EXPLORIZED FILTER-BASED ON-POLICY ADAPTIVE OPTIMAL LQR WITH UNKNOWN DYNAMICS

by

JAYANT CHHILLAR

Under the Supervision of

Dr Sayan Basu Roy

Indraprastha Institute of Information Technology Delhi

AUGUST 2023

EXPLORIZED FILTER-BASED ON-POLICY ADAPTIVE OPTIMAL LQR WITH UNKNOWN DYNAMICS

by

JAYANT CHHILLAR

Submitted

in partial fulfillment of the requirements for the degree of Master of Technology

to

Indraprastha Institute of Information Technology Delhi

AUGUST 2023

Certificate

This is to certify that the thesis entitled **"Explorized Filter-Based On-Policy Adaptive Optimal LQR with Unknown Dynamics"** being submitted by **Jayant Chhillar** of the Indraprastha Institute of Information Technology Delhi, for the award of the Master of Technology, is an original research work carried out by him under my supervision. In my opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree.

The results contained in this thesis have not been submitted in part or in full to any other University or Institute for the award of any degree or diploma to the best of our knowledge.

AUGUST 2023

Dr Sayan Basu Roy
Department of Electronics and Communications,
Indraprastha Institute of Information Technology Delhi.
New Delhi, 110020

Acknowledgements

First and foremost, I wish to express my deepest gratitude to Dr Sayan Basu Roy. His unwavering support, kindness, and mentorship have been invaluable in my academic journey. The wisdom he imparted and the faith he placed in me has been nothing short of transformative. Dr Roy, I am profoundly grateful for everything and aspire to make you proud of the scholar you have helped shape.

To my family, words cannot capture the depth of my appreciation. Your unconditional love, understanding, and encouragement have been my guiding light, allowing me to take risks, explore uncharted territories, and grow both academically and personally. My parents, the pillars of strength, have been my anchor, providing unwavering support in every step of this journey. Thank you for your understanding and for being there in ways more than one.

Lastly, to my friends, while it is impossible to list every name of those I would wish to convey my appreciation to; please know that every one of you holds a special place in my heart. Your camaraderie, encouragement, and faith in me have been invaluable. I am immensely grateful to all of you and promise to stand by your side just as you have stood by mine.

Abstract

Conventionally, optimal control solutions demand offline a priori knowledge of the system dynamics. Adaptive optimal control methods have been advantageous in allowing for dynamic controllers to approximate optimal control solutions by estimating system parameters online. For continuous-time linear quadratic regulator (LQR), the optimal control is inferred from the non-linear algebraic Riccati equation (ARE). This thesis presents a history of previous research on optimal and adaptive optimal control (AOC). Then we build on a policy iteration algorithm for solving online, on-policy adaptive solutions to the LQR problem. The proposed algorithm is a filter-based explorized approach to designing an adaptive optimal controller (AOC) for systems with unknown dynamics, where a two-layer low pass filter architecture is introduced. The initial layer mitigates the necessity for sensing state derivatives and using computationally complex finite window integrals (FWI), and the subsequent layer offers appropriate algebraic connections, negating the need for "intelligent" data storage techniques. An exploration signal is utilized in the control parameter to guarantee stability and convergence. We conclude by providing analytical assurances for the stability of the closed-loop dynamics of the system and presenting the simulation setup for the same.

Contents

List of Figures

Abbreviations

ADP	**A**pproximate **D**ynamic **P**rogramming
AOC	**A**daptive **O**ptimal **C**ontrol
ARE	**A**lgebraic **R**iccati **E**quation
CT	**C**ontinuous **T**ime
DP	**D**ynamic **P**rogramming
FBEPI	**F**ilter **B**ased **E**xplorized **P**olicy **I**teration
FWI	**F**inite **W**indow **I**ntegral
HJB	**H**amilton **J**acobi **B**ellman
IE	**I**nital **E**xcitation
LQR	**L**inear **Q**uadratic **R**egulator
LTI	**L**inear **T**ime **I**nvariant
PI	**P**olicy **I**teration
PE	**P**ersistence of **E**xcitation
RL	**R**einforcement **L**earning
UUB	**U**niformly **U**ltimately **B**ounded
VI	**V**alue **I**teration

Symbols

$\forall$	For All
$\exists$	There Exists
$\mathbb{R}$	Real Space
$\mathbb{R}^n$	Real Vector Space of dimension n
$\mathbb{R}^{n\times m}$	Real Matrix Space of dimension $n \times m$
$\mathbb{W}$	Set of whole numbers
I_n	Identity matrix of the order n
$\lambda_{min}(.)$	minimum eigenvalue of the argument matrix
$\lambda_{max}(.)$	maximum eigenvalue of the argument matrix

Chapter 1

Introduction

Control is an integral element of our lives, where control of any dynamic system means influencing the behaviour of the system, often with the goal of ensuring conditions of stability and optimality. Stability ensures the control system's reliability, safety, and predictability; whereas optimality in control systems is concerned with achieving efficiency while meeting specific criteria or objectives, it goes beyond stability and focuses on minimizing costs, maximizing outputs, or achieving desired goals in the most efficient way. This practice of seeking the "best" policy (known as a cost function) for a chosen objective of the performance criteria is studied under the heading of optimal control [1].

It is well known that the value function obtained as the solution to the non-linear partial differential equation, the Hamilton-Jacobi-Bellman (HJB) equation, allows us to derive a control strategy that is necessary for optimality [2]. For a linear system, the matrix Algebraic Riccati Equation (ARE) is considered as a particular case of the HBJ equation, where the value function of the HJB has a quadratic form [3]. One of the most commonly employed linear time-invariant (LTI) controllers, first introduced by R.E. Kalman, the Linear Quadratic Regulator (LQR) is a method to

design optimal feedback controls; where the optimal state-feedback gain of the LQR controller is derived from the infinite-horizon solution of the ARE [3].

Optimal controllers obtained through the HJB equation, and in extension, the ARE, require prior knowledge about the system dynamics and are typically computed in an offline manner. Such limitations have rendered these methods inaccessible to a wide variety of real-world control problems where exact knowledge of the system parameters is not available.

Adaptive control is a strategy developed to work online with systems that have uncertain, time-varying dynamics; adaptive controllers use real-time measurements to learn online control of systems with unknown parameters [4, 5]. Even though the goal of adaptive controllers is to minimize the output error of the system, adaptive controllers are not classically optimal as they do not aim to minimize any user-defined cost function [6].

To take advantage of the benefits that optimal and adaptive control strategies provide, a new branch of controllers, namely adaptive optimal controllers (AOC), was proposed. These methods aim to design a control system that adapts to uncertainties or changes in the system's dynamics and optimizes a specified performance criterion. The control system dynamically updates its control law based on observed system behaviour to adapt to changes in the system and optimize the specified performance index [6, 7, 8].

Reinforcement Learning (RL) has emerged as a promising approach to AOC [9]. At its core, RL seeks to learn optimal policies by interacting with an environment and receiving feedback in the form of rewards or penalties [10]. Unlike traditional control methods, RL does not require a known model of the environment, making it particularly suited for systems with uncertain or complex dynamics.

This thesis aims to tackle the problem of on-policy online (adaptive) optimal control of continuous-time (CT) LTI systems with completely unknown system parameters by approximating the LQR gains.

1.1 Literature Survey and Motivation

There has been considerable research in solving the HJB (or the ARE) numerically in an offline setting where all the methods provide sound, numerical advantage and have been proved to converge to the optimal solution of the ARE. These approaches operate on either the Hamiltonian matrix of the ARE [11, 12, 13, 14, 15] or they work on solving the Lyapunov equations (Newton's method) [16, 17, 18]. However, numerical solutions require system identification as a necessary precondition. Furthermore, an exact description of model dynamics is impossible to be learnt due to the presence of real-world non-linear dynamics. Hence the solutions obtained are only approximations of the actual process.

Adaptive controllers are well-versed in solving the tracking problem for models without a detailed and accurate system description. Techniques such as adaptive inverse optimization methods have been developed for non-linear systems [19, 20]. However, they have been largely unsuccessful in wide adoption as they attempt optimal control without directly solving the core equations (HJB / ARE) of optimal control, restricting the choice of the desired cost function. Moreover, these methods require a priori understanding of the stabilizing control laws for the system.

RL techniques have been extensively researched for the scope of adaptive optimal control [9, 21, 22, 23]. The iterative (recursive) approach of approximate dynamic programming (ADP) in RL allows for learning the solution to the optimal control

equation by online approximation of the cost function and the control policy based on the system state [24, 25]. Many popular ADP algorithms (e.g. policy iteration (PI,) value iteration (VI,) and Q-Learning) have been adopted to solve the optimal control problem [26, 27, 28, 29, 30].

Initial research in the domain was focused on the discrete-time problems [28, 30, 31, 32]. [22] first presented a solution to continuous-time discrete-state systems by exploiting the discrete-time methods for the case when the sampling time tends to zero. However, solutions for continuous-time continuous-state systems are more challenging to prove to be stable or converging. An initial model-based solution for the problem in CT systems is proposed in [33], [26] showcased two model-free RL PI algorithms which required measurement of the state derivative.

Policy Iteration algorithms first originated in the domain of stochastic decision theory [34] and have been extensively applied in solving optimal control for Markov decision problems [35]. PI algorithms guarantee convergence to optima for continuous-time LQRs [18]. As a significant contribution to the field, [27] suggested an online AOC for a continuous-time linear system with partial unknown system dynamics that employed a generalized PI algorithm. [27] further motivated the design of control for completely unknown systems in [36, 37].

Most of work done in the field demands high memory requirements as they work with finite window integrals (FWIs) [38, 27, 37, 36, 25] of functions of states;[1] the FWIs need data for the interval $[t-\delta, t]$ (t and δ are time instant and time window length respectively) which builds up the storage complications or they employ "intelligent" mode of storage that saves state data points along the system trajectory to be able to satisfy a full-rank condition guaranteeing optimal convergence [39, 40, 41]. [2]

[1] Refer to section 2.8.2
[2] Refer to section 2.8.3

The work done in [42, 43] presents a memory-efficient alternative to the existing methods; in [43], the authors further introduce an exploration signal to the control input to satisfy the excitation assumption on the regressor, however both work for a partially unknown system.

1.2 Contribution

The contributions made to the field of adaptive optimal control, particularly in the context of the infinite-horizon LQR problem, are as follows:

1. Development of an On-Policy Online Memory-Efficient Policy Iteration Algorithm for Approximation of Unknown System Dynamics: A policy iteration algorithm that stands out due to its memory efficiency is adapted to cater to systems with unknown dynamics. Traditional methods often demand high memory requirements, especially when dealing with finite window integrals of functions of states. The proposed algorithm tactically eliminates such demands, making it more feasible for real-time applications. The algorithm is designed to work online and on-policy. It is a significant advancement, given that it can approximate an optimal solution to the LQR problem even when complete system dynamics are unknown.

2. Comprehensive Background and Preliminaries: To ensure that the reader has a solid foundation for understanding the problem, this thesis provides a thorough background; this includes the necessary mathematical preliminaries, historical context, and a review of existing methods in the domain.

3. Simulink Model for Real-Time Continuous Systems: A practical contribution of this work is the development of a Simulink model tailored for continuous

systems. The model is a tangible representation of the theoretical constructs discussed in the thesis. The performance of the algorithm is also presented, emphasizing its practical applicability.

In summary, this thesis not only advances the theoretical understanding of AOC for the LQR problem but also offers practical tools and demonstrations that accentuate the real-world applicability of the proposed method.

