Learning-based Optimal Control of Time-Varying Linear Systems Over Large Time Intervals

Vasanth Reddy Baddam

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science and Applications

Hoda Eldardiry, Chair
Almuatazbellah (Muataz) Boker, Co-chair
Layne Watson

December 12th, 2022
Falls Church, Va

Keywords: optimal control; singular perturbation; reinforcement learning

Learning-based Optimal Control of Time-Varying Linear Systems Over Large Time Intervals

Vasanth Reddy Baddam

(ABSTRACT)

We solve the problem of two-point boundary optimal control of linear time-varying systems with unknown model dynamics using reinforcement learning. Leveraging singular perturbation theory techniques, we transform the time-varying optimal control problem into two time-invariant subproblems. This allows the utilization of an off-policy iteration method to learn the controller gains. We show that the performance of the learning-based controller approximates that of the model-based optimal controller and the approximation accuracy improves as the control problem's time horizon increases. We also provide a simulation example to verify the results.

Learning-based Optimal Control of Time-Varying Linear Systems Over Large Time Intervals

Vasanth Reddy Baddam

(GENERAL AUDIENCE ABSTRACT)

We use reinforcement learning to find two-point boundary optimum controls for linear time-varying systems with uncertain model dynamics. Using singular perturbation theory techniques, we divided the LTV control problem into two LTI subproblems. As a result, it is possible to identify the controller gains via a learning technique. We show that the training based controller's performance approaches that of the model-based optimal controller, with approximation accuracy growing with the temporal horizon of the control issue. In addition, we provide a simulated scenario to back up our findings.

Dedication

To my loving parents, supportive sister, cherished family, and dear friends, whose unwavering encouragement and endless love have been my guiding light throughout this academic journey

Acknowledgments

I'd want to thank my parents, friends, and advisors for pushing me ahead.

Contents

List of Figures

Chapter 1

Introduction

Over the past 10 years, several studies on the analysis and control of time-varying (dynamic) systems have been conducted. In this work, we emphasize on time-varying systems with limited horizons. Example applications of such systems include rocket landing [17] and energy conservation in electronic circuits [21], [12]. This category of problems is tackled using the time varying Riccati equation for linear time-variant systems or time-varying Hamiltonian Jacobi equation for nonlinear systems. When compared to time-invariant equations, the complexity level needed to solve time-variant equations is much higher. Even though the original system is not singularly disturbed, previous work, [22], presents two-value boundary issues demonstrating a two-time scale phenomena. Their method converts the underlying LTV system in this case into 2 LTI systems. The original system is stabilized by the initial boundary issue in forward time, whereas the terminal layer is stabilized by the final boundary problem in backward time. The accuracy of the approximation grows as the control issue time-horizon perturbation parameter lowers. Various works use this approximation in nonlinear fixed-end point control problems [3], nonlinear problems with singular arcs [1], and in optimal control of quasi-linear systems [11]. More recently, this approximation is mostly used to study the turnpike properties in wave equations [5].

If the model's dynamics are understood, an approximate solution for the two-value boundary is given. But in real-time applications, the dynamics might not be known or might even have modeling inadequacies. It is challenging to resolve the initial and final boundary difficulties

under these circumstances. [6] presented adaptive control principles in earlier research, which allow us to infer the model's control from its input-output data. The development of reinforcement learning, as noted in [16] study, has increased in popularity over time, opening up new opportunities for learning controllers. One of the first learning methods in reinforcement learning is called adaptive dynamic programming. It employs an iterative procedure with variable changes to identify the ideal control gain, [9] iterative method. When the model dynamics are unknown, learning techniques like Q-learning [18] and Actor-Critic [19] have generally proven successful in helping people learn the controller for continuous time systems. Additionally, the previous works [18], [19] mostly focus on learning the dynamics of the system using off-line data. This approach does not cope well if there are exogenous disturbances affecting the system. To tackle this issue, [20] proposes to train the closed-loop model on the running data in an online fashion. However, in these cases, the system is assumed to be time invariant.

The primary goal of this study is to develop a controller for a time-varying system with a limited horizon when the boundary conditions are known but the system model is not. Various extensions and adaptations of this particular research were made pertaining to time varying systems. While the work [8] proposed a data-driven approach, the other work [23], [14] used a policy iteration method to find the controllers. These are, however, limited to only discrete time-varying systems. Previous work [15], on the other hand, investigated continuous time periodic systems. It used adaptive programming to learn the controller and off-policy value iteration [13] but considered the infinite horizon without any boundary conditions. For finite horizon time changing systems, generally, less work has been done, and consequently fewer findings are accessible. Recently, [4] proposed dual loop iterative algorithm but considered the fact that the state transition matrix is unknown.

Based on pioneering work, we applied the concept of dealing with restricted LTV situations

with stated boundary conditions, [22] to solve the aforementioned restrictions. The theory predicts that the closed-loop time-varying system will display a two-time scale behavior when the control time horizon is long enough. That is, in comparison to the time scale of the control effort, the system dynamics will change more quickly. The system may then be broken down into two time-invariant issues, as demonstrated by the [22]. Thus, by combining the answers to the initial and final layer issues, the final original state may be roughly estimated. We take advantage of this strategy and apply the offline learning technique described in [7] to find the solution for two boundary LTI systems. We shall be able to discover the controllers of both challenges by separating the intricacy of the existing system into two basic problems. We show that when the time interval required to optimize the cost function rises, the learnt controller regains the efficiency of the model-based optimum controller. We use a simple linear time-variant system as an example to demonstrate the aforementioned idea. In conclusion, our suggested model makes two main contributions.:

1. We offer a method for solving the two-boundary optimum control issue for LTV systems that is free of the system's model.

2. Based on our understanding of physical dynamics, we provide a reinforcement learning paradigm. This framework is straightforward to implement and produces a near-optimal solution that merges to the ideal solution as the control time interval increases.

The rest of this dissertation is organized as follows. Chapter 2.1 portrays the system setup and problem formulation. Chapter 3.1 describes the two-time scale reduction of the original problem. Section 4.1 outlines the offline learning algorithm to estimate the control gains of the system and then followed by the Chapter 5.0.1 evaluating the performance of the proposed method.

