Reinforcement Learning for Model-Free Output Feedback Optimal Control

---

A Dissertation

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

---

in partial fulfillment
of the requirements for the degree

Doctor of Philosophy

by

Syed Ali Asad Rizvi

May 2020

APPROVAL SHEET

This Dissertation
is submitted in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

Author Signature: Ali Asad

This Dissertation has been read and approved by the examining committee:

Advisor: Zongli Lin

Committee Member: Gang Tao

Committee Member: Stephen G. Wilson

Committee Member: Nikolaos Sidiropoulos

Committee Member: Hongning Wang

Committee Member:

Accepted for the School of Engineering and Applied Science:

Craig H. Benson, School of Engineering and Applied Science

May 2020

Abstract

Stability is a bare minimum requirement in the design of control systems. It is often desirable that a control system operates in a prescribed optimal manner. Traditionally, optimal control has been a model-based paradigm, which relies on the availability of an accurate system model to design a controller that achieves closed-loop stability while minimizing a certain cost function. An accurate system model is often difficult and sometimes impossible to obtain owing to the increasing complexity of systems and the ever presence of modeling uncertainties. Modeling inaccuracies not only compromise the optimality of the controller, but may also destabilize the system. The frameworks of adaptive and robust control address the modeling uncertainty problem but optimality is generally not the prime goal in these approaches.

Reinforcement learning (RL) is a form of machine learning, which is inspired by living organisms that learn and optimize their behavior based on their past experiences. Recently, RL has been used to design model-free controllers that are both adaptive and optimal. However, the applications of RL based controllers are still limited owing to some challenges. Being a data-driven approach, RL requires a number of sensors to capture the complete internal state of the system. In many applications, the measurement of the complete internal state is not feasible owing to the availability and cost associated with installing a large number of sensors. Another difficulty is that many of these methods need an initially stabilizing control. Furthermore, practical control systems are subject to challenges such as external disturbances, actuator limitations, and time delays. The development of RL algorithms that address these difficulties is essential for their practical applicability, which serves as a motivation of this research.

In the first part of this research, we present model-free output feedback RL algorithms to solve the optimal control problem for linear dynamical systems. The proposed output feedback methods relax the requirement of the number of sensors, and thereby, improve upon the cost, reliability and complexity of the control system. These new algorithms have the advantage that they do not incur any estimation bias because of the use of exploration signals. Furthermore, the need of employing a discounted cost function is eliminated in our approach, which has been a bottleneck in the earlier works in ensuring closed-loop stability. In the second part of this research, we address some practical control challenges in the design of RL controllers. We employ the framework of game theory to develop an output feedback model-free H-infinity controller, which is capable of rejecting external disturbances. Another practical issue we address is the saturation of the actuators. A model-free low gain feedback design method is developed that achieves global stabilization of the system without causing the actuators to saturate. Finally, this research also presents model-free techniques to handle time delays in the control loop. All these developments are aimed towards strengthening the framework of reinforcement learning control by enabling it to deal with practical control challenges.

This work is dedicated to my parents and grandmother, whose prayers and unconditional love and support have taken me far beyond what my capabilities would have allowed.

Acknowledgments

This work has been accomplished with the help and support of many people. Firstly, I would like to express my heartiest gratitude to my advisor Prof. Zongli Lin for believing in me and giving me the freedom to think and formulate my research. He is undoubtedly one of the finest mentors I have come across when it comes to guidance and motivation in both academic and practical life.

I would also like to thank my dissertation committee members for their valuable suggestions in improving this work. In particular, I would like to thank Prof. Gang Tao for teaching me the principles of adaptive control in his adaptive control course, which has helped me establishing connections between reinforcement learning and adaptive control. Last but not the least, Prof. Stephen Wilson's feedback has helped me improve this work and see things from a non-controls perspective.

In the end, I would like to thank my colleagues including Yusheng Wei, Yijing Xie and Tingyang Meng for their support and making my PhD journey a delightful one.

CONTENTS

1. INTRODUCTION

1.1. Optimal Control and Reinforcement Learning

We live in a technology driven world which is strongly dependent on the operation of complex systems found in automobiles, aerospace, industrial automation, manufacturing systems, communication, computers systems, defense, medicine, and finance. Control theory provides mathematical tools for designing control systems to obtain desired system behavior and ensure stable operation of these system. While stability is essential in controls, it is only a minimum requirement for the operation of a control system. It is desirable that a control system operates in a prescribed optimal manner to meet certain cost and performance objectives.

Optimal control theory is one of the founding pillars of modern control that enables the synthesis of optimal controllers. The optimal control problem is to design a control law which satisfies the control objectives while minimizing a prescribed cost function that reflects the balance between the desired performance and the available resources. The foundations of optimal control theory were laid in the 1950's based on the Pontryagin's minimum principle [15] and Bellman's theory of dynamic programming [12]. Formally, the solution of the optimal control problem is obtained by solving the famous Hamilton-Jacobi-Bellman (HJB) equation. Although mathematically elegant, the solution of the HJB equation is intractable to obtain because it is a partial differential equation. Owing to this difficulty, a significant effort has been devoted to approximate the solution of the HJB equation [8]–[11]. For the special case of linear systems with quadratic cost functions, solving the HJB equation amounts to finding the solution of the nonlinear algebraic Riccati equation (ARE). Computational iterative schemes are used to overcome the difficulty in solving these equations [24], [33]. All these optimal control design algorithms are offline in nature and require a complete knowledge of the system dynamics.