1.3 Thesis Organization

This thesis is structured to provide a comprehensive understanding of the topics discussed, organized into distinct chapters as follows:

- Chapter 1: Introduction - This chapter sets the stage by introducing the research topic. It encompasses a literature survey and motivation detailing the existing work in the field and the gaps that this thesis aims to address. The contributions of this research are also highlighted.

- Chapter 2: Background and Preliminaries - Serving as the foundation, this chapter delves into the essential concepts and theories that underpin the research. The contents of this chapter should be sufficient for learning about the optimal control problem and its solution through the HJB. The chapter is written with a bottom-up approach, providing the necessary information to lead to the infinite-horizon LQR problem and its solution through the ARE.

- Chapter 3: Filter-Based Explorized Policy Iteration Algorithm - At the core of the thesis, this chapter presents the filter-based explored policy iteration algorithm. It includes the derivation of the algorithm, a proof of concept,

and an in-depth stability analysis. Both the system's general and closed-loop stability under the proposed algorithm are discussed.

- Chapter 4: Experimental Setup and Results - This chapter showcases the practical application of the proposed algorithm. It describes the experimental setup and presents the results witnessed from the experiments, highlighting the efficacy of the algorithm.

- Chapter 5: Conclusion - The concluding chapter summarizes the research findings, emphasizing the contributions and implications of the study.

The organization ensures a logical flow, starting from foundational concepts and building up to the primary research contributions, followed by practical applications and conclusions.

Chapter 2

Background and Preliminaries

We begin by considering the general state space representation used to describe any dynamic system:

$$\begin{aligned} \dot{x} &= f(x, u, t), \\ y &= g(x, u, t), \end{aligned} \tag{2.1}$$

where $x \in \mathbb{R}^n$ is the state variable, $u \in \mathbb{R}^m$ is the input variable and $y \in \mathbb{R}^p$ is the output variable; $\dot{x} = f(x, u, t)$ are the state equations, and $y = g(x, u, t)$ are the output equations and $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^n$ and $g : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^p$ are non-linear functions.

For a time-invariant system, the functions f and g are not dependent on t:

$$\begin{aligned} \dot{x} &= f(x, u), \\ y &= g(x, u), \end{aligned} \tag{2.2}$$

Further, if the system is also linear, f and g will be linear as well, and the state space representation can be written as:

$$\begin{aligned} \dot{x} &= Ax + Bu, \\ y &= Cx + Du, \end{aligned} \tag{2.3}$$

where $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}, C \in \mathbb{R}^{p\times n}, D \in \mathbb{R}^{p\times m}$ are constant state, input, output, and feedforward matrices that define the system dynamics.

2.1 Stability, Controlability, and Observability

Stability, controllability, and observability are three core pillars of control theory. They ensure that a system behaves as desired, can be driven to any state, and its internal dynamics can be inferred from its outputs.

2.1.1 Stability

Stability is a cornerstone of control theory, ensuring a system's behaviour remains predictable and bounded over time, irrespective of initial conditions or external disturbance. The study of a system's stability is crucial because for a system to be useful, it must exhibit predictable behaviour.

Intuitively, stability suggests that in the absence of an external input, a system's response will converge to (and remain at) some equilibrium point. Consider a dynamic system with no input: $\dot{x} = f(x,t)$ (where the initial state of the system is: $x(t_{o)} = x_o,$) for the system to be stable it must converge to an equilibrium point x^* such that $f(x^*,t) = 0 \ \forall \ t \geq t_o$.

A system is asymptotically stable if, for every initial state $x(0)$, $x(t) \to 0$ as $t \to \infty$.

Understanding the stability of a system is a difficult task. The most commonly utilized stability analysis method is the Lyapunov stability theorem which says that for a system with an equilibrium point $x_o = 0$, we can devise an energy function $V(x)$. $V(x)$ should be positive definite (i.e. $V(0) = 0$ and $V(x) \geq 0 \; \forall \; x \neq 0$). Then if the system's dynamics ensure that its energy consistently decreases over time, eventually reaching zero, the system's trajectories can only converge to the origin. As a result, the system displays asymptotic stability.

Definition 1. *The energy function $V(x)$ is called the Lyapunov function, and the stability of the system can be learnt from its time derivative $\dot{V}(x)$:*

- *If $\dot{V}(x) \leq 0 \; \forall \; x$, the system is said to show Lyapunov stability; this ensures that trajectories starting close to the equilibrium remain close to it, but they might not necessarily converge to the equilibrium.*
- *And if $\dot{V}(x) < 0 \; \forall \; x \neq 0$ and $\dot{V}(x) = 0$ for $x = x_o$, then the system is asymptotically stable for the equilibrium point and the system trajectories not only remain close to the equilibrium but also converge to it over time.*

Lyapunov stability analysis is instrumental as it only requires clever construction of the Lyapunov function, and one can infer stability properties without explicitly solving the system's equations. However, finding an appropriate Lyapunov function for a given system can be challenging, and for some systems, it might not exist at all.

However, for an LTI system $\dot{x} = Ax$, asymptotic stability can be confirmed by assessing the eigenvalues of the state matrix A. The system is asymptotically stable if A has eigenvalues where all real parts are negative.

2.1.2 Controllabilty

Controllability is the ability to drive a system from any initial state to any desired final state in a finite time using suitable control inputs. Controllability ensures that we can move the eigenvalues or poles of a system to any desirable locations to achieve stability or optimality by state feedback.

An LTI system is controllable if for all possible states x_o, and $\forall\ \Delta > 0$, there exists an input function $u(t)$ (where $t \in [0, \Delta]$) such that under this input, the state of the system moves from x_o at $t = 0$ to 0 at $t = \Delta$ [44].

We can infer the controllability of an LTI system through the controllability matrix $\mathcal{C}$:

$$\mathcal{C} = \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix}.$$

If $rank(\mathcal{C}) = n$ (where n is the dimension of the state variable,) then the system is said to be controllable; for the system to be controllable, the matrix $\mathcal{C}$ needs to have n linearly independent rows so that each of the n states is reachable by the input u.

2.1.2.1 Stabilizability

Stabilizability is the characteristic of a dynamic system that suggests that some possible control input can lead the system to stable dynamics, even if the state variables involve some uncontrollable elements.

An LTI system (eq. 2.3) is said to be stabilizable if a feedback matrix K exists such that $(A + BK)$ is asymptotically stable [45].

2.1.3 Observability

Observability ensures that we can estimate or reconstruct the state variables from the output, making state feedback feasible. A system is called observable if all its states can be observed and their values can be determined from the output.

An LTI system is said to be observable if the initial state $x(0) = x_0$ can be determined from the output measurements ($y(t)$) and the input to the system ($u(t)$) over a finite time interval $t \in [0, \Delta]$ (where $\Delta > 0$) [45].

Similar to the controllability matrix $\mathcal{C}$, the observability of a system can be determined using the observability matrix $\mathcal{O}$:

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}.$$

If $rank(\mathcal{O}) = n$ (where n is the dimension of the state variable,) then the system is said to be observable. Here, the idea is that if all n columns are linearly independent, all the n states can be observed as linear combinations of the output variables y.

2.1.3.1 Dectectability

Detectability is the dual of stabilizability, where a system is said to be detectable if its unobservable states remain stable (do not lead to instability). The LTI system (eq. 2.3) is detectable if the pair (C, A) are detectable, which would only be possible iff (A^T, C^T) is stabilizable [45].

2.2 The optimal control problem

Optimal control theory seeks to determine the control input $u(t)$ that minimizes a certain performance criterion, typically described by a cost functional J. The performance index can have multiple forms, as follows [46]:

2.2.1 Performance index for a time-optimal control system

Here, the controller's goal is to minimize the time it takes for the system to go from an initial state $x(t_o)$ to the specified final state $x(t_f)$. The performance index for the same is:

$$J = \int_{t_o}^{t_f} dt = t_f - t_o = t^*,$$

2.2.2 Performance index for a fuel-optimal control system

Consider a case where the controller's goal is to minimize the total fuel consumption. If $u(t)$ is the input applied to a system for control, then we can assume that the magnitude $\|u(t)\|$ is proportional to the fuel the system needs. The performance index here will become:

$$J = \int_{t_o}^{t_f} \|u(t)\| dt.$$

For multiple controls, the J function can be re-written as:

$$J = \int_{t_o}^{t_f} \sum_{i=1}^{m} R_i \|u_i(t)\| dt,$$

where R is a weight matrix on the control input.

2.2.3 Performance index for a minimum-energy control system

Similar to the fuel-optimal problem, the rate of energy consumed by the system is proportional to the input of the system and the performance criteria for the controller is provided by:

$$J = \int_{t_o}^{t_f} u^T(t) R u(t) dt,$$

where $R \in \mathbb{R}^{m \times m}$ is a symmetric positive definite weight matrix for the control input and R can be chosen to be significant to have stricter conditions on energy expenditure.

2.2.4 Performance index for minimum tracking error control system

If the objective of the system is to keep a bounded performance and minimize the tracking error, we can use the integral of the squared tracking error as the performance index:

$$J = \int_{t_o}^{t_f} \left(x(t) - x_d(t)\right)^T Q \left(x(t) - x_d(t)\right) dt$$

$$= \int_{t_o}^{t_f} x_e^T(t) Q x_e(t) dt,$$

where $x(t)$ is the system state, $x_d(t)$ is the desired state, and $Q \in \mathbb{R}^{n \times n}$ is a symmetric, positive semi-definite weight matrix on the state variables in x. We choose Q to be significant if we want to accomplish precise tracking.

2.2.5 Performance index for terminal control system

The terminal cost function depends on the error between the desired target position $x_d(t_f)$ and the actual target position $x(t_f)$ at the final time t_f. The cost function here is formulated by using the target error $x_e(t_f) = x(t_f) - x_d(t_f)$:

$$J = x_e^T(t_f) F x_e(t_f),$$

where $F \in \mathbb{R}^{n \times n}$ is also a symmetric, positive semi-definite weight matrix capable of defining the importance of reaching the desired final state.

2.2.6 Performance index for general optimal control system:

Combining the different interpretations, we have a general form of the optimal control performance index:

$$J = x_e^T(t_f) F x_e(t_f) + \int_{t_o}^{t_f} \left[x_e^T(t) Q x_e(t) + u^T(t) R u(t) \right] dt. \tag{2.4}$$

Rewriting the general performance index for a nonlinear system:

$$J = g(x_e(t_f), t_f) + \int_{t_o}^{t_f} V(x_e(t), u(t), t) dt. \tag{2.5}$$

Now we can formally define that the general optimal control problem aims to find an optimal input $u^*(t)$ for $t \in [t_o, t_f]$, such that:

$$u^*(t) = \arg\min_{u(t)} \left[g(x_e(t_f), t_f) + \int_{t_o}^{t_f} V(x_e(t), u(t), t) dt \right]. \tag{2.6}$$

2.3 Dynamic Programming and Optimality Principle

Dynamic programming (DP) is a potent mathematical optimization technique introduced by R. E. Bellman in the 1950s. The DP approach decomposes a multifaceted problem into simpler yet identical subproblems, solving each once recursively and storing their solutions to prevent redundant computations [47]. This method is invaluable in control theory, where the objective often involves determining the optimal control strategy over time.

The Bellman equation captures the essence of dynamic programming in control theory, a recursive relationship that relates the value of an optimal solution to the values of its subproblems [23]. For a discrete-time, finite-horizon control problem, the Bellman equation is:

$$V_t(x) = \min_{u \in U} \{c(x,u) + \beta V_{t+1}(f(x,u))\},$$

where $V_t(x)$ is the value function at time t given state x, $c(x,u)$ is the immediate cost of choosing control u in state x, $f(x,u)$ is the state transition function and β is the discount factor.