Chapter 2

Problem Formulation

2.1 Problem Formulation

Consider about the differential equation's representation of the linear time-varying system:

$$\dot{y} = M(t)y(t) + N(t)v(t) \tag{2.1}$$

where $y(t) \in \mathbb{R}^n$, $v(t) \in \mathbb{R}^m$, $M(t) \in \mathbb{R}^{n\times n}$ and $N(t) \in \mathbb{R}^{n\times m}$ are the system states, control input, state matrix and input matrix, correspondingly. The matrices $M(t)$ and $N(t)$ can be unknown, and are $\forall\ t \in [0, T]$. The control aim is to minimize the objective function while designing $v(t)$ to move the states from the starting state $y(0) = y_0$ to the end state $y(T) = y_T$ in a time period of T.

$$J = \int_0^T y^\top(t)G(t)y(t) + v^\top(t)H(t)v(t)\ \ dt \tag{2.2}$$

Where $G(t) = G^\top(t) \succeq 0$ and $H(t) \succ 0\ \ \forall\ t \in [0, T]$. It is assumed that the matrices $G(t)$ and $H(t)$ are spherical functions of time. In addition, we assume the following:

Assumption 1. The pair $(M(t), N(t))\ \ \forall\ t \in [0, T]$ can be controlled and the pair $(M(t), \sqrt{G(t)})\ \ \forall\ t \in [0, T]$ can be observed.

Assumption 2. When compared to the system dynamics, the time T required to achieve

the control aim is long.

According to the theory of optimum control [2], assumption 2 is a common presumption. The system is thought to be slowly altering in relation to the control aim, according to assumption 2 . The next section will go into more detail about this. We define the system's Hamiltonian function in order to address the optimum control challenge. (2.1) as [2]:

$$\mathcal{H} = y^\top(t)G(t)y(t) + v^\top(t)H(t)v(t) + \lambda^\top(t)(M(t)y(t) + N(t)v(t)) \tag{2.3}$$

where, $\lambda(t) \in \mathbb{R}^n$ is defined as:

$$\dot{\lambda}(t) = -\nabla_y \mathcal{H} = -G(t)y(t) - M^\top(t)\lambda(t). \tag{2.4}$$

The controller $v(t)$ is given by $\nabla_v \mathcal{H} = 0$:

$$v(t) = -H(t)N^\top(t)\lambda(t). \tag{2.5}$$

Without having to know the system matrices $M(t)$ and $N(t)$, our goal is to learn the best controller $v(t)$ provided by (2.5) for the system (2.1).

Chapter 3

Singular Perturbation Design

3.1 Singular Perturbation-based Design

The closed-loop system dynamics are known to act in two time scales for control problems that are specified over a limited time period $\in [0, T]$ (see [10]). The distance between the time scales depends on how long the time interval T is. In this study, we take use of this fact by thinking about the situation where the amount of time required to change the state of the system from one point to another is negligible in comparison to T. This makes it possible to represent the system in a single disrupted form. Accordingly, prior research has demonstrated that when T is reasonably long and the matrices $M(t)$ and $N(t)$ are known, it is conceivable to arrive at a sub-optimal solution to the control issue. In the following sections, we demonstrate how applying a reinforcement learning method makes it feasible to obtain a comparable outcome without having knowledge of the system matrix. To do this, we must construct the system's single perturbation model. By adding the scaled time, we first normalize the time range 0 to T to the interval $[0, 1]$.

$$\tau = \frac{t}{T} \tag{3.1}$$

and defining

$$\varepsilon = 1/T \tag{3.2}$$

and then reconstructing (2.1), (2.2), (2.4), and (2.5) to obtain the singulat perturbarion form of system (3.3)

$$\varepsilon \begin{bmatrix} \dot{y} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} M(\tau) & 0 \\ -G(\tau) & -M^T(\tau) \end{bmatrix} \begin{bmatrix} y \\ \lambda \end{bmatrix} + \begin{bmatrix} N(\tau) \\ 0 \end{bmatrix} v, \tag{3.3}$$

$$v(\tau) = - H(\tau)N^\top(\tau)\lambda(\tau), \tag{3.4}$$

$$J = T * \int_0^1 y^\top(\tau)G(\tau)y(t) + v^\top(\tau)H(\tau)v(\tau)\ dt, \tag{3.5}$$

where $y(0) = y_0$ and $y(1) = y_T$.

Assumption 3. The eigenvalues of the Hamiltonian matrix

$$M_H \triangleq \begin{bmatrix} M(\tau) & -N(\tau)H^{-1}(\tau)N^\top(\tau) \\ -G(\tau) & -M^\top(\tau) \end{bmatrix} \tag{3.6}$$

lie off the imaginary axis $\forall\ t \in [0, 1]$.

According to this presumption, the closed-loop matrix M_H, which results from replacing (3.4) in (3.3), is nonsingular $\forall\ t \in [0, 1]$. Next, we will decouple the dynamics of the (3.3). This job is completed using the transformation [10]

$$\begin{bmatrix} y \\ \lambda \end{bmatrix} = \begin{bmatrix} I & I \\ P_a(\tau, \varepsilon) & P_b(\tau, \varepsilon) \end{bmatrix} \begin{bmatrix} y_a \\ y_b \end{bmatrix}. \tag{3.7}$$

The transformation (3.7) is nonsingular under the presumptions of this article for sufficiently tiny ε, as demonstrated in [Lemma 2.3, [10]]. Therefore, the system (3.7), system (3.3) with

(3.4) may be changed into

$$\varepsilon \frac{d}{d\tau} y_a = M(\tau) y_a + N(\tau) v_a(\tau), \tag{3.8}$$

$$\varepsilon \frac{d}{d\tau} y_b = M(\tau) y_b + N(\tau) v_b(\tau), \tag{3.9}$$

where

$$v_a = - H(\tau) N^{\top}(\tau) P_a(\tau, \varepsilon), \tag{3.10}$$

$$v_b = - H(\tau) N^{\top}(\tau) P_b(\tau, \varepsilon), \tag{3.11}$$

where $P_a(\tau, \varepsilon) \geq 0$ and $P_b(\tau, \varepsilon) \leq 0$ are the below differential Riccati equation's roots

$$\begin{aligned} \varepsilon \dot{P} = & - M^{\top}(\tau) P - P M(\tau) \\ & + P N(\tau) R(\tau) N^{\top}(\tau) P - G(\tau). \end{aligned} \tag{3.12}$$

