{V}_b - V_b^*\| \leq \delta,$$

where δ represents the approximation error dependent on the neural network's capacity and training accuracy.

This lemma establishes that both the forward and backward neural network-based value function estimates, $\hat{V}_a$ and $\hat{V}_b$, converge closely to their respective optimal value functions, V_a^* and V_b^*, within an error bound δ.

Corollary 5 *The finite-horizon optimal value function V^*, which is the solution to the original control problem, can be approximated by the difference between the optimal forward and backward value functions V_a^* and V_b^*, with an additional error dependent on the neural network approximation error δ and the singular perturbation parameter ε:*

$$V^* = V_a^* - V_b^* + \mathcal{O}(\varepsilon).$$

By substituting the neural network-based estimates $\hat{V}_a$ and $\hat{V}_b$ for V_a^* and V_b^*, we refine the approximation as:

$$V^* = \hat{V}_a - \hat{V}_b + \mathcal{O}(\delta) + \mathcal{O}(\varepsilon).$$

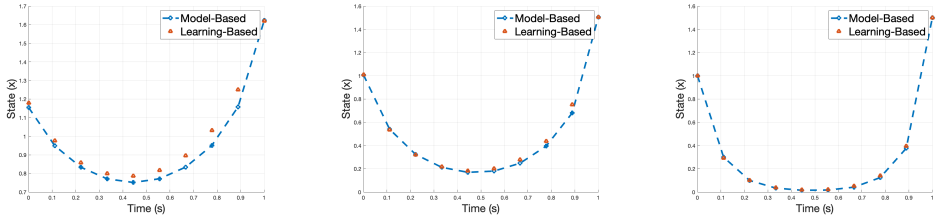

Figure 1: The state space trajectory for the state x for the system (29) is depicted using plots for various values of $\varepsilon = 0.5, 0.2$, and 0.1 accordingly. [**Normalized time axis**]

This result indicates that the finite-horizon value function V^* can be effectively approximated using the estimated value functions $\hat{V}_a$ and $\hat{V}_b$, with accuracy determined by both the neural network approximation and the time-scaling perturbation.

Remark 6 *Building upon the above corollary, the optimal control input u^* for the finite-horizon problem can also be approximated using the neural network-based estimates $\hat{V}_a$ and $\hat{V}_b$ as follows:*

$$u^* = \hat{u}_a + \hat{u}_b + \mathcal{O}(\delta) + \mathcal{O}(\varepsilon),$$

where $\hat{u}_a$ and $\hat{u}_b$ are the control inputs derived from the estimated value functions $\hat{V}_a$ and $\hat{V}_b$, respectively.

This corollary demonstrates that the finite-horizon optimal control input u^* can be approximated by combining the neural network-based forward and backward control inputs, with an error dependent on both the neural network approximation accuracy δ and the perturbation parameter ε.

5. Examples

This section demonstrates the effectiveness of the proposed approach through two examples.

5.1. Nonlinear System

Consider the nonlinear scalar system:

$$\dot{x} = x^3 + u. \tag{29}$$

The objective is to steer x from $x_0 = 1$ to $x_T = 1.5$, minimizing (2) with $\mathcal{S}(x) = 1$ and $R = 1$. Exploration noise $e = 2\sin t$ is added during training. The activation function is chosen as, $\phi(x) = \begin{bmatrix} x^2 \\ x^4 \end{bmatrix}$, with weights initialized randomly between -1 and 1. Using Algorithm 1, the weights converge to:

$$W_+ = \begin{bmatrix} 0.9740 \\ 0.6933 \end{bmatrix}, \quad W_- = \begin{bmatrix} -0.9385 \\ -0.6360 \end{bmatrix}.$$

For comparison, the singular perturbation method (Section 3) is used, assuming the system model is known. Solving the associated HJB equations yields the analytical value functions:

$$V_+(x) = \frac{1}{2}x^2\left(\sqrt{1+x^4} + x^2\right) + \frac{1}{2}\ln\left(\sqrt{1+x^4} + x^2\right), \tag{30}$$

$$V_-(x) = -\frac{1}{2}x^2\left(\sqrt{1+x^4} + x^2\right) + \frac{1}{2}\ln\left(\sqrt{1+x^4} + x^2\right). \tag{31}$$

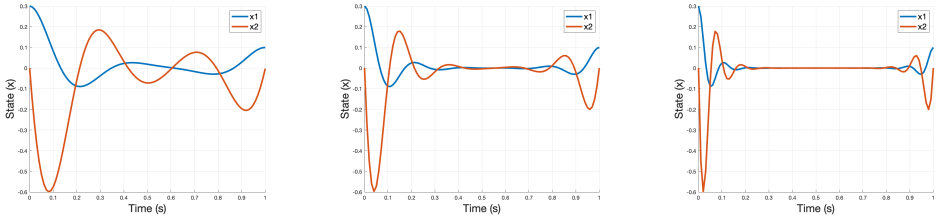

Figure 2: The state space trajectory of the robotic manipulator system (32) is depicted using plots for various values of $T = 5$, sec, 10 sec, and 20 sec, from left to right. [**Normalized time axis**]

The forward and backward regulators are derived by substituting these functions into (10) and (13), respectively.

As shown in Fig. 1, the learning-based and model-based controllers exhibit similar performance, validating the proposed approach for nonlinear systems.

5.2. Robotic Manipulator

Consider a robotic manipulator modeled by:

$$\dot{x} = \begin{bmatrix} x_2 \\ -2x_2 - 10\sin(x_1) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \tag{32}$$

where x_1 and x_2 are the angular position and velocity, and u is the torque. The goal is to drive x from $x_0 = [0.3\ 0]^\top$ to $x_T = [0.1\ 0]^\top$, minimizing (2) with $Q = \begin{bmatrix} 10 & 1 \\ 1 & 10 \end{bmatrix}$ and $R = 1$ Exploration noise $e = 2\sin t$ is used during training.