The optimality principle, a cornerstone of DP, asserts that an optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision [23]. I.E. if $u(t) = u^*(t)$ for $t \in [t_o, t_f]$ is an optimal solution, then $u(t)$ is also optimal over the interval $[t_o + \Delta t, t_f]$ where $t_o \leq t_o + \Delta t \leq t_f$. This principle ensures that each subproblem is solved optimally, leading to an overall optimal solution.

2.4 The Hamltion-Jacobi-Bellman equation

The HJB is a nonlinear partial differential equation (PDE) that characterizes the value function, which represents the optimal cost-to-go from a given state, as the solution to this PDE. It provides a necessary and sufficient condition for optimality of a control over a given time horizon [2].

Let us begin by considering $J^*(x(t), t)$ as the optimal cost in the interval $[t, t_f]$ and initial state $x(t)$:

$$J^*(x(t), t) = \min_{\substack{u(t) \\ \tau \in [t, t_f]}} \left[g(x_e(t_f), t_f) + \int_{t_o}^{t_f} V(x_e(t), u(t), t) dt \right].$$

By applying the principle of optimality,[1] we can say:

$$J^*(x(t), t) = \min_{\substack{u(t) \\ \tau \in [t, t+\Delta t]}} \left[J^*(x(t + \Delta t), t + \Delta t) + \int_{t_o}^{t+\Delta t} V(x_e(t), u(t), t) dt \right].$$

Considering Δt is very small in the above equation, we can approximate the integral $\int_{t_o}^{t+\Delta t} V(x_e(t), u(t), t) dt$ as $V(x_e(t), u(t), t)\Delta t$ and $J^*(x(t + \Delta t), t + \Delta t)$ can be approximated by its Taylor series expansion. We thus obtain:

$$J^*(x(t), t) = \min_{u(t)} [V(x_e(t), u(t), t)\Delta t + J^*(x(t), t) + \frac{\partial J^*(x(t), t)}{\partial t} \Delta t + \left[\frac{\partial J^*(x(t), t)}{\partial x} \right]^T \Delta x + \mathcal{O}(x, t)],$$

where $\Delta x = x(t + \Delta t) - x(t)$ and $\mathcal{O}(x, t)$ represents the higher order terms in the Taylor expansion (which can be ignored).

[1] Refer to section 2.3

Since, $J^*(x(t),t)$ and $\frac{\partial J^*(x(t),t)}{\partial t}\Delta t$ don't depend on $u(t)$, the above equation can be written as:

$$J^*(x(t),t) = J^*(x(t),t) + \frac{\partial J^*(x(t),t)}{\partial t}\Delta t + \min_{u(t)}\left[V(x_e(t),u(t),t)\Delta t + \left[\frac{\partial J^*(x(t),t)}{\partial x}\right]^T \Delta x\right],$$

$\therefore$

$$-\frac{\partial J^*(x(t),t)}{\partial t} = \min_{u(t)}\left[V(x_e(t),u(t),t) + \left[\frac{\partial J^*(x(t),t)}{\partial x}\right]^T \frac{\Delta x}{\Delta t}\right].$$

Now, if we let $\Delta t \to 0$,then $\frac{\Delta x}{\Delta t} \to \dot{x} = f(x(t),u(t),t)$ which yields the Hamilton-Jacobi-Bellman equation:

$$-\frac{\partial J^*(x(t),t)}{\partial t} = \min_{u(t)}\left[V(x_e(t),u(t),t) + \left[\frac{\partial J^*(x(t),t)}{\partial x}\right]^T f(x(t),u(t),t)\right]. \quad (2.7)$$

The optimal input $u(t)$ which staisfies (2.7) can now then be symbolically expressed as:

$$u^*(t) = \mathcal{U}\left(\frac{\partial J^*(x(t),t)}{\partial x}, x(t), t\right),$$

which, when substituted back to the HJB equation (2.7), leads to the optimal cost equation:

$$-\frac{\partial J^*(x(t),t)}{\partial t} = V(x_e(t),\mathcal{U},t) + \left[\frac{\partial J^*(x(t),t)}{\partial x}\right]^T f(x(t),\mathcal{U},t),$$

and the solution for this equation must satisfy the boundary condition:

$$J^*(x(t_f),t_f) = g(x(t_f)).$$

2.5 Linear Quadratic Regulator & the Riccati equation

The linear quadratic regulator problem takes into consideration an LTI system (2.3) as described before with an initial state $x(t_o) = x_o$. The goal of the LQR is to drive the initial state x_o to the smallest possible value, as soon as possible, in the interval $[t_o, t_f]$, without spending too much control effort. The cost function for this goal is defined as:

$$J = x^T(t_f)Fx(t_f) + \int_{t_o}^{t_f} \left[x^T(t)Qx(t) + u^T(t)Ru(t)\right] dt.$$

The HJB equation (2.7) then becomes:

$$-\frac{\partial J^*(x(t),t)}{\partial t} = \min_{u(t)} \left[x^TQx + u^TRu + \left[\frac{\partial J^*(x(t),t)}{\partial x}\right]^T (Ax + Bu) \right], \tag{2.8}$$

where,

$$J^*(x(t_f), t_f) = x^TFx.$$

Let us now consider a solution of the form $J^*(x(t),t) = x^TP^*(t)x$ where $P^*(t) \in \mathbb{R}^{n\times n}$ is a symmetric matrix. The time derivative of the optimal cost function then equals $x^T\dot{P}(t)x$ and the state derivative becomes $2P^*(t)x$ by substituting these in (2.8) we get

$$-x^T\dot{P}^*(t)x = \min_{u(t)} \left[x^TQx + u^TRu + 2x^TP(t)Ax + 2x^TP(t)Bu\right]. \tag{2.9}$$

To obtain the minimal u $(= u^*)$, we have to set the gradient of this equation with respect to u to 0, and that gives us:

$$2B^T P^*(t)x + 2Ru^* = 0,$$

or

$$u^* = -R^{-1}B^T P^*(t)x = -K^* x, \tag{2.10}$$

where, $K^* \in \mathbb{R}^{m \times n}$ is known as the optimal control gain martix.

Substituting this minimal value of u back in (2.9), we get

$$-x^T \dot{P}^*(t)x = x^T(P^*(t)A + A^T P^*(t) - P^*(t)BR^{-1}B^T P^*(t) + Q)x.$$

Therefore, for $J^*(x(t_f), t_f) = x^T P^*(t)x$ to be a valid solution, $P^*(t)$ (called the Riccati matrix) must satisfy the following matrix differential equation (also known as the continuous-time dynamic Riccati equation):

$$\dot{P}^*(t) = P^*(t)A + A^T P^*(t) - P^*(t)BR^{-1}B^T P^*(t) + Q,$$

with the boundary condition that $P^*(t_f) = F$.

If we modify the LQR problem to have an infinite horizon, the cost function becomes:

$$J = \int_{t_o}^{\infty} \left[x^T(t)Qx(t) + u^T(t)Ru(t)\right] dt.$$

Here, the matrix $P^*(t) \to P^*$ becomes a constant matrix $\therefore \dot{P}^*(t) = 0$, and the continuous-time dynamic Riccati equation reduces to the algebraic Riccati equation:

$$P^*A + A^T P^* + Q - P^*BR^{-1}B^T P^* = 0. \tag{2.11}$$

Moreover, the condition imposed on the system is to be stabilizable. If the system is not stabilizable, as $x(t) \not\to 0$

$$J = \int_{t_o}^{\infty} \left[x^T(t)Qx(t) + u^T(t)Ru(t)\right] dt \to \infty.$$

Hence optimal control does not exist.

2.6 Policy Iteration

PI is a fundamental iterative method in dynamic programming and reinforcement learning. The core idea behind PI is to iteratively refine a policy until convergence to an optimal policy is achieved.

The PI algorithm comprises two fundamental, sequential steps:

1. Policy Evaluation: this is the step where the current acting policy π (starting with the initial policy) is evaluated for its effectiveness.

2. Policy Iteration: is an iterative method to refine and improve a policy based on its evaluated value. The goal is to find a policy π' that optimizes the system's performance, such as minimizing costs or maximizing rewards.

The algorithm terminates when $\pi = \pi'$, i.e. the old and new policies are identical (or sufficiently close), guaranteeing that the final policy is optimal.

PI is guaranteed to converge as Bellman's principle of optimality ensures that if a policy π' derived during the policy improvement step is identical to π, then π is an optimal policy; convergence is a result of finite states and action spaces, ensuring that there are only a finite number of policies to consider.

PI's advantage of an explicit control policy at each iteration allows for a real-time implementation of the algorithm.

2.7 Kleinman's Algorithm

The ARE provides us with an algebraic solution to the optimal control of an LQR problem. However, solving the ARE quickly becomes complex due to the nonlinearity of the Riccati matrix, P, as the dimensions of the system grow. Kleinman's algorithm [18] offers a numerical algorithm to approximate the Riccati matrix iteratively.

Theorem 2.1. *Let $A_k = A - BK_k$ where $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are the state and control matrices of an LTI system, $\forall k \in \mathbb{W}$,[2] and $K_k = R^{-1}B^T P_k$ [3]. The ARE (2.11) then becomes:*

$$P_k A_k + A_K^T P_k + Q + K_k^T R K_k = 0. \tag{2.12}$$

Now if we iteratively update the gain matrix K_k $\forall k \in \mathbb{W}$ by:

$$K_{k+1} = R^{-1}B^T P_k. \tag{2.13}$$

Then for a stable K_0, the following holds true:

1. $0 < P^* \leq P_{k+1} \leq P_k$
2. $lim_{k\to\infty} P_k = P^*$ *and* $lim_{k\to\infty} K_k = K^*$

The proof of convergence can be found in [18].

[2] $\mathbb{W}$ represents the set of whole numbers
[3] See equation 2.10

2.8 Other topics of interest

2.8.1 Excitation

In control systems and signal processing, excitation refers to an external input or stimulus applied to a system; this input is crucial as it drives the system's response to be able to extract valuable information about the system's characteristics. Excitation is especially meaningful for adaptive systems because it ensures that the system is sufficiently stimulated, allowing the adaptive algorithm to learn or estimate the system's parameters accurately.

Definition 2. *Consider a bounded signal* $\psi(t) \in \mathbb{R}^n$, *where* $t \in [t_o, \infty]$ $\forall t_o \geq 0$. *We call* $\psi(t)$ *exciting over an interval* $[t_1, t_1 + T]$, $T > 0$ *and* $t_1 \geq t_o$, *if* $\exists\ \gamma > 0$, *such that the following inequality holds:*

$$\int_{t_1}^{t_1+T} \psi(\tau)\psi(\tau)^T d\tau \geq \gamma I_n$$

2.8.1.1 Persistance of Excitation

The Persistence of Excitation (PE) condition ensures that the input signal contains enough information about the system, preventing the adaptive algorithm from getting stuck in undesirable local minima and ensuring convergence to the true system parameters. In adaptive control, PE is crucial for the stability and convergence of adaptive algorithms [48, 49]. Without a persistently exciting signal, the adaptive controller might not gather enough information to accurately model the system, leading to suboptimal or unstable control.