Remark 3.1. Let $\epsilon \to 0$, in (3.12) so that we have

$$- M^{\top}(\tau) P - P M(\tau) + P N(\tau) R(\tau) N^{\top}(\tau) P - G(\tau) = 0. \tag{3.13}$$

The existence of the (3.13) solution and the Hurwitzian nature of the $M(\tau) - N(\tau) H^{-1}(\tau) G^{\top}(t) P(\tau)$ for each tau in [0,1] are demonstrated in the [10]. Two roots of the type $P = P_a(\tau) \geq 0$ and $P = P_b(\tau) \leq 0$ are also included in the solution.

We transform next the singular perturbation system (3.8)-(3.9) by taking the time-scale change into account in the boundary layer system

$$\gamma = \frac{\tau}{\varepsilon}, \quad \beta = \frac{1 - \tau}{\varepsilon}. \tag{3.14}$$

Setting $\varepsilon \to 0$ in (3.8)-(3.12) results in the boundary layer system in light of (3.14) and Remark 3.1. This results in the issue with the *initial regulator problem*.

$$\frac{d}{d\gamma}y_a = M(0)y_a + N(0)v_a, \quad y_a(0) = x_0, \tag{3.15}$$

$$v_a(\gamma) \triangleq -K_a y_a(\gamma) = -H(0)N^\top(0)P_a(0)y_a(\gamma), \tag{3.16}$$

$$J(y_a, u_a) = \int_0^\infty y_a^\top G(0)y_a + v_a^\top H(0)v_a \, d\gamma, \tag{3.17}$$

and the *terminal regulator problem*

$$\frac{d}{d\beta}y_b = -M(1)y_b - N(1)u_b, \quad x_N(1) = x_T \tag{3.18}$$

$$u_b(\beta) \triangleq -K_b y_b(\beta) = -R(1)B^\top(1)P_N(1)y_b(\beta), \tag{3.19}$$

$$J(y_b, u_b) = \int_0^\infty y_b^\top G(1)y_b + u_b^\top H(1)v_b \, d\beta. \tag{3.20}$$

Theorem 2.1 in [10] demonstrates that, for reasonably small ε and assuming the dynamic model is available, the solution to two reduced LTI systems, that are currently problems for linear time-invariant systems, can be found, will resemble that of the original control problem, .(2.1)-(2.5) The initial and terminal regulator issues (3.15)-(3.20) are solved using a reinforcement learning approach in the next section, which does not need the knowledge of $M(t)$ and $N(t)$.

Chapter 4

Two Point Boundary Problem

4.1 Learning Design

In this part, we learn the initial and terminal regulator problems (3.15)-(3.20) using a reinforcement learning strategy. We employ the initial state $y(t)$ for the learning process under the guidance of singular perturbation theory. It should be emphasized that the time scaling factors γ and β are respectively the forward and backward times of t, according to (3.1) and (3.14). The initial and terminal learning regulator issues will then each be solved independently.

4.1.1 Initial Regulator Learning Problem

Without knowledge of the system dynamics, the goal is to learn the feedback control gain $K_a \triangleq H(0)N^\top(0)P_a(0)$ for the system specified in (3.15). The cost function stated in (3.17) is optimized utilizing control $v_a = -K_a y$ and the measurement data of system states y and v_a. Note that the algebraic Riccati equation's solution, $P_a(0)$, is:

$$\begin{aligned} M^\top(0)P_a(0) + P_a(0)M(0) - P_a(0)N(0)H(0)N^\top(0)P_a(0) \\ + G(0) = 0. \end{aligned} \tag{4.1}$$

where the answer to the Riccati equation (4.1) is P_a^*. $K_a^* = -H(0)N^\top(0)P_a^*$ provides the best feedback gain in this scenario. The Kleimanns procedure, which is iterative and [9] , is then used to find the best values.

1. Solve for P^k of the Lyapunov equation

$$M_k^\top(0)P_a^k + P_a^k M_k(0) + Q(0) + P_a^k N(0)H(0)N^\top(0)P_a^k = 0. \tag{4.2}$$

2. Update the feedback gain:

$$K_a^{k+1} = H(0)N^\top(0)P_a^k, \tag{4.3}$$

where$M_k(0) = M(0) - N(0)K_a^k$. If the model dynamics are known, then the matrix P_a^k and the gain K_a^{k+1} may be learnt iteratively by using the procedures above. Assuming that the model dynamics are unknown, we take a few measures to avoid using $M(0)$ and $N(0)$ in the learning of the controller.

Initialization: Take into account a random control signal u_0 to stimulate the system (3.15). In order to obtain the following, we define the control policy $u_a = u_0 - K_a^k y_a + K_a^k y_a$ with $K_a^k > 0$.

$$\begin{aligned} \dot{y}_a &= M(0)y_a + N(0)(v_0 - K_a^k y_a + K_a^k y_a) \\ &= A_k(0)y_a + N(0)(v_0 + K_a^k y_a). \end{aligned} \tag{4.4}$$

We create the Lyapunov function $V^k(y_a) = y_a^\top P_a^k y_a$ in an effort to break the dependence between $M(0)$ and $N(0)$. Taking this function's system-wide derivative (4.4) results in

$$\frac{d}{dt}(y_a^\top P_a^k y_a) = y_a^\top (A_k^\top(0)P_a^k + P_a^k A_k(0))y_a + 2(v_0 + K_a^k)^\top N(0)^\top P^k y_a. \tag{4.5}$$

Replacing $-G_k(0) = (M_k^\top(0)P_a^k + P_a^k M_k(0)) = -G(0) - K^\top H(0)K$ from (4.2) and $N^\top(0)P^k =$