The activation function vector is:$\phi(x) = \begin{bmatrix} x_1^2 & x_1x_2 & x_2^2 & x_1^3 & x_1^2x_2 & x_1x_2^2 & x_2^3 \end{bmatrix}^\top$ with weights initialized randomly between 0 and 1. After training, the weights converge to:

$$W_+ = \begin{bmatrix} 2.7713 & 0.1235 & 0.2622 & 0.0829 & 0.0161 & -0.0037 & -0.0033 \end{bmatrix}^\top,$$

$$W_- = \begin{bmatrix} -2.5970 & -0.0688 & -0.2507 & -0.0108 & -0.0039 & -0.0001 & 0.0011 \end{bmatrix}^\top.$$

Fig. 2 shows that as T increases, the state trajectory remains near steady-state for most of the time horizon, deviating only near boundaries. This aligns with the *turnpike property* of optimal control, where the trajectory stays near a steady-state except near initial and final times.

6. Conclusion

We proposed an optimal controller design using reinforcement learning for two-point boundary nonlinear systems over finite-horizon time periods. The proposed design leverages the fast time scale occurring at the boundary conditions to avoid the need to solve the time-varying HJB equation. Furthermore, we design a learning-based control strategy that does not need knowledge of the system model. We show that the accuracy of the controller performance improves as the problem time horizon increases. We presented simulation results to support our claims using three examples. In the future, we plan to investigate the robustness of the proposed approach to noisy data and uncertain control input function.

References

Brian DO Anderson and Petar V Kokotovic. Optimal control problems over large time intervals. *Automatica*, 23(3):355–363, 1987.

Michael Athans and Peter L Falb. *Optimal control: an introduction to the theory and its applications*. Courier Corporation, 2013.

John T Betts. Survey of numerical methods for trajectory optimization. *Journal of guidance, control, and dynamics*, 21(2):193–207, 1998.

Anthony Siming Chen and Guido Herrmann. Adaptive optimal control via continuous-time q-learning for unknown nonlinear affine systems. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 1007–1012. IEEE, 2019.

Zhe Chen, Wenqian Xue, Ning Li, and Frank L Lewis. Two-loop reinforcement learning algorithm for finite-horizon optimal control of continuous-time affine nonlinear systems. *International Journal of Robust and Nonlinear Control*, 32(1):393–420, 2022.

Yu Jiang and Zhong-Ping Jiang. Global adaptive dynamic programming for continuous-time nonlinear systems. *IEEE Transactions on Automatic Control*, 60(11):2917–2929, 2015.

Hassan K Khalil. Nonlinear systems third edition. *Patience Hall*, 115, 2002.

Jong Woo Kim, Byung Jun Park, Haeun Yoo, Jay H Lee, and Jong Min Lee. Deep reinforcement learning based finite-horizon optimal tracking control for nonlinear system. *IFAC-PapersOnLine*, 51(25):257–262, 2018.

Petar Kokotović, Hassan K Khalil, and John O'reilly. *Singular perturbation methods in control: analysis and design*. SIAM, 1999.

Hanquan Lin, Qinglai Wei, and Derong Liu. Online identifier–actor–critic algorithm for optimal control of nonlinear systems. *Optimal Control Applications and Methods*, 38(3):317–335, 2017.

Biao Luo, Derong Liu, Tingwen Huang, and Ding Wang. Model-free optimal tracking control via critic-only q-learning. *IEEE transactions on neural networks and learning systems*, 27(10): 2134–2144, 2016.

Yongfeng Lv, Jing Na, Qinmin Yang, Xing Wu, and Yu Guo. Online adaptive optimal control for continuous-time nonlinear systems with completely unknown dynamics. *International Journal of Control*, 89(1):99–112, 2016.

Jing Na, Muhammad Nasiruddin Mahyuddin, Guido Herrmann, Xuemei Ren, and Phil Barber. Robust adaptive finite-time parameter estimation and control for robotic systems. *International Journal of Robust and Nonlinear Control*, 25(16):3045–3071, 2015.

Michael A Patterson and Anil V Rao. Gpops-ii: A matlab software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming. *ACM Transactions on Mathematical Software (TOMS)*, 41(1):1–37, 2014.

Anil V Rao. A survey of numerical methods for optimal control. *Advances in the astronautical Sciences*, 135(1):497–528, 2009.

Vasanth Reddy, Hoda Eldardiry, and Almuatazbellah Boker. Singular perturbation-based reinforcement learning of two-point boundary optimal control systems. In *2022 American Control Conference (ACC)*, pages 3323–3328. IEEE, 2022.

Kyriakos G Vamvoudakis. Q-learning for continuous-time linear systems: A model-free infinite horizon optimal control approach. *Systems & Control Letters*, 100:14–20, 2017.

Kyriakos G Vamvoudakis and Frank L Lewis. Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem. *Automatica*, 46(5):878–888, 2010.

Mathukumalli Vidyasagar. *Nonlinear systems analysis*. SIAM, 2002.

R Wilde and P Kokotovic. A dichotomy in linear control theory. *IEEE Transactions on Automatic control*, 17(3):382–383, 1972.

Jingang Zhao and Minggang Gan. Finite-horizon optimal control for continuous-time uncertain nonlinear systems using reinforcement learning. *International Journal of Systems Science*, 51 (13):2429–2440, 2020.

Qiming Zhao, Hao Xu, and Jagannathan Sarangapani. Finite-horizon near optimal adaptive control of uncertain linear discrete-time systems. *Optimal Control Applications and Methods*, 36(6): 853–872, 2015.