Definition 3. *A bounded regressor* $\varphi(t) \in \mathbb{R}^n$ *is presistently exciting (PE), if* $\exists$ $\varepsilon, \gamma > 0$ *such that*

$$\int_t^{t+\varepsilon} \varphi(\tau)\varphi^T(\tau)d\tau \geq \gamma I_n, \qquad \forall t$$

where ε *is the window length.*

The restrictive condition of PE is widely apparent across many works in AOC [38, 27, 50, 36, 39], employed to obtain the convergence of the estimated parameters to the optimal parameters P^* and K^*. However, even though PE is a theoretically elegant condition for parameter convergence, it is infeasible in practical applications and online implementation due to its dependence on the future values of the regressor signal.

2.8.2 Finite Window Integrals

Finite window integrals (FWIs) refer to the integration of certain functions of system states over a finite time window. In the case of AOC, usually, FWIs are used on an equation derived from the ARE to formulate a suitable linear regression association [27, 37, 36, 26].

Mathematically, for a function $f(t)$ and a time window of length γ, the FWI is given by:

$$\int_{t-\gamma}^{t} f(\tau)d\tau,$$

where t is the current time instant.

FWIs accumulate information about the system's behaviour over a recent time window. By integrating the system's responses over a finite time window, more accurate estimates of system parameters can be obtained. FWI ensures that the

controller does not make decisions based on transient or momentary behaviours but considers a more extended system response. Nonetheless, while FWIs provide valuable information, they also have a high computational cost. Ideally, FWIs require infinite data points in the time interval γ, which leads to high memory demands and very short windows might not provide enough data for accurate decision-making.

2.8.3 Concurrent Learning

Concurrent learning is a methodology that uses current data (obtained through parameter update laws) and past data to improve the learning process and guarantee parameter convergence [51, 52]. Unlike PE conditions, concurrent learning allows for online verification of parameter convergence based on a sufficiency condition associated with the rank of the matrix formed out of stored data. Moreover, unlike FWIs, concurrent learning can be more memory-efficient because it does not require continuous storage of past data; instead, specific "informative" data points are stored "intelligently" and used alongside real-time data.

Chapter 3

Filter-Based Explorized Policy Iteration Algorithm

This chapter presents a PI algorithm for a CT infinite-horizon LQR problem where the system matrices A, B are unknown. The first section begins by comparing methods proposed in past literature to the techniques employed in this algorithm. The second section offers a mathematical derivation of the algorithm, beginning from a classical CT LTI system to finally realize an algebraic relation that can be used to solve online the ARE of an infinite-horizon LQR problem. The third section provides all the necessary and sufficient conditions that guarantee proof of work from the algorithm; this section rationalizes and satisfies any assumptions made in the derivation of the solution. Finally, the fourth section concludes the chapter with the stability and convergence analysis of the algorithm.

3.1 Derivation

Consider the CT LTI system:

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{3.1}$$

where $x \in X \subseteq \mathbb{R}^n$, $u(t) \in U \subseteq \mathbb{R}^m$ are the state and control variables respectively; $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are the system and control matrices respectively with the pair (A, B) are assumed to be controllable. $t \in [t_0, \infty]$, $t \geq 0$. The infinite-horizon quadratic value function:

$$V(x(t)) = \int_t^\infty \left[x^T(t)Qx(t) + u^T(t)Ru(t)\right] dt, \tag{3.2}$$

where $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$ are weight matrices on the state and control values, respectively. Q is symmetric positive semi-definite, and R is symmetric positive definite, and $(A, Q^{\frac{1}{2}})$ is assumed to be detectable, is then minimized by the optimal control input $u^*(t)$, given by:

$$u^*(t) = -K^*x(t),$$

where $K^* = R^{-1}B^TP^* \in \mathbb{R}^{m\times n}$ is the optimal control gain matrix and $P^* \in \mathbb{R}^{n\times n}$ is the constant symmetric positive definite matrix solution of the ARE:

$$P^*A + A^TP^* + Q - K^{*T}RK^* = 0.$$

Now, according to Kleinman's algorithm,[1] for any initially stable $K_{k=0}$ where $\forall k \in \mathbb{W}$, the gain matrix can be made to converge to the optimal value if it is updated

[1] Refer to section 2.7

iteratively as per the policy iteration step (2.13) for the evaluation of (2.12). The infinite-horizon value function for any stabilizing K_k can also be expressed as the quadratic parametric function of states as [53]:

$$V_k(x(t)) = x^T(t)P_k x(t). \tag{3.3}$$

In this proposed algorithm, we introduce an exciting exploration signal in the control input, now defined for any stabilizing K_k as:

$$u(t) = -K_k x(t) + \epsilon(t), \qquad \forall t \in [T_k, T_{k+1}), \tag{3.4}$$

where, T_k is the policy update time instance with $T_0 = 0$ and $\epsilon \in \mathbb{R}^m$ is an exploration signal, upper bounded as $\|\epsilon(t)\| \leq \epsilon_m > 0$.

Equating the RHS of (3.2) and (3.3) and differentiating w.r.t time along the system trajectory and the proposed control input (3.4), we get the following:

$$2x^T P_k \dot{x} = -x^T(Q + K_k^T R K_k)x + 2x^T P_k B\epsilon. \tag{3.5}$$

However, from equation (2.13)

$$\begin{aligned} K_{k+1} &= R^{-1}B^T P_k \\ \implies RK_{k+1} &= B^T P_k \\ \implies (RK_{k+1})^T &= (B^T P_k)^T \\ \implies K_{k+1}^T R^T &= P_k^T B^{T^T}, \end{aligned}$$

and since, R and P are symmetric, we have,

$$K_{k+1}^T R = P_k B. \tag{3.6}$$

Substituting (3.6) in (3.5),

$$2x^T P_k \dot{x} = -x^T (Q + K_k^T R K_k)x + 2x^T K_{k+1}^T R\epsilon, \tag{3.7}$$

or

$$2x^T P_k \dot{x} - 2x^T K_{k+1}^T R\epsilon = -x^T (Q + K_k^T R K_k)x.$$

The above equation can be rewritten as follows using the standard properties of the $vec(\cdot)$ and Kronecker product [2]:

$$2(\dot{x} \otimes x)^T vec(P_k) - 2(R\epsilon \otimes x)^T vec(K_{k+1}^T) = q_k, \tag{3.8}$$

where $q_k \triangleq -x^T (Q + K_k^T R K_k)x \in \mathbb{R}_0^-$. Equation (3.8) can be further simplified as

$$\begin{bmatrix} 2(\dot{x} \otimes x)^T & -2(R\epsilon \otimes x)^T \end{bmatrix} \begin{bmatrix} vec(P_k) \\ vec(K_{k+1}^T) \end{bmatrix} = q_k, \tag{3.9}$$

and since P_k is a symmetric matrix, (3.9) can have a reduced dimensional form:

$$\begin{bmatrix} \dot{g}^T & d^T \end{bmatrix} \phi_k = q_k, \tag{3.10}$$

where $g \in \mathbb{R}^r$, $d \in \mathbb{R}^{n.m}$ and $\phi_k \in \mathbb{R}^{r+n.m}$ (where $r \triangleq \frac{(n)(n+1)}{2}$) are defined as

$$g \triangleq vech(2xx^T - diag^2(x)),$$

[2] Refer to link for information on $vec(\cdot)$ and Kroneker product

$$d \triangleq -2(R\epsilon \otimes x),$$

$$\phi_k = \begin{bmatrix} vech(P_k) \\ vec(K_{k+1}^T) \end{bmatrix},$$

respectively. Here $vech(\cdot)$ operates on a symmetric matrix and generates a column vector by placing the consecutive columns of the argument matrix one after another while neglecting symmetric terms, and $diag(\cdot)$ generates a diagonal matrix from the elements of the argument vector.

Nevertheless, equation (3.10) cannot be utilized directly for solving P_k and K_{k+1} due to their dependence on $\dot{g}$, which in turn depends on $\dot{x}$. To obliviate the need to realize the state derivative, we employ a low-pass filter architecture that works as a state estimator; the filter equations are defined as

$$\dot{h_l} = -p_l h_l + \dot{g}, \qquad h_l(T_k) = 0 \tag{3.11}$$

$$\dot{q_{kl}} = -p_l q_{kl} + q_k, \qquad q_{kl}(T_k) = 0 \tag{3.12}$$

$$\dot{d_l} = -p_l d_l + d, \qquad d_l(T_k) = 0 \tag{3.13}$$

where $h_l \in \mathbb{R}^r$, $q_{kl} \in \mathbb{R}_0^-$ and $d_l \in \mathbb{R}^{n.m}$ are filter outputs for $\dot{g}$, q_k and d respectively, and p_l is a positive tunable gain stabilizing the filters. The explicit solutions to the equations (3.11), (3.12), and (3.13) is given by

$$h_l(t) = e^{-p_l t} \int_{T_k}^{t} e^{p_l \tau} \dot{g}(\tau) d\tau \tag{3.14}$$

$$q_{kl}(t) = e^{-p_l t} \int_{T_k}^{t} e^{p_l \tau} q_k(\tau) d\tau \tag{3.15}$$

$$d_l(t) = e^{-p_l t} \int_{T_k}^{t} e^{p_l \tau} d(\tau) d\tau \tag{3.16}$$

Substituting the value of q_k from (3.10) in (3.15) and using (3.14) and (3.16), the filterized form of equation (3.10) becomes

$$w_l^T \phi_k = q_{kl}, \tag{3.17}$$

where $w_l \triangleq \begin{bmatrix} h_l \\ d_l \end{bmatrix} \in \mathbb{R}^{r+n.m}$. However, here too, the need for $\dot{g}$ still persists if h_l is calculated by (3.11). Whereas, by applying the by-parts rule of integration on equation (3.14), we have,

$$h_l(t) = g(t) - e^{-p_l(t-T_k)} g(T_k) - p_l g_l, \tag{3.18}$$

where $g_l \in \mathbb{R}^r$ is evaluated using the filter equation

$$\dot{g}_l = -p_l g_l + g, \qquad g_l(T_k) = 0$$

Since g is measurable, then g_l is measurable; hence h_l can be computed online using (3.18).

Finally, equation (3.17) allows for a completely measurable algebraic relation for P_k and K_{k+1} with no dependence on the state derivatives; still, it must be noted that the equation (3.17) is in linear regression form.

There is still a need for "intelligent" data storage mechanisms to be able to satisfy a full rank condition of the information-rich past data matrix (multiple rows of w_l matrix in this case) to solve P_k and K_{k+1}; to discount the need for such measures,

the proposed algorithm exploits a second low-pass filter defined by the equations

$$\dot{W}_{ll} = -p_{ll} W_{ll} + w_l w_l^T, \qquad W_{ll}(T_k) = 0 \tag{3.19}$$

$$\dot{q}_{kll} = -p_{ll} q_{kll} + w_l q_{kl}, \qquad q_{kll}(T_k) = 0 \tag{3.20}$$

where $W_{ll} \in \mathbb{R}^{z \times z}$ and $q_{kll} \in \mathbb{R}^z$ (where $z = r + n.m$) are the second layer filter outputs and p_{ll} is the positive tunable gain for the second filter.

Integrating (3.19) and (3.20) and using (3.17) we can derive,

$$W_{ll} \phi_k = q_{kll} \tag{3.21}$$

Assumption 1. *Let us assume $W_{ll}(t)$ is invertible.*

If assumption (1) holds, online computation of ϕ_k can be done as follows:

$$\phi_k = W_{ll}^{-1}(T_{k+1}) q_{kll}(T_{k+1}) \tag{3.22}$$

where $T_{k+1} \geq T_k + L_k$, $L_k > 0$. From equation (3.22) we can extract both P_k and K_{k+1} by reversing the $vec(\cdot)$ and $vech(\cdot)$ operations.