$H(0)K_a^{k+1}$ from (4.3) leads to

$$\frac{d}{dt}(y_a^\top P_a^k y_a) = -y_a^\top G_k(0) y_a + 2(v_0 + K_a^k)^\top H(0) K_a^{k+1} y_a \tag{4.6}$$

You should be aware that (4.6) is independent of $M(0)$ and $N(0)$. Integrating (4.6) on the interval on both sides $[t, t+\delta t]$

$$\begin{aligned} y_a^\top(t+\delta t) P_a^k y_a(t+\delta t) - y^\top P_a(t) P_a^k y_a(t) = \\ - \int_t^{t+\delta t} y_a^\top G_k(0) y_a \, dw + 2 \int_t^{t+\delta t} \left(K_a^k y_a + v_0\right)^T H(0) K_a^{k+1} y_a \, dw \end{aligned} \tag{4.7}$$

Rearranging the terms (4.7) leads to the offline policy iteration equation

$$\begin{aligned} y_a^\top(t+\delta t) P_a^k y_a(t+\delta t) - y_a^\top(t) P_a^k y_a(t) - 2 \int_t^{t+\delta t} \left(K_a^k y_a + v_0\right)^T H(0) K_a^{k+1} y_a \, dw \\ = - \int_t^{t+\delta t} y_a^\top G_k(0) y_a \, dw \end{aligned} \tag{4.8}$$

We then express (4.8) in compact form using the Kronecker product ($\otimes$)

$$y_a^\top P_a^k y_a = \left(y_a^\top \otimes y_a^\top\right) \mathrm{vec}\left(P_a^k\right), \tag{4.9}$$

and then (4.9) is used to transform the term from (4.8)

$y_a^\top(t+\delta t) P_a^k y_a(t+\delta t) - y_a^\top(t) P_a^k x_M(t)$

into:

$$y_a^\top(t+\delta t) P_a^k y_a(t+\delta t) - y_a^\top(t) P_a^k y_a(t) = \left[y_a^\top \otimes y_a^\top \Big|_t^{t+\delta t} \right] \left[\mathrm{vec}(P_a^k)\right]. \tag{4.10}$$

And the next term from (4.8)

$$\begin{aligned}\left(v_0 + K_a^k x\right)^\top H(0) K_a^{k+1} x_a = \\ \left[\left(x_a \top \otimes x_a^\top\right)\left(I_n \otimes \left(K^k\right)^\top H(0)\right) + \left(x_a^\top \otimes u_0^\top\right)\left(I_n \otimes H(0)\right)\right] K_a^{k+1}\end{aligned} \tag{4.11}$$

Using (4.11) to transform the term from (4.8)
$-2\int_t^{t+\delta t}\left(K_a^k y_a + u_0\right)^T H(0) K_a^{k+1} y_a \, dw$
into:

$$\begin{aligned}&-2\int_t^{t+\delta t}\left(K_a^k y_a + v_0\right)^T H(0) K_a^{k+1} y_a \, dw = \\ &-\left[2\left(\int_{t_1}^{t_1+\tilde{t}} y_a^\top \otimes y_a^\top dw\right)\left(I_n \otimes (K^k)^\top H(0)\right)\right]\kappa - \left[2\left(\int_{t_1}^{t_1+\tilde{t}} y_a^\top \otimes v_0^\top dw\right)\left(I_n \otimes H(0)\right)\right]\kappa\end{aligned} \tag{4.12}$$

where $\kappa = \left[\mathrm{vec}(K_a^{k+1})\right]$ and finally the last term from (4.8):

$$y_a^\top G_k(0) y_a = \left(y_a^\top \otimes y_a^\top\right)\mathrm{vec}\left(G_k(0)\right). \tag{4.13}$$

Using (4.13) to transform the term $-\int_t^{t+\delta t} y_a^\top G_k(0) y_a \, dw$ into:

$$-\int_t^{t+\delta t} y_a^\top G_k(0) y_a \, dw = -\left[\int_{t_1}^{t_1+\delta t} y_a^\top \otimes y_a^\top dw\right]\mathrm{vec}\left(G_k(0)\right) \tag{4.14}$$

Using (4.10), (4.12) and (4.14) to express the offline policy iteration (4.8) in a compact form:

$$\begin{aligned}\left[y_a^\top \otimes y_a^\top\Big|_t^{t+\delta t}\right]\left[\mathrm{vec}(P_a^k)\right] &- 2\left[\left(\int_{t_1}^{t_1+\tilde{t}} y_a^\top \otimes y_a^\top dw\right)\left(I_n \otimes (K^k)^\top H(0)\right)\right]\kappa \\ &- 2\left[\left(\int_{t_1}^{t_1+\tilde{t}} y_a^\top \otimes v_0^\top dw\right)\left(I_n \otimes H(0)\right)\right]\kappa \\ &= -\left[y_a^\top \otimes y_a^\top\Big|_t^{t+\delta t}\right]\mathrm{vec}\left(G_k(0)\right)\end{aligned} \tag{4.15}$$

We can now express (4.15) in the compact matrix form as

$$\tilde{\psi}\begin{bmatrix} \text{vec}\left(P_a^k\right) \\ \text{vec}\left(K_a^{k+1}\right) \end{bmatrix} = \tilde{\Gamma} \tag{4.16}$$

where, $\tilde{\psi} = \left[\tilde{\delta}_{yy}, -2\tilde{I}_{yy}\left(I_n \otimes \left(K^k\right)^\top H(0)\right) - 2\tilde{I}_{x\bar{u}_0}\left(I_n \otimes H\right)\right]$; $\tilde{\delta}_{yy} = \left[y_a^\top \otimes y_a^\top\big|_t^{t+\delta t}\right]$
$\tilde{\Gamma} = \tilde{\delta}_{x,x}\text{vec}(G_k(0)$; $\tilde{I}_{yy} = -2\left[\left(\int_{t_1}^{t_1+\tilde{t}} y_a^\top \otimes y_a^\top dw\right)\left(I_n \otimes (K^k)^\top H(0)\right)\right]$ and
$\tilde{I}_{xv_0} = -2\left[\left(\int_{t_1}^{t_1+\tilde{t}} y_a^\top \otimes u_0^\top dw\right)\left(I_n \otimes H(0)\right)\right]$.
Offline policy iteration (4.8) is stated in compact form in this manner (4.16). It should be noted that the state $y(t)$ will be used to facilitate learning. The next step is to gather this information in order to calculate the feedback control gain.