Optimal control theory is fundamentally a model-based control paradigm which employs system models of the underlying process to obtain the optimal control law. The HJB and the ARE equations incorporate the complete knowledge of the dynamical system, and therefore, the optimality of the solution is subject to the accuracy of the system model. The requirement of accurate system models is hard to satisfy in practice because of the presence of unavoidable modeling uncertainties. Furthermore, these uncertainties proportionally add up as the complexity of system increases. More often, the modeling and system identification process is undesirable owing to the cost and effort associated with these processes. As a result, the dynamics of the underlying system are often completely unknown to the controller. Even when

the system dynamics are available in the early design phase, the system parameters are subject to change over the operating life span of the system due to variations in the process itself. These model variations may arise as a result of aging, faults or subsequent upgradation of the system components. Thus, it is desirable to develop optimal control methods that do not rely on the knowledge of the system dynamics.

Machine learning is an area of artificial intelligence which involves the development of computational intelligence algorithms that learn by analyzing the system behavior instead of being explicitly programmed to perform certain tasks. Recent advances in machine learning have opened a new avenue for the design of intelligent controllers. Machine learning techniques come into the picture when accurate system models are not known in advance but examples of system behavior are available or a measure of the goodness of the behavior can be assigned [6], [46]. Machine learning algorithms are formally categorized into three types based on the kind of supervision available, namely supervised learning, unsupervised learning and reinforcement learning. Supervised learning is undoubtedly the most common type of machine learning which finds use in applications where the examples of the desired behavior are available in the form of input-output training data. However, the nature of control problems requires selecting control actions whose consequences emerge over time for which the optimal control inputs that achieve the desired output are not known in advance. In these scenarios, reinforcement learning can be used to learn the unknown desired control inputs by providing the controller with a suitable evaluation of its performance during the course of learning.

Reinforcement learning (RL) is inspired from animal learning in which the living organisms exhibit the ability to learn, adapt and optimize their behavior over time by interacting with their environment. RL techniques are computational intelligence algorithms which are based on the principle of action and reward that can be used to solve dynamic optimization problems in which an agent (controller) interacts with its environment (system) and modifies its actions (control policies), based on some stimuli (reward) received in response to its actions. This is based on evaluative information from the environment and is referred to as action-based learning. RL implies a cause and effect relationship between actions and reward or punishment. RL naturally incorporates a feedback mechanism, a fundamental concept in control theory, to capture optimal behavior, and therefore, it fits well in the framework of feedback control of dynamical systems. Fig. 1.1 shows a block diagram of reinforcement learning system, which shows that an RL algorithm incorporates a reward feedback signal in addition to the system state feedback. This reward mechanism is what enables the controller to improve its policies, thereby making

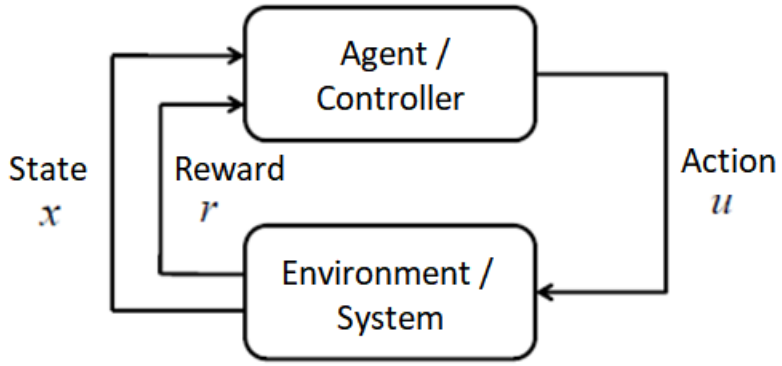

Fig. 1.1: Reinforcement learning is action-based learning that employs a feedback mechanism.

it different from the standard feedback control systems. Reinforcement learning methods were first developed for sequential decision making problems with finite state-spaces such as those found in the Markov decision processes (MDP). Although dynamic programming (DP) principles have been developed to solve sequential decision problems, they require a complete specification of the MDP in terms of its probabilistic model. Furthermore, their computational complexity makes DP impractical for large-scale problems as the size of the state-space increases. Reinforcement learning solves these decision making problems by searching for an optimal policy that maximizes the expected value of the performance measure by examining the reward at each step. Examples of such sequential decision processes where reinforcement learning has been successfully applied include robots navigation problems, board games such as chess, and more recently the Google Alpha Go.

Recently, reinforcement learning has received significant attention in the automatic control community for the control of dynamical systems. These dynamical systems are represented by differential or difference equations and are the most studied systems in engineering. System dynamics plays an essential part in the design of human engineered systems. RL based control of dynamical system requires to incorporate the system dynamics with infinite state-spaces, which makes it different from the traditional RL algorithms. With the recent advances in function approximation such as neural networks, RL techniques have been extended to approximate the solution of HJB equation based on dynamic programming principles. These methods are often referred in the control literature as approximate or adaptive dynamic programming (ADP) [36]. The development of reinforcement learning controllers is motivated by their optimal and model-free nature. Reinforcement learning controllers are inherently optimal because the problem for-

mulation is carried over some performance criterion or cost function. Thus, several optimization criteria, such as minimum energy, and minimum time, can be taken into account. In addition to being optimal, reinforcement learning also inherits adaptation capability in which the controller is able to adapt to the changes in system dynamics during its operation by observing the real-time data. In other words, the controller does not need to be reprogrammed if some parameters of the systems are changed. Reinforcement learning control is different from the adaptive control theory in the sense that adaptive control techniques adapt the controller parameters based on the error between the desired output and the measured output. As a result, the learned controllers do not take into account the optimality aspect of the problem. On the other hand, RL techniques are based on the Bellman equation