3.2 Proof of Concept

Equation (3.19) can be solved as,

$$W_{ll}(t) = \underbrace{e^{-p_{ll} t}}_{\geq 0} \int_{T_k}^{t} \underbrace{e^{p_{ll} t}}_{\geq e^{p_{ll} T_k} \geq 1} \underbrace{w_l(\tau) w_l^T}_{\geq 0} \; d\tau,$$

from (3.2), we claim,

Property 1. *The matrix* $W_{ll}(t)$ *is a positive semi-definite, i. e.* $W_{ll}(t) \geq 0;\ \forall t \geq T_k$

Now, let's make another assumption,

Assumption 2. *The regressor* $w_l(t)$ *is exciting over a time interval* $[T_k, T_k + L_k]$, *where* $L_k,\ \alpha > 0,\ \forall k \in \mathbb{W}$. *Therefore, by definition (of excitation)*[3]*:*

$$\int_{T_k}^{T_k+L_k} w_l(\tau) w_l^T(\tau) d\tau \quad \geq \alpha I_{r+n.m}$$

where L_k *and* α *are time window length and degrees of excitation respectively.*

Remark 3.1. *The above assumption is made possible by the exploration signal we introduced in the control input, & provides sufficient information content to obtain the intermediate policies* $(K_k's)$. *However, it must also be noted that this assumption is weaker than the classical PE condition, due to the excitation being limited to discrete sets of time-windows. Our assumption is similar to a recently introduced convergence criteria [54] for asymptotic stability.*

To establish that Assumption 2 is sufficient for computing the cost (P_k) and the improved gain (K_{k+1}) we claim Lemma 3.2 and 3.3.

Lemma 3.2. $w_l(t)$ *is exciting in the interval* $[T_k, T_k+L_k]$ *iff* $W_{ll}(t)$ *is positive definite (PD) at* $t = T_k + L_k$,

Lemma 3.3. *If* $W_{ll}(T_k + L_k)$ *is PD, then* $W_{ll}(t)$ *will remain PD* $\forall t\ \in [T_k + L_k, t_f]$, *where* $T_k + L_k < t_f < \infty$.

The proof for Lemma 3.2- 3.3 are available in [55].

Now Property 1 claims $W_{ll}(t) \geq 0,\ \forall t \geq T_k$, hence, $det(W_{ll}(t)) = \prod_{j=1}^{r+n.m} \lambda_j(t) \geq 0$ where $\lambda_j(t) \geq 0,\ j = 1(1)(r + n.m)$, is the j^{th} eigenvalue of $W_{ll}(t)$. According to

[3]Refer to definition 2

Assumption 2, the value of $det(W_{ll}(t))$ will grow over time. For when $det(W_{ll}(t)) > 0$, the matrix W_{ll} will be PD, which, along with Lemma 3.2 and 3.3 affirm that Assumption 2 holds, and is verifiable online. Furthermore, positive definiteness of W_{ll} confirms that W_{ll} is invertible, i. e. Assumption 1 is also satisfied.

3.3 Stability Analysis

3.3.1 Stability and Convergence Analysis

Theorem 3.4. *Provided $Q > 0$ and control gain K_k is stabilizing, the improved policy obtained from (3.22) will lead to a stabilizing control policy $u(t) = -K_{k+1}x(t) + \epsilon(t)$ and the closed-loop system $\dot{x} = A_{k+1}x + B\epsilon$ will be uniformly ultimately bounded (UUB) $\forall k \in \mathbb{W}$.*

Proof: Considering $V_k(x(t))$ as defined in (3.3), as a positive definite Lypaunov function for the LTI system (3.1) with control policy $u(t) = K_{k+1}x(t) + \epsilon(t)$. Taking the time derivative of the Lyapunov function along (3.1), using (2.12) & (2.13), we get:

$$\dot{V}_k(x) = -x^T\{P_k(A - BK_{k+1}) + (A - BK_{k+1})^T P_k\}x + 2x^T P_k B\epsilon. \tag{3.23}$$

Adding and subtracting $x^T P_k B K_k x$ and $x^T K_k^T B^T P_K x$ from the equation, and using (2.12) & (2.13), equation (3.23) is expressed as:

$$\begin{aligned}\dot{V}_k(x) &= -x^T\{Q + K_{k+1}^T R K_{k+1}\}x + 2x^T P_k B\epsilon \\ &\quad - x^T\{(K_k - K_{k+1})^T R(K_k - K_{k+1})\}x \\ &\leq -\lambda_{min}(Q)\|x\|^2 + 2\epsilon_m\|P_k B\|\ \|x\|,\end{aligned} \tag{3.24}$$

where $\lambda_{min}(\cdot)$ refers to the minimum eigenvalue of the argument matrix. Inequality (3.24) establishes that the control policy K_{k+1} is stable, and the closed-loop dynamics $\dot{x} = A_{k+1}x + B\epsilon$ is UUB [56].

Lemma 3.5. *P_k obtained from (3.22) is equivalent to the solution of the ARE (2.12), if the closed loop system $\dot{x} = A_k x + B\epsilon$ where $A_k = A - BK_k$ is stable.*

Proof: Here, P_k is obtained from (3.22) is derived from (3.17) which is further deduced from (3.5). If we rewrite (3.5) using (3.4) as

$$x^T P_k(A_k x + B\epsilon) + (A_k x + B\epsilon)^T P_k x = -x^T(Q + K_k^T R K_k)x + 2x^T P_k B\epsilon$$

$$\implies x^T(A_k^T P_K + P_k A_k + Q + K_k^T R K_k)x = 0$$

which is equivalent to (2.12).

Remark 3.6. *Lemma 3.5 is sufficient to conclude that all intermediate policies obtained using (3.22) are stabilizing, provided the initial control policy is stabilizing.*

Theorem 3.7. *The proposed PI algorithm (3.22) provides convergence of intermediate policies to the optimal control policy, for an initial stabilizing policy.*

Proof: Kleinman's algorithm is proven to establish optimal convergence of the ARE for the PI scheme (2.12) & (2.13).[4] Lemma 3.5 establishes equivalence of the solution of (3.22) to Kleinman's algorithm. Moreover, Theorem 3.4 shows that the intermediate control policies obtained using (3.22) are also stabilizing.

[4]Refer to 2.7

3.3.2 Closed-loop stability analysis

We proved using Lemma 3.6 and Theorem 3.7 that the intermediate control policies are stabilizing and converge to optimal control. Nevertheless, the closed loop system obtained using the control law (3.4)

$$\dot{x} = (A - BK_k)x + B\epsilon, \qquad t \in [T_k, T_{k+1}], \tag{3.25}$$

is a switched closed-loop system due to the iterative policy updates (K_ks) at update instances (T_ks); and [57] shows that the stability of intermediate steps of a switched closed-loop system may not always guarantee the stability of the overall system.

Theorem 3.8. *The control policy (3.4) guarantees global UUB stability of the overall switched closed-loop system (3.25) for the LTI system (3.1), iff $Q > 0$ and Assumption 2 is satisfied.*

Proof: The proof for the same is provided in [43], and goes as follows... Consider (3.3) as a radially unbounded Lyapunov candidate function, then the following inequality can be established for V_k, which is exploited in this proof later.

$$\lambda_{\min}(P_k)\,\|x(t)\|^2 \leq V_k(t) \leq \lambda_{\max}(P_k)\,\|x(t)\|^2, \quad \forall t \geq T_0, \tag{3.26}$$

where $\lambda_{\max}(\cdot)$ denote the maximum eigenvalue of the argument matrix. Differentiating V_k along the closed-loop system dynamics (3.25) for $t \in [T_k, T_{k+1})$ yields

$$\begin{aligned}\dot{V}_k &= -x^T \left(Q + K_k^T R K_k\right) x + 2x^T P_k B\epsilon \\ &\leq -\lambda_{\min}(Q)\|x\|^2 + 2\epsilon_m \|P_k\| \, \|B\| \|x\|, \\ &\quad t \in [T_k, T_{k+1}), \quad \forall k \in \mathbb{W}.\end{aligned}$$

Expressing $\lambda_{\min}(Q) = \frac{\lambda_{\min}(Q)}{2} + \frac{\lambda_{\min}(Q)}{2}$ and completing the squares yields

$$\begin{aligned}\dot{V}_k \leq \frac{-\lambda_{\min}(Q)}{2}\|x\|^2 + \frac{2\epsilon_m^2 \|B\|^2 \|P_k\|^2}{\lambda_{\min}(Q)}, \\ t \in [T_k, T_{k+1}), \quad \forall k \in \mathbb{W}.\end{aligned} \tag{3.27}$$

Using (3.26) and the standard definition of matrix norm, $\|P_k\| = \sigma_{\max}(P_k) = \lambda_{\max}(P_k)$ (since P_k is positive definite and symmetric),[5] inequality (3.27) can be written as

$$\begin{aligned}\dot{V}_k \leq \underbrace{\frac{-\lambda_{\min}(Q)}{2\lambda_{\max}(P_k)}}_{\sigma_k} V_k + \frac{2\epsilon_m^2 \|B\|^2 \lambda_{\max}^2(P_k)}{\lambda_{\min}(Q)}, \\ t \in [T_k, T_{k+1}), \quad \forall k \in \mathbb{W}.\end{aligned} \tag{3.28}$$

Using the comparison Lemma 3.4 of [56], the following inequality can be deduced from (3.28)

$$\begin{aligned}V_k(t) \leq e^{-\sigma_k(t-T_k)} V_k(T_k) + \beta_k \left(1 - e^{-\sigma_k(t-T_k)}\right), \\ t \in [T_k, T_{k+1}), \quad \forall k \in \mathbb{W},\end{aligned} \tag{3.29}$$

where the bound $\beta_k \in \mathbb{R}^+$of k^{th} subsystem is defined as

$$\beta_k \triangleq \frac{4\epsilon_m^2 \|B\|^2 \lambda_{\max}^3(P_k)}{\lambda_{\min}^2(Q)}. \tag{3.30}$$

[5] σ_{max} is the largest singular value of P_k

Since the equivalence between the proposed algorithm and the Kleinman's algorithm is already established in Lemma 3, hence from Lemma 3 and Theorem 1, it can be stated that Since Lemma 3.5 confirms that the solution obtained using (3.22) is equivalent to that of the Kleinman's algorithm, using Theorem 2.1 it can be stated

$$P_{k+1} \leq P_k \leq P_{k-1}. \tag{3.31}$$

Assuming $\bar{v} \in \mathbb{R}^n$ be the eigenvector of $\lambda_{\max}(P_k)$ and using the inequality (3.31), following is established,

$$\begin{aligned} \lambda_{\max}(P_{k-1}) \|\bar{v}\|^2 &\geq \bar{v}^T P_{k-1} \bar{v}, \quad \text{using (34)} \\ &\geq \bar{v}^T P_k \bar{v} = \lambda_{\max}(P_k) \|\bar{v}\|^2, \end{aligned} \tag{3.32}$$

which implies $\lambda_{\max}(P_{k-1}) \geq \lambda_{\max}(P_k)$. Using (3.30) and (3.32), an important relation can be established between consecutive ultimate bounds β_{k-1} and β_k as follows

$$\beta_{k-1} \geq \beta_k, \quad \forall k \in \mathbb{W}. \tag{3.33}$$

Moreover, (3.31) can be used to establish,

$$\underbrace{x^T(T_{k+1}) P_{k+1} x(T_{k+1})}_{V_{k+1}(T_{k+1})} \leq \underbrace{x^T(T_{k+1}) P_k x(T_{k+1})}_{V_k(T_{k+1})}. \tag{3.34}$$