Data Collection: For the time period $[t_i, t_j]$ with the sampling interval $t_{i+1} - t_i = \delta t = \tilde{t}$, we gather data during learning, including state space $y(t)$ and control policy u_0. After that, the matrices are computed, δ_{yy}, I_{yy} and I_{yv_0} as follows:

$$\delta_{yy} = \left[y^\top \otimes y^\top\big|_{t_1}^{t_1+\tilde{t}}, \quad \ldots, \quad y^\top \otimes y^\top\big|_{t_j}^{t_j+\tilde{t}}\right]^\top, \tag{4.17}$$

$$I_{yy} = \left[\int_{t_1}^{t_1+\tilde{t}} y^\top \otimes y^\top dw, \ldots, \quad \int_{t_l}^{t_l+\tilde{t}} y^\top \otimes y^\top dw\right]^\top, \tag{4.18}$$

$$I_{xv_0} = \left[\int_{t_1}^{t_1+\tilde{t}} y^\top \otimes v_0^\top dw, \ldots, \quad \int_{t_l}^{t_l+\tilde{t}} y^\top \otimes v_0^\top dw\right]^\top. \tag{4.19}$$

Assumption 4. There exists a large number $j > 0$ in such a way that $\text{rank}([I_{yy} \quad I_{yv_0}]) = \frac{n(n+1)}{2} + mn$.

Assumption 4 ensures the collection of enough data for the learning process [7]. **Policy Iteration**: There are two further stages that are part of this phase. I Policy evaluation; assessing the performance of the existing policy using information from the current data

phase. ii) Policy improvement: After assessing the present policy, we replace it with the new one. The feedback gain may be calculated as $K_a = K_a^{k+1}$ at the conclusion of learning (i.e., at the conclusion of convergence), and the system controller (3.15) is $u_a^l = -K_a y$.

4.1.2 Final Boundary Learning Problem

We proceed in the same manner as when solving the initial regulator issue, with the exception that the control gain initialization must ensure that $K_b < 0$. Keep in mind that the system's controller (3.18) is defined as $u_b^l = -K_b^* y$, where K_b^* is the gain matrix. Algorithm1 contains the pseduocode for the training algorithm. The closed-loop performance

Algorithm 1 Learning Algorithm for two regulator problems

while $rank([I_{yy} \quad I_{xv_0}]) < \frac{n(n+1)}{2} + mn$ **do**
  Collection of data $y(t)$ using the excitation control $v_a = v_0$
  Construction of the the matrices δ_{yy}, I_{yy} and I_{yv_0}
end while
Initialize $K_a > 0$
while $|P_a^k - P_a^{k+1}| <$ Threshold **do**
  Estimate the values of P_a^k and K_a^{k+1} through (4.16)
end while$u_a^l = -K_a^{k+1} y$
Initialize $K_b < 0$
while $|P_b^k - P_b^{k+1}| <$ Threshold **do**
  Estimate the values of P_b^k and K_b^{k+1} through (4.16)
end while$u_b^l = -K_b^{k+1} y$

of the learned-control system, composed of (2.1) will resemble that of the original control system, composed of (2.1) and (2.5), after the learning procedure described in this section, provided that ε is small enough or T is long enough. Theorem 1 below, along with the simulation example provided in Section 5.0.1, will serve to verify this.

4.1.3 Analysis of the closed-loop system performance

Consider the closed-loop system that may be expressed as K_a^* and K_b^* that is based on the LTV system (2.1) and training-based controllers.

$$\frac{d}{d\gamma}y_a^l = (M(0) - N(0)K_a^*)y_a^l, \quad y_a^l(0) = y_0, \tag{4.20}$$

$$\frac{d}{d\beta}y_b^l = (M(1) - N(1)K_b^*)y_b^l, \quad y_b^l(1) = y_T. \tag{4.21}$$

Recall that γ and β are the forward and reverse times as defined in (3.14). Now we have the following theorem.

Theorem 4.1. *Under Assumptions 1-4, there exists $\varepsilon_1 > 0$ such that for all $\varepsilon \in (0, \varepsilon_1]$, the solution $x(\tau)$ of (3.3)-(3.4) satisfies*

$$y(\tau) = y_a^l(\gamma) + y_b^l(\beta) + \mathcal{O}(\varepsilon), \tag{4.22}$$

$$\begin{aligned} v(\tau) =& - H^{-1}(0)N^\top(0)P_a(0)y_a^l(\gamma) \\ & - H^{-1}(1)N^\top(1)P_N(1)y_b^l(\beta) + \mathcal{O}(\varepsilon), \\ \triangleq & \; v_a(\gamma) + v_b(\beta) + \mathcal{O}(\varepsilon) \end{aligned} \tag{4.23}$$

where y_a^l and y_b^l are the solutions of the closed-loop systems (4.20) and (4.21), respectively. [1]

Remark 4.2. Theorem 1 explains how the closed loop system performance (4.20)-(4.21) approximates the closed loop system's performance (3.3)-(3.4) when under optimum management. The controllers of the beginning and terminal learning regulator issues also approximate the optimum control, as shown by equation (4.23). All of these findings demonstrate that the trained controllers operate sub-optimally and that, when ε (or $1/T$) increases, the

[1] The "order of magnitude" is denoted by the symbol $\mathcal{O}$, which is defined as: $\delta_1(\varepsilon) = \mathcal{O}(\delta_2(\varepsilon))$ if there exist positive constants k and c such that $|\delta_1(\varepsilon)| \leq k\,|\delta_2(\varepsilon)|\,,\ \forall\ |\varepsilon| < c$.

closed-loop performance approaches the optimal.

We will now go through a few findings that demonstrate the learnt closed-loop system's performance and relative convergence in order to demonstrate Theorem 4.1.