If (3.29) is used to upper bound $V_k(t)$ in $[T_k, T_{k+1})$ as

$$V_k(t) \leq \max\{V_k(T_k), \beta_k\}, \tag{3.35}$$

then, with repeated application of (3.34) and (3.35), (3.35) can be expressed as

$$\begin{aligned} V_k(t) &\leq \max\left\{V_{k-1}\left(T_k\right), \beta_k\right\}, && \text{from (3.34)} \\ &\leq \max\left[\max\left\{V_{k-1}\left(T_{k-1}\right), \beta_{k-1}\right\}, \beta_k\right], && \text{from (3.35)} \\ &\leq \max\left[V_{k-1}\left(T_{k-1}\right), \beta_{k-1}\right], && \text{from (3.33)} \\ &\;\;\vdots \\ &\leq \max\left\{V_0\left(T_0 = 0\right), \beta_0\right\}, \forall k \in \mathbb{W}, \forall t \geq T_0. \end{aligned} \tag{3.36}$$

And as, $P_{k+1} \leq P_k$, assuming $\underline{v} \in \mathbb{R}^n$ be the eigenvector of $\lambda_{\min}\ (P_k)$,

$$\begin{aligned} \lambda_{\min}\left(P_{k+1}\right)\|\underline{v}\|^2 &\leq \underline{v}^T P_{k+1}\underline{v}, && \text{using (3.26)} \\ &\leq \underline{v}^T P_k \underline{v} = \lambda_{\min}\left(P_k\right)\|\underline{v}\|^2, \end{aligned} \tag{3.37}$$

implies $\lambda_{\min}\left(P_{k+1}\right) \leq \lambda_{\min}\left(P_k\right)$. Using (3.26), the following can be deduced,

$$\|x(t)\|^2 \leq \frac{V_k(t)}{\lambda_{\min}\left(P_k\right)} \tag{3.38}$$

$$\begin{aligned} &\leq \frac{\max\left\{V_0(0), \beta_0\right\}}{\lambda_{\min}\left(P_k\right)}, && \text{from (3.36)} \\ &\leq \frac{\max\left\{V_0(0), \beta_0\right\}}{\lambda_{\min}\left(P_{k+1}\right)}, && \text{from (3.37)} \\ &\leq \frac{\max\left\{V_0(0), \beta_0\right\}}{\lambda_{\min}\left(P^*\right)}, && \text{since } P^* \leq P_{k+1} \\ &\leq \frac{\max\left\{\lambda_{\max}\left(P_0\right)\|x(0)\|^2, \beta_0\right\}}{\lambda_{\min}\left(P^*\right)}, && \text{from (3.26)} \end{aligned} \tag{3.39}$$

Hence, the inequality (3.39) implies that the states of the overall (switched) closed-loop system dynamics (3.25) are globally uniformly bounded. Further to determine

the uniform ultimate bound, consider, as $k \to \infty$ and $t \to \infty$, the inequality (3.38) can be written using (3.29) as

$$\begin{aligned}&\lim_{t\to\infty} \|x(t)\|^2 \\ &\quad \leq \lim_{\substack{k\to\infty \\ t\to\infty}} \frac{e^{-\sigma_k(t-T_k)} V_k\left(T_k\right) + \beta_k\left(1 - e^{-\sigma_k(t-T_k)}\right)}{\lambda_{\min}\left(P_k\right)}\end{aligned}$$

Using Assumption 2 and Lemma 3.5, $\lim_{k\to\infty} V_k \to V^*$, $\lim_{k\to\infty} P_k \to P^*$, which yields

$$\begin{aligned}\lim_{t\to\infty} \|x(t)\|^2 &\leq \frac{\beta^*}{\lambda_{\min}\left(P^*\right)} \\ \Rightarrow \lim_{t\to\infty} \|x(t)\| &\leq \sqrt{\frac{\beta^*}{\lambda_{\min}\left(P^*\right)}},\end{aligned} \tag{3.40}$$

where $\lim_{k\to\infty} \beta_k \to \beta^* \triangleq \frac{4\epsilon_m^2 \|B\|^2 \lambda_{\max}^3(P^*)}{\lambda_{\min}^2(Q)}$, where $\sqrt{\frac{\beta^*}{\lambda_{\min}(P^*)}}$ is the UUB. Therefore, using (3.39) and (3.40), we show (3.25) to have global UUB stability.

Chapter 4

Experimental Setup and Results

The focus of this chapter is to demonstrate the practical efficacy of the proposed PI algorithm; for this purpose, we consider the problem of angular position control of the shaft in a DC motor [58].

The state space representation of the system is noted as:

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 4.438 \\ 0 & -12 & -24 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 20 \end{bmatrix} u.$$

The weight matrices are selected as $Q = I_3$ and $R = 1$.

For this system, the optimal values of the controller parameters known to be:

$$K^* = \begin{bmatrix} 1.0000 & 0.8549 & 0.4791 \end{bmatrix},$$

and

$$P^* = \begin{bmatrix} 1.4549 & 0.3783 & 0.0500 \\ 0.3783 & 0.3770 & 0.0427 \\ 0.0500 & 0.0427 & 0.0240 \end{bmatrix}.$$

The filter gains are chosen to be $p_l = 0.5$, and $p_{ll} = 0.1$.

The excitation signal was chosen on a trial and error basis by adding 100 sinusoidal waveforms with amplitude 1 where the frequencies where uniformly distributed between $[1, 500]$.

The initial state of the system was set as $x_0 = \begin{bmatrix} 10 & -15 & 10 \end{bmatrix}^T$ and the initial stable gain is $K_0 = \begin{bmatrix} 0.4732 & -0.1043 & -0.0500 \end{bmatrix}$. With this setup, the base system behaviour is shown in figure 4.1

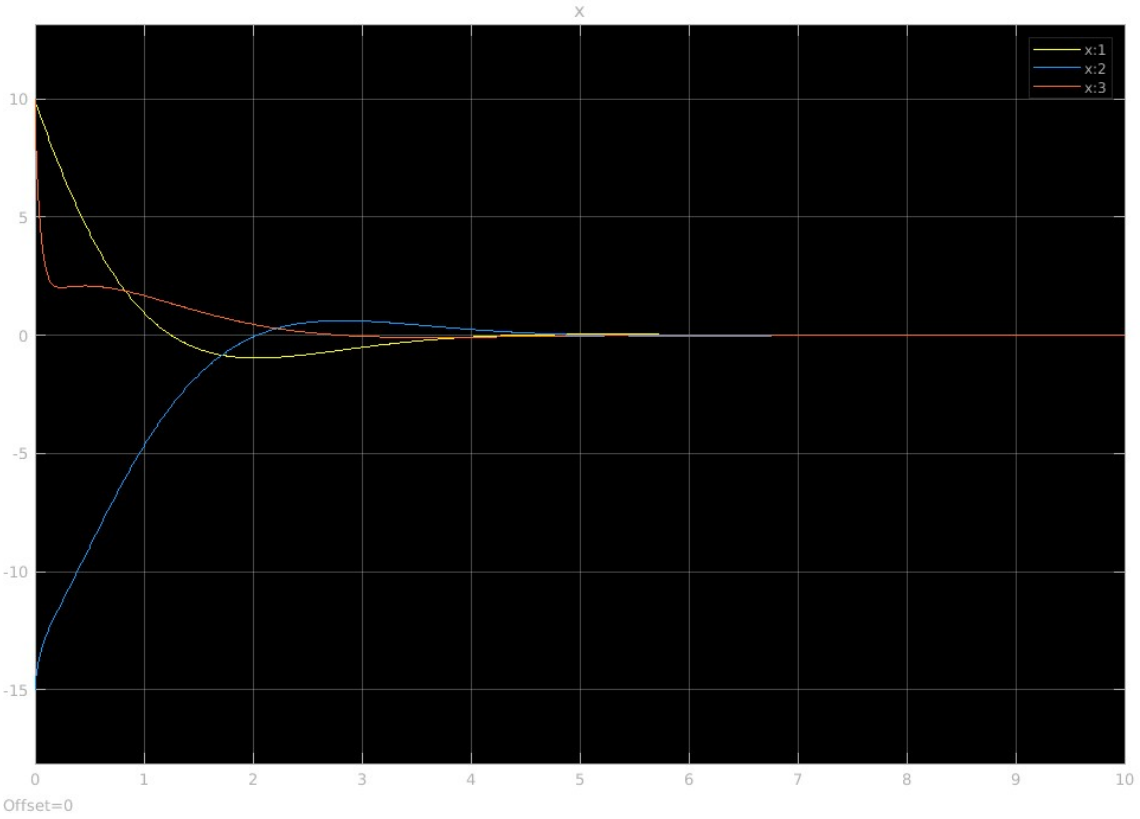

FIGURE 4.1: System dynamics with initial gain.

After adding excitation signal, the system dynamics look like:

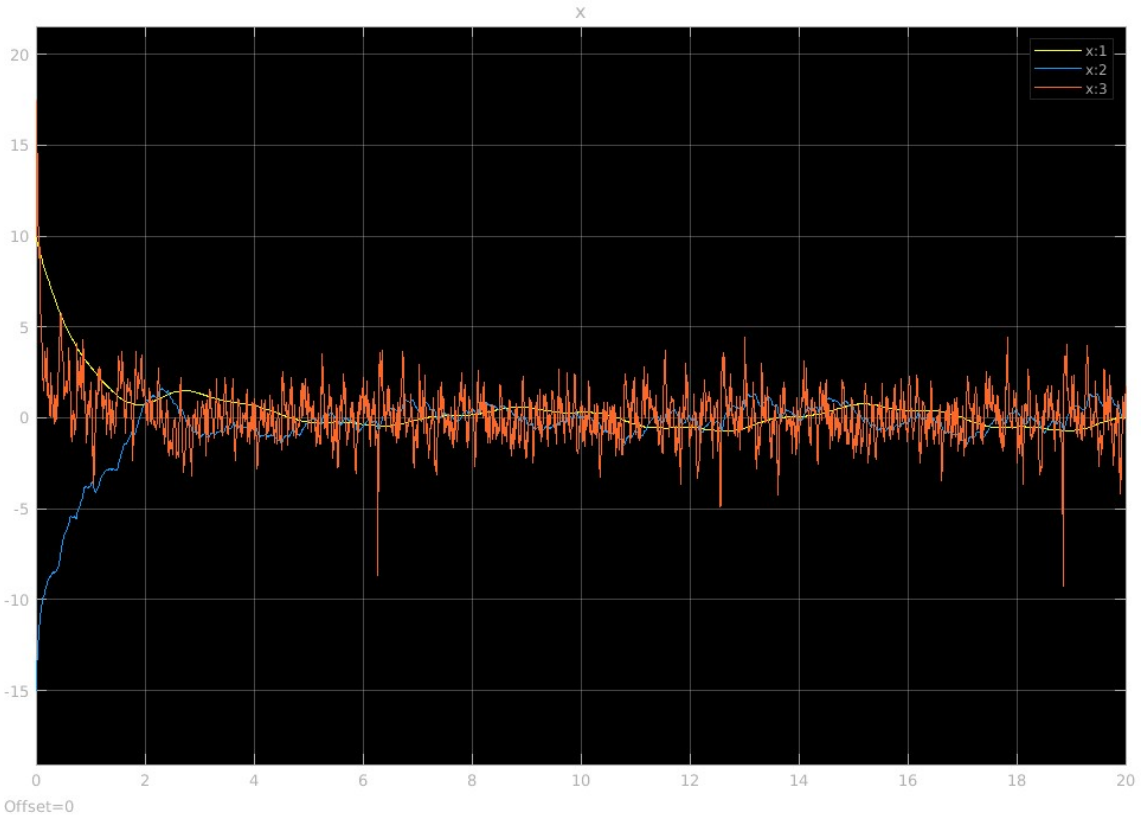

FIGURE 4.2: System with excitation.