Lemma 4.3. *The matrices P_a^k and P_b^k, as well as the feedback control gains K_a^k and K_b^k, converge at the conclusion of the training process according to Algorithm 1 as follows:*

$$\lim_{k\to\infty} K_a^k = K_a^*, \quad \lim_{k\to\infty} P_a^k = P_a^*, \tag{4.24}$$

$$\lim_{k\to\infty} K_b^k = K_b^*, \quad \lim_{k\to\infty} P_b^k = P_b^*, \tag{4.25}$$

where K_a^* and P_a^* are the optimal controller gain and Riccati solution of the initial and terminal regulator problems (3.15)-(3.17) and (3.18)-(3.20), respectively.

Proof. Below, Kleinman's algorithm [9] is recited to demonstrate the convergence of K_a^* and P_a^*.

Let P_a^0 be the finite and positive definite solution of the Lyapunov equation (4.2) at $k = 0$, which is given as

$$P_a^0 = \int_0^\infty e^{(M_0^\top(0))t} \left(G(0) + (K_a^0)^\top H(0) K_a^0\right) e^{(M_0(0))t} dt. \tag{4.26}$$

and the feedback control gain at $k = 1$ is given by $K_a^1 = HN^\top P_a^0$. Similarly P_a^1 is the solution at $k = 1$, which is given by

$$P_a^1 = \int_0^\infty e^{(M_1^\top(0))t} \left(G(0) + (K_a^1)^\top H(0) K_a^1\right) e^{(M_1(0))t} dt. \tag{4.27}$$

Subtracting equation (4.27) from (4.26) gives

$$\begin{aligned} &P_a^0 - P_a^1 \quad = \\ &\int_0^\infty e^{(M_1^\top(0))t} \left(K_a^0 - K_a^1\right)^\top H \left(K_a^0 - K_a^1\right) e^{(M_1(0))t} dt \geq 0. \end{aligned} \tag{4.28}$$

From the above equation we see that $P_a^0 \geq P_a^1$. Similarly

$$\begin{aligned} &P_a^1 - P_a^* \quad = \\ &\int_0^\infty e^{(M_1^\top(0))t} \left(K_a^1 - K_a^*\right)^\top H \left(K_a^1 - K_a^*\right) e^{(M_1(0))t} dt \geq 0. \end{aligned} \tag{4.29}$$

From the equations (4.28) and (4.29) It is evident that $P_a^0 \geq P_a^1 \geq P_a^*$. We can see that the series P_a^* lower limit the sequence Pa^k, which is monotonically declining. Since $k = \infty$, the convergence value of P_a^k and K_a^K is given by

$$\lim_{k\to\infty} P_a^k = P_a^*. \tag{4.30}$$

$$\lim_{k\to\infty} K_a^k = \lim_{k\to\infty} H(0)N^\top(0)P_a^k = H(0)N^\top(0) \lim_{k\to\infty} P_a^k \tag{4.31}$$

Using equation (4.30) we can deduce that

$$\lim_{k\to\infty} K_a^k = H(0)N^\top(0)P_a^* = K_a^*. \tag{4.32}$$

We can also demonstrate convergence for the final regulator problem in the same fashion. □

Lemma 4.4 (Lemma 2.2 in [10]). *Under Assumptions 1-3, there exists $\varepsilon_2 > 0$ such that for all $\varepsilon \in (0, \varepsilon_2]$ and $\forall\, \tau \in [0, 1]$, the roots of (3.12), $P_a(\tau, \varepsilon)$ and $P_b(\tau, \varepsilon)$, satisfy*

$$P_a(\tau, \varepsilon) = P_a(\tau) + \mathcal{O}(\varepsilon), \tag{4.33}$$

$$P_b(\tau, \varepsilon) = P_b(\tau) + \mathcal{O}(\varepsilon), \tag{4.34}$$

where $P_a(\tau)$ and $P_b(\tau)$ are the roots of (3.13).

Chapter 5

Experiments

5.0.1 Example - RL Circuit

Consider looking at an electrical circuit that includes a resistor (R) and an inductor ($L(t)$). In Fig. 5.1, the RL circuit is illustrated. The differential state equation is given as:

$$\dot{y} = -\frac{R}{L(t)}y + \frac{1}{L(t)}v, \tag{5.1}$$

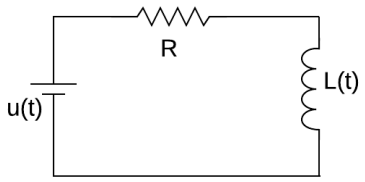


Figure 5.1: RL Circuit

where the state (circuit current) is $y(t)$, and the control input is $v(t)$ (circuit voltage). wherw $M(t) = -\frac{H}{L(t)}$ and $N(t) = \frac{1}{L(t)}$, a ssumed to be unknown are the values of the time-varying dissipating inductor L and resistor. In order to set the current $y(t)$ at desirable values at both the beginning and final times while maintaining the current and voltage at their lowest levels in between, a controller $v(t)$ is needed. So, in order to minimize the objective function, the controller is required.

$$J = \frac{1}{2}\int_0^T G(t)y^2(t) + H(t)v^2(t)\ dt, \tag{5.2}$$

while ensuring that the system's starting and final conditions are, respectively, $y_0 = 0.5$ and $y_T = 0.9$. The parameters for the cost function are set to $G(t)$=1 and $H(t)$=1. We'll attempt a new solution at time T in the end. This is to prove that when T grows big, the

approximation result will be more accurate as mentioned in Theorem 4.1. For the sake of the simulation, we assume that the inductor is supplied by the equations $L(t) = (1+0.2t)H$ and $R = 1$. Using the signal $v_0 = \sum_0^{100} sin(wt)$, where w is a random number between 1 and 100, we excite the system during the offline data collection, and we then collect the data at intervals of 0.01 seconds. Using the learning approach given in Section 4.1, the control gains Ka and Kb are calculated and determined to be 0.4 and -2.4, respectively.

In order to compare our findings, we determined the values for K_a and K_b by solving the problems eqrefiniregulator-eqrefj b under the assumption that $M(t)$ and $N(t)$ are known. Therefore, the Riccati equation for the system where the controller must minimize the objective function and the objective (5.1)

$$P^2 + 2(1 + 0.2t)P - (1 + 0.2t)^2 = 0 \tag{5.3}$$

The two roots for the Riccati equation (5.3) are: $P = (-1 \pm \sqrt{2})(1 + 0.2t)$. Since $P_a \geq 0$ and $P_b \leq 0$ the values of $P_a(0)$ and $P_b(1)$ are $P_a(0) = 0.414$, $P_b(1) = -2.89$. This yields the control gains $K_a = \frac{1}{r(0)} N^\top(0) P_a(0) = 0.4$ and $K_b = \frac{1}{r(1)} N^\top(1) P_b(1) = -2.4$. Notice that that the learning controller gains converge to those obtained using the singular perturbation method. This implies that the closed-loop response will be identical for these two controllers. This is indeed observed in the simulation shown in Fig. 5.2.

As demonstrated in Theorem 1 to simulate the learning- and singular perturbation-based controllers, the solutions are superimposed on one another to find the original state trajectory.

Remark 5.1. It is worth mentioning that the response of the closed-loop system can also be simulated by finding the time $t_c > 0$ when $y_a(t_c) = y_b(1 - t_c)$ and then the the controller $v_a = -K_a y_a$ is applied in the time interval $[0, t_c]$ and the controller $v_b = -K_a y_b$ is applied in

the time interval $[t_c, 1]$ [22].

In order to clearly compare the controllers, we normalized the time range to $[0, 1]$ rather than $[0, T]$, where $T = 1/\varepsilon$.

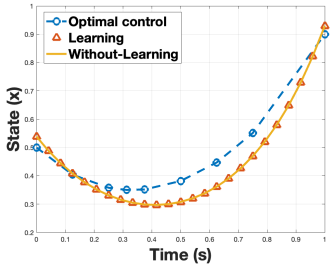
Using the $M(t)$ and $N(t)$ knowledge, the two value boundary problem solver from the MATLAB function *bvp4c* is used to find the best state trajectory and the optimal controller. The state space trajectory and controller generated using our learning technique are then compared to those created using the approximation method based on singular perturbation and the optimum controller created using the knowledge of $M(t)$ and $N(t)$. We can observe from Fig. 5.2 that when ε is dropped (or control time period T grows), the learning-based controller closely approximates the optimum controller. This suggests that the learning controller's performance is suboptimal and improves as the time interval T increases to the ideal performance.

5.0.2 Example - Mass System

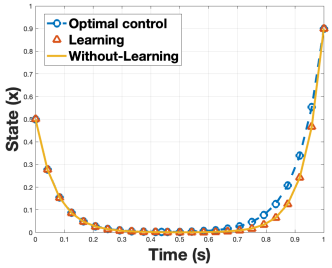
Consider an object with a dissipating mass $m(t)$ on a frictionless environment that is described by the following equation [2]

$$\dot{y} = \frac{1}{m(t)} y + v(t) \tag{5.4}$$

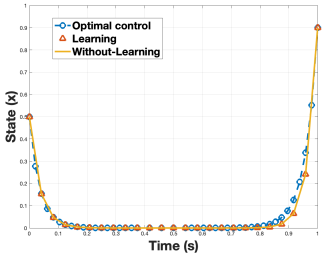
where $y(t)$ is the position of the body. We assume that the time-varying dissipating mass is unknown. In the absence of the external disturbances, the force $v(t)$ is applied on the object to move it from one position to another. A controller is sought to achieve the control objective while keeping the force and position minimum in between the initial and final values.So, in order to minimize the objective function, the controller is required..



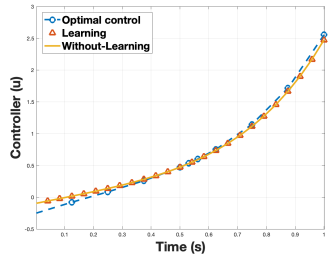
(a) State trajectory for $\varepsilon = 0.5$



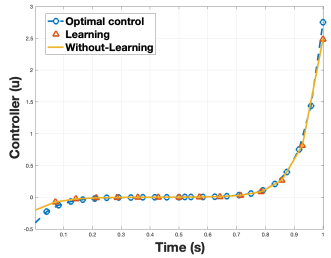
(b) State trajectory for $\varepsilon = 0.1$



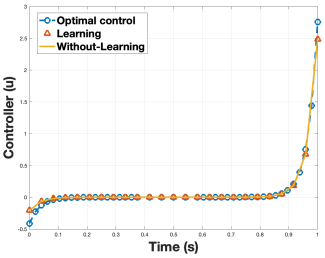
(c) State trajectory for $\varepsilon = 0.05$



(d) Controller for $\varepsilon = 0.5$



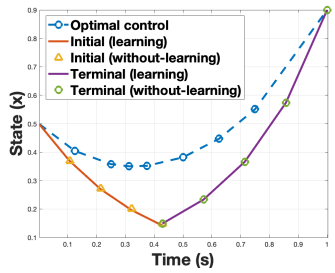
(e) Controller for $\varepsilon = 0.1$



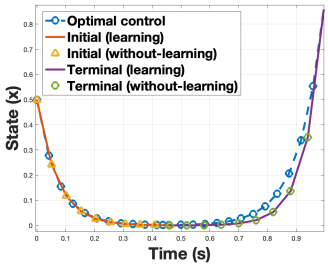
(f) Controller for $\varepsilon = 0.05$

Figure 5.2: Plots (a), (b), and (c) illustrate the trajectory of the state $y(t)$ of the system for a range of ε values, whereas plots (d), (e), and (f) show the control law $v(t)$ for a range of ε values.