4.1 Exemplary Results

Figure 4.3 showcases the efficacy of the algorithm. Where for the initial conditions defined above, we were able to iteratively converge (sufficiently close) to the optimal gain. The lines with "Flagged" list the time instance of policy (gain) update. And 'P_diff' and 'K_diff' variables denote the difference between the optimal values of these parameters and the newly learnt value of said parameters. As can be observed, both the parameters tend towards their optimal value.

```
Flagged 2.680000
DET_P =
    0.0093
EIG_P =
   0.0182 + 0.0000i
   0.2835 + 0.0000i
   1.8066 + 0.0000i
K =
    1.3526    1.3433    0.8698
EIG =
 -36.7982 + 0.0000i
  -0.8765 + 0.0000i
  -3.7222 + 0.0000i
P_diff =
   -0.2043   -0.0632   -0.0391
   -0.0632   -0.0349   -0.0438
   -0.0391   -0.0438   -0.0132
K_diff =
   -0.3526   -0.4884   -0.3907
Flagged 8.620000
DET_P =
  -6.4773e-04
EIG_P =
  -0.0015 + 0.0000i
   0.2807 + 0.0000i
   1.5257 + 0.0000i
K =
    0.8800    0.8750    0.5163
EIG =
 -30.0557 + 0.0000i
  -0.7354 + 0.0000i
  -3.5340 + 0.0000i
P_diff =
    0.0305    0.0391    0.0215
    0.0391   -0.0019    0.0080
    0.0215    0.0080    0.0223
K_diff =
    0.1200   -0.0201   -0.0372
```

FIGURE 4.3: Results

4.2 Simulink Model

Figure 4.4 presents the complete model diagram used to implement this algorithm and produce the above listed results.

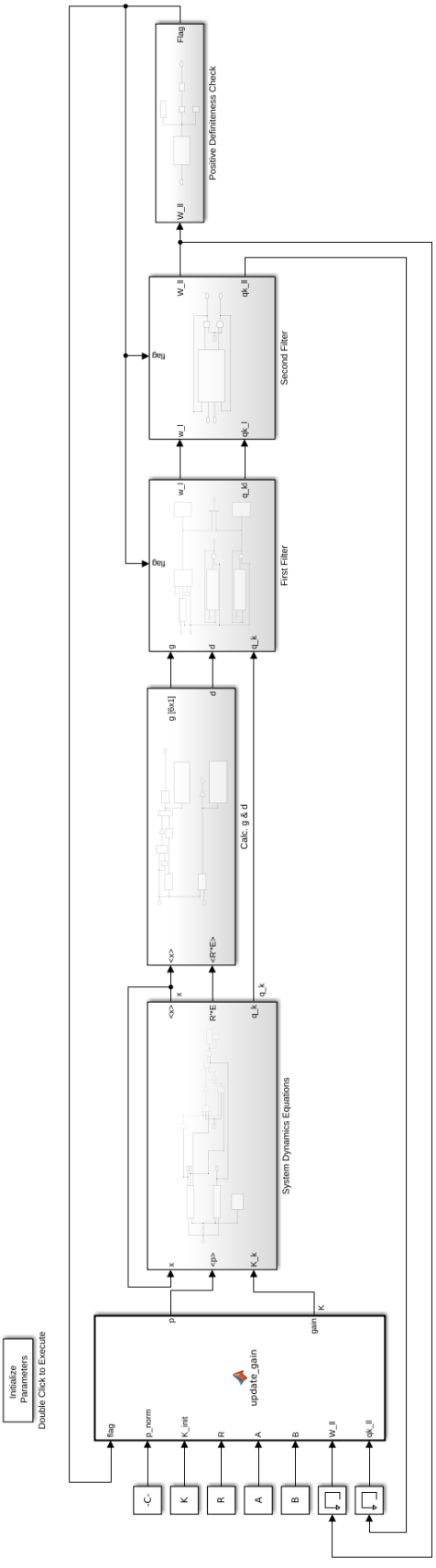

FIGURE 4.4: Simulink Model

Chapter 5

Conclusion

In the final chapter of our thesis, we summarise the contributions of our proposed algorithm. We also list the current limitation of the algorithm.

The proposed PI algorithm has been shown to work on policy in an online setting to adapt to the optimal solution of the LQR problem where system parameters were completely unknown. We were able to showcase the efficiency of our algorithm in comparison to other proposed methods. Our solution does not require observing the state derivatives, is memory efficient without requiring the use of FWIs or other intelligent data storage mechanisms, and provides UUB stability and convergence guarantees for both intermediate policies and the overall (switched) closed-loop system. Furthermore, through the use of Simulink & MATLAB, we also showcase our algorithm's application in a continuous-time system model.

However, we must also note the current shortcomings (or limitations) of the current version of the algorithm.

- The scope of the current algorithm is limited to infinite-horizon LQR problems, as the basis for this solution was derived from Klienman's algorithm, which provides a method to approximate the optimal solution to the ARE, which in turn is only a result of the HJB equation for such problems.

- This is an approximation solution, which means the solution obtained may not be optimal, only an approximation that is sufficiently close to the optimal solution.

- The algorithm demands an initial stabilizing gain. Although setting the initial gain to 0 works for many systems, it might not work for most systems.

- The efficiency of the algorithm is also dependent on the initial gain K_0 and the exploration signal $\epsilon(t)$. The choice of either affects how the algorithm performs. Although any stabilizing K_0 will work for the algorithm, not every exploration signal could provide enough excitation to the system to explore the state space.

- Simulation of the algorithm was also prone to instability due to simulations being inherently limited to discrete-time steps and certain significant digits after the decimal, where an error in calculation between time steps could accumulate in the output of the two filters leading to system instability.

Scope for Future Work

The ability to provide adaptive optimal control solutions to systems with unknown dynamics has tremendous potential in itself, with a wide variety of applications.

The immediate scope for future work includes implementing the algorithm to a real-world problem, like controlling an inverted pendulum, to test the algorithm's competence in interacting with real systems. Such advances can be further analyzed and appended with robustness criteria to work against noises and disturbances that real-world systems have to face.

Furthermore, the algorithm can be attempted to work with problems other than the infinite-horizon LQR, such as the linear quadratic tracker (LQT). There is even scope to extend the application of the algorithm to non-linear systems.

References

[1] Aaron Strauss. *An Introduction to Optimal Control Theory*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1968. ISBN 978-3-540-04252-5 978-3-642-51001-4. doi: 10.1007/978-3-642-51001-4. URL `http://link.springer.com/10.1007/978-3-642-51001-4`.

[2] Donald E. Kirk. *Optimal control theory: an introduction*. Dover Publications, Mineola, N.Y, 2004. ISBN 978-0-486-43484-1.

[3] Rudolf E. Kálmán. Contributions to the Theory of Optimal Control. *Boletin de la Sociedad Matematica Mexicana*, 5(2):102–119, 1960. URL `https://www.ee.iitb.ac.in/~belur/ee640/optimal-classic-paper.pdf`.

[4] Shankar Sastry and Marc Bodson. *Adaptive control: stability, convergence, and robustness*. Prentice-Hall information and system sciences series. Prentice Hall, Englewood Cliffs, N.J, 1989. ISBN 978-0-13-004326-9.

[5] Karl J. Åström and Björn Wittenmark. *Adaptive control*. Dover books on engineering. Dover Publications, Mineola, N.Y, 2nd ed., dover ed edition, 2008. ISBN 978-0-486-46278-3.

[6] Draguna L. Vrabie, K. G. Vamvoudakis, and Frank L. Lewis. *Optimal adaptive control and differential games by reinforcement learning principles*. Number 81

in IET control engineering series. Inst. of Engineering and Technology, Stevenage, 2013. ISBN 978-1-84919-490-7 978-1-84919-489-1.

[7] David Sworder. *Optimal adaptive control systems*. Elsevier, Burlington, 1966. ISBN 978-0-08-095531-5.

[8] Robert R. Bitmead, Michel Gevers, and Vincent Wertz. *Adaptive optimal control: the thinking man's GPC*. Prentice Hall international series in systems and control engineering. Prentice Hall, New York, 1990. ISBN 978-0-13-013277-2.

[9] R.S. Sutton, A.G. Barto, and R.J. Williams. Reinforcement learning is direct adaptive optimal control. *IEEE Control Systems Magazine*, 12(2):19–22, April 1992. ISSN 1941-000X. doi: 10.1109/37.126844.

[10] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: an introduction*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass, 1998. ISBN 978-0-262-19398-6.

[11] L. BALZER. Accelerated convergence of the matrix sign function method of solving Lyapunov, Riccati and other matrix equations. *International Journal of Control*, 32:1057–1078, April 2007. doi: 10.1080/00207178008910040.

[12] Ralph Byers. Solving the algebraic Riccati equation with the matrix sign function. *Linear Algebra and its Applications*, 85:267–279, January 1987. ISSN 0024-3795. doi: 10.1016/0024-3795(87)90222-9. URL `https://www.sciencedirect.com/science/article/pii/0024379587902229`.

[13] Mohammed A Hasan and Ali A. Hasan. Method for solving the algebraic Riccati and Lyapunov equations using higher order matrix sign function algorithms: Proceedings of the 1998 37th IEEE Conference on Decision and Control (CDC).

Proceedings of the IEEE Conference on Decision and Control, 4:4416–4421, December 1998. ISSN 0191-2216. URL http://www.scopus.com/inward/record.url?scp=0032278489&partnerID=8YFLogxK.

[14] A. G. J. MACFARLANE. An Eigenvector Solution of the Optimal Linear Regulator Problem. *Journal of Electronics and Control*, 14(6):643–654, June 1963. ISSN 0368-1947. doi: 10.1080/00207216308937540. URL https://doi.org/10.1080/00207216308937540.

[15] James E. Potter. Matrix Quadratic Solutions. *SIAM Journal on Applied Mathematics*, 14(3):496–501, March 1966. ISSN 0036-1399. doi: 10.1137/0114044. URL https://epubs.siam.org/doi/10.1137/0114044.

[16] H. T. Banks and K. Ito. A Numerical Algorithm for Optimal Feedback Gains in High Dimensional Linear Quadratic Regulator Problems. *SIAM Journal on Control and Optimization*, 29(3):499–515, May 1991. ISSN 0363-0129. doi: 10.1137/0329029. URL https://epubs.siam.org/doi/10.1137/0329029.

[17] Chun-Hua Guo and Peter Lancaster. Analysis and modificaton of Newton's method for algebraic Riccati equations. *Mathematics of Computation*, 67(223):1089–1105, 1998. ISSN 0025-5718, 1088-6842. doi: 10.1090/S0025-5718-98-00947-8. URL https://www.ams.org/mcom/1998-67-223/S0025-5718-98-00947-8/.

[18] D. Kleinman. On an iterative technique for Riccati equation computations. *IEEE Transactions on Automatic Control*, 13(1):114–115, February 1968. ISSN 1558-2523. doi: 10.1109/TAC.1968.1098829.