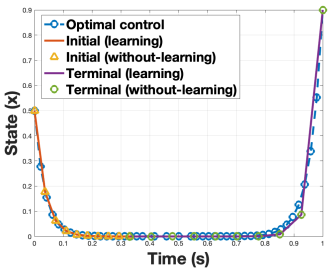
The cost function parameters are defined as $G(t) = 1$, $H(t) = 1$, and the initial and final conditions of the system are given as $y_0 = 0.5$ and $y_T = 0.9$. For the sake of the simulation, we assume that $m(t) = \frac{-1}{(1+0.2t)}$, and $N(t) = 1$ represent the mass, respectively. In order to excite the system during offline data collection, we use the controller $u_0 = \sum_0^{100} \sin(wt)$ with a time interval of 0.01 seconds, where w is the frequency. Using the learning approach outlined in Section IV, the control gains K_a and K_b are calculated and determined to be 0.410 and -2.46, respectively. Analytically, the control gains K_a and K_b are calculated as 0.413 and 2.414, respectively.



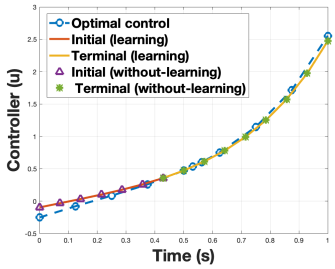
(a) State trajectory for $\varepsilon = 0.5$



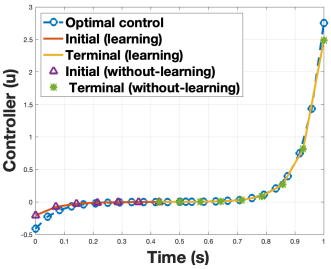
(b) State trajectory for $\varepsilon = 0.1$



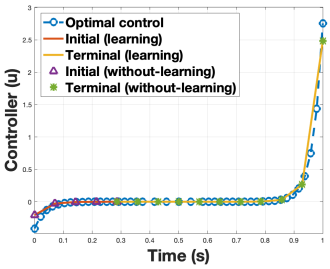
(c) State trajectory for $\varepsilon = 0.05$



(d) Controller for $\varepsilon = 0.5$



(e) Controller for $\varepsilon = 0.1$



(f) Controller for $\varepsilon = 0.05$

Figure 5.3: Plots (a), (b), and (c) illustrate the trajectory of the state $y(t)$ of the system for a range of ε values, whereas plots (d), (e), and (f) show the control law $v(t)$ for a range of ε values.

Chapter 6

Conclusion

For time-varying systems with two boundary conditions, we proposed utilizing reinforcement learning to create the best controller design. The time-varying issue is split into two simple time-invariant problems by the suggested approach, which takes use of the rapid time scale that exists at boundary conditions. In addition, we develop a learning-based control technique that is independent of system model knowledge. We demonstrate that when the issue time horizon gets longer, the precision of the controller performance gets better. Using an RL circuit, we produced simulated findings to back up our statements. We intend to continue our work on understanding nonlinear systems' controllers in the future.

Bibliography

[1] Mark Ardema. Nonlinear singularly perturbed optimal control problems with singular arcs. *Automatica*, 16(1):99–104, 1980.

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

[3] JH Chow. A class of singularly perturbed, nonlinear, fixed-endpoint control problems. *Journal of Optimization Theory and Applications*, 29(2):231–251, 1979.

[4] Justin Fong, Ying Tan, Vincent Crocher, Denny Oetomo, and Iven Mareels. Dual-loop iterative optimal control for the finite horizon lqr problem with unknown dynamics. *Systems & Control Letters*, 111:49–57, 2018.

[5] Martin Gugat, Emmanuel Trélat, and Enrique Zuazua. Optimal neumann control for the 1d wave equation: Finite horizon, infinite horizon, boundary tracking terms and the turnpike property. *Systems & Control Letters*, 90:61–70, 2016.

[6] Petros Ioannou and Bariş Fidan. *Adaptive control tutorial*. SIAM, 2006.

[7] Yu Jiang and Zhong-Ping Jiang. *Robust adaptive dynamic programming*. John Wiley & Sons, 2017.

[8] Bahare Kiumarsi, Frank L Lewis, Mohammad-Bagher Naghibi-Sistani, and Ali Karimpour. Optimal tracking control of unknown discrete-time linear systems using input-output measured data. *IEEE transactions on cybernetics*, 45(12):2770–2779, 2015.

[9] David Kleinman. On an iterative technique for riccati equation computations. *IEEE Transactions on Automatic Control*, 13(1):114–115, 1968.

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

[11] Petar V Kokotovic, Robert E O'Malley Jr, and Peddapullaiah Sannuti. Singular perturbations and order reduction in control theory—an overview. *Automatica*, 12(2):123–132, 1976.

[12] J Mahdavi, A Emaadi, MD Bellar, and M Ehsani. Analysis of power electronic converters using the generalized state-space averaging approach. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 44(8):767–770, 1997.

[13] Bo Pang and Zhong-Ping Jiang. Adaptive optimal control of linear periodic systems: An off-policy value iteration approach. *IEEE Transactions on Automatic Control*, 66 (2):888–894, 2020.

[14] Bo Pang, Tao Bian, and Zhong-Ping Jiang. Adaptive dynamic programming for finite-horizon optimal control of linear time-varying discrete-time systems. *Control theory and technology*, 17(1):73–84, 2019.

[15] Bo Pang, Zhong-Ping Jiang, and Iven Mareels. Reinforcement learning for adaptive optimal control of continuous-time linear periodic systems. *Automatica*, 118:109035, 2020.

[16] Richard S Sutton and Andrew G Barto. Reinforcement learning: an introduction mit press. *Cambridge, MA*, 22447, 1998.

[17] Michael Szmuk and Behcet Acikmese. Successive convexification for 6-dof mars rocket powered landing with free-final-time. In *2018 AIAA Guidance, Navigation, and Control Conference*, page 0617, 2018.

[18] 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.

[19] 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.

[20] Kyriakos G Vamvoudakis, Draguna Vrabie, and Frank L Lewis. Online adaptive learning of optimal control solutions using integral reinforcement learning. In *2011 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pages 250–257. IEEE, 2011.

[21] Xinning Wang, Chong Li, Dalei Song, and Robert Dean. A nonlinear circuit analysis technique for time-variant inductor systems. *Sensors*, 19(10):2321, 2019.

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

[23] 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.