[19] Randy A. Freeman and Petar Kokotović. *Robust Nonlinear Control Design*. Birkhäuser, Boston, MA, 1996. ISBN 978-0-8176-4758-2 978-0-8176-4759-9.

doi: 10.1007/978-0-8176-4759-9. URL http://link.springer.com/10.1007/978-0-8176-4759-9.

[20] Zhong-Hua Li and Miroslav Krstić. Optimal design of adaptive tracking controllers for non-linear systems. *Automatica*, 33(8):1459–1473, August 1997. ISSN 0005-1098. doi: 10.1016/S0005-1098(97)00072-1. URL https://www.sciencedirect.com/science/article/pii/S0005109897000721.

[21] P.J. Werbos. Neural networks for control and system identification. *Proceedings of the 28th IEEE Conference on Decision and Control*, pages 260–265, 1989. doi: 10.1109/CDC.1989.70114. URL http://ieeexplore.ieee.org/document/70114/.

[22] L.C. Baird. Reinforcement learning in continuous time: advantage updating. *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, pages 2448–2453 vol.4, 1994. doi: 10.1109/ICNN.1994.374604. URL https://ieeexplore.ieee.org/document/374604/.

[23] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*. Dynamic programming and optimal control / Dimitri P. Bertsekas, Massachusetts Institute of Technology. Athena Scientific, Belmont, Massachusetts, fourth edition, updated printing edition, 2018. ISBN 978-1-886529-08-3 978-1-886529-44-1.

[24] Warren B. Powell. *Approximate dynamic programming: solving the curses of dimensionality*. Wiley series in probability and statistics. Wiley, Hoboken, N.J, 2nd ed edition, 2011. ISBN 978-0-470-60445-8 978-1-118-02917-6 978-1-118-02915-2 978-1-118-02916-9.

[25] Frank L. Lewis, Draguna Vrabie, and Kyriakos G. Vamvoudakis. Reinforcement Learning and Feedback Control: Using Natural Decision Methods to Design

Optimal Adaptive Controllers. *IEEE Control Systems Magazine*, 32(6):76–105, December 2012. ISSN 1941-000X. doi: 10.1109/MCS.2012.2214134.

[26] J.J. Murray, C.J. Cox, G.G. Lendaris, and R. Saeks. Adaptive dynamic programming. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 32(2):140–153, May 2002. ISSN 1558-2442. doi: 10.1109/TSMCC.2002.801727.

[27] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. L. Lewis. Adaptive optimal control for continuous-time linear systems based on policy iteration. *Automatica*, 45(2):477–484, February 2009. ISSN 0005-1098. doi: 10.1016/j.automatica.2008.08.017. URL `https://www.sciencedirect.com/science/article/pii/S0005109808004469`.

[28] S.J. Bradtke, B.E. Ydstie, and A.G. Barto. Adaptive linear quadratic control using policy iteration. In *Proceedings of 1994 American Control Conference - ACC '94*, volume 3, pages 3475–3479 vol.3, June 1994. doi: 10.1109/ACC.1994.735224.

[29] Prashant Mehta and Sean Meyn. Q-learning and Pontryagin's Minimum Principle. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 3598–3605, December 2009. doi: 10.1109/CDC.2009.5399753.

[30] Asma Al-Tamimi, Frank L. Lewis, and Murad Abu-Khalaf. Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control. *Automatica*, 43(3):473–481, March 2007. ISSN 0005-1098. doi: 10.1016/j.automatica.2006.09.019. URL `https://www.sciencedirect.com/science/article/pii/S0005109806004249`.

[31] Pingan He and S. Jagannathan. Reinforcement Learning Neural-Network-Based Controller for Nonlinear Discrete-Time Systems With Input Constraints. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(2): 425–436, April 2007. ISSN 1941-0492. doi: 10.1109/TSMCB.2006.883869.

[32] Asma Al-Tamimi, Frank L. Lewis, and Murad Abu-Khalaf. Discrete-Time Nonlinear HJB Solution Using Approximate Dynamic Programming: Convergence Proof. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 38(4):943–949, August 2008. ISSN 1941-0492. doi: 10.1109/TSMCB.2008.926614.

[33] Kenji Doya. Reinforcement Learning in Continuous Time and Space. *Neural Computation*, 12(1):219–245, January 2000. ISSN 0899-7667. doi: 10.1162/089976600300015961. URL `https://doi.org/10.1162/089976600300015961`.

[34] Ronald A. Howard. *Dynamic programming and Markov processes.* Dynamic programming and Markov processes. John Wiley, Oxford, England, 1960.

[35] Dimitri P. Bertsekas and John N. Tsitsiklis. *Neuro-dynamic programming.* Optimization and neural computation series. Athena Scientific, Belmont, Mass, 1996. ISBN 978-1-886529-10-6.

[36] Jae Young Lee, Jin Bae Park, and Yoon Ho Choi. Integral Q-learning and explorized policy iteration for adaptive optimal control of continuous-time linear systems. *Automatica*, 48(11):2850–2859, November 2012. ISSN 0005-1098. doi: 10.1016/j.automatica.2012.06.008. URL `https://www.sciencedirect.com/science/article/pii/S0005109812002592`.

[37] Yu Jiang and Zhong-Ping Jiang. Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics. *Automatica*, 48(10):2699–2704, October 2012. ISSN 0005-1098. doi: 10.1016/

j.automatica.2012.06.096. URL https://www.sciencedirect.com/science/article/pii/S0005109812003664.

[38] Kyriakos G. Vamvoudakis. Q-learning for continuous-time linear systems: A model-free infinite horizon optimal control approach. *Systems & Control Letters*, 100:14–20, February 2017. ISSN 01676911. doi: 10.1016/j.sysconle.2016.12.003. URL https://linkinghub.elsevier.com/retrieve/pii/S0167691116301967.

[39] Hamidreza Modares, Frank L. Lewis, and Mohammad-Bagher Naghibi-Sistani. Adaptive Optimal Control of Unknown Constrained-Input Systems Using Policy Iteration and Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 24(10):1513–1525, October 2013. ISSN 2162-237X, 2162-2388. doi: 10.1109/TNNLS.2013.2276571. URL http://ieeexplore.ieee.org/document/6583978/.

[40] Rushikesh Kamalapurkar, Patrick Walters, and Warren E. Dixon. Model-based reinforcement learning for approximate optimal regulation. *Automatica*, 64:94–104, February 2016. ISSN 00051098. doi: 10.1016/j.automatica.2015.10.039. URL https://linkinghub.elsevier.com/retrieve/pii/S0005109815004392.

[41] Kyriakos G. Vamvoudakis, Marcio Fantini Miranda, and Joao P. Hespanha. Asymptotically Stable Adaptive–Optimal Control Algorithm With Saturating Actuators and Relaxed Persistence of Excitation. *IEEE Transactions on Neural Networks and Learning Systems*, 27(11):2386–2398, November 2016. ISSN 2162-237X, 2162-2388. doi: 10.1109/TNNLS.2015.2487972. URL http://ieeexplore.ieee.org/document/7302057/.

[42] Sumit Kumar Jha, Sayan Basu Roy, and Shubhendu Bhasin. Memory-Efficient Filter Based Novel Policy Iteration Technique for Adaptive LQR. In *2018 Annual American Control Conference (ACC)*, pages 4963–4968, June 2018. doi: 10.23919/ACC.2018.8431061.

[43] Sumit Kumar Jha, Sayan Basu Roy, and Shubhendu Bhasin. Filter based Explorized Policy Iteration Algorithm for On-Policy Approximate LQR. In *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 133–140, December 2019. doi: 10.1109/SSCI44817.2019.9002891.

[44] Graham C. Goodwin, Stefan F. Graebe, and Mario E. Salgado. *Control system design*. PHI Learning, New Delhi, 2009. ISBN 978-81-203-2119-9.

[45] Feng Lin. *Robust control design: an optimal control approach*. RSP series in control theory and applications. John Wiley/RSP, Chichester, West Sussex, England ; Hoboken, NJ, 2007. ISBN 978-0-470-03191-9.

[46] D. S. Naidu. *Optimal control systems*. Electrical engineering textbook series. CRC Press, Boca Raton, Fla, 2003. ISBN 978-0-8493-0892-5.

[47] Richard Bellman. The theory of dynamic programming. *Bulletin of the American Mathematical Society*, 60(6):503–515, 1954. ISSN 0273-0979, 1088-9485. doi: 10.1090/S0002-9904-1954-09848-8. URL `https://www.ams.org/bull/1954-60-06/S0002-9904-1954-09848-8/`.

[48] Kumpati S. Narendra and Anuradha M. Annaswamy. *Stable adaptive systems*. Dover Publications, Mineola, N.Y, 2005. ISBN 978-0-486-44226-6.

[49] A. Loria, E. Panteley, and A. Zavala-Rio. Adaptive Observers With Persistency of Excitation for Synchronization of Chaotic Systems. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 56(12):2703–2716, December 2009.

ISSN 1549-8328, 1558-0806. doi: 10.1109/TCSI.2009.2016636. URL http://ieeexplore.ieee.org/document/4798184/.

[50] 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, May 2010. ISSN 00051098. doi: 10.1016/j.automatica.2010.02.018. URL https://linkinghub.elsevier.com/retrieve/pii/S0005109810000907.

[51] Girish Chowdhary and Eric Johnson. Concurrent learning for convergence in adaptive control without persistency of excitation. In *49th IEEE Conference on Decision and Control (CDC)*, pages 3674–3679, December 2010. doi: 10.1109/CDC.2010.5717148.

[52] Girish Chowdhary, Tansel Yucelen, Maximillian Mühlegg, and Eric N. Johnson. Concurrent learning adaptive control of linear systems with exponentially convergent bounds: CONCURRENT LEARNING ADAPTIVE CONTROL OF LINEAR SYSTEMS. *International Journal of Adaptive Control and Signal Processing*, 27(4):280–301, April 2013. ISSN 08906327. doi: 10.1002/acs.2297. URL https://onlinelibrary.wiley.com/doi/10.1002/acs.2297.

[53] Dimitri P. Bertsekas. *Dynamic programming: deterministic and stochastic models*. Prentice-Hall, Englewood Cliffs, N.J, 1987. ISBN 978-0-13-221581-7.

[54] Nikita Barabanov and Romeo Ortega. On global asymptotic stability of x = (t) (t) x with not persistently exciting. *Systems & Control Letters*, 109:24–29, November 2017. ISSN 01676911. doi: 10.1016/j.sysconle.2017.09.005. URL https://linkinghub.elsevier.com/retrieve/pii/S016769111730169X.

[55] Sumit Kumar Jha, Sayan Basu Roy, and Shubhendu Bhasin. Supplement to "Initial Excitation based Iterative Algorithm for Approximate Optimal Control

of Completely Unknown LTI Systems" in IEEE Transactions on Automatic Control 2019. URL `https://web.iitd.ac.in/~sbhasin/docs/Supplement_TAC2019.pdf`.

[56] Hassan K. Khalil. *Nonlinear systems*. Prentice Hall, Upper Saddle River, N.J, 3rd ed edition, 2002. ISBN 978-0-13-067389-3.

[57] Daniel Liberzon. *Switching in Systems and Control*. Systems & Control: Foundations & Applications. Birkhäuser Boston, Boston, MA, 2003. ISBN 978-1-4612-6574-0 978-1-4612-0017-8. doi: 10.1007/978-1-4612-0017-8. URL `http://link.springer.com/10.1007/978-1-4612-0017-8`.

[58] Christopher Edwards and Sarah K. Spurgeon. *Sliding mode control: theory and applications*. Systems and control book series ; vol. 7. Taylor & Francis, London, 1998. ISBN 978-0-7484-0601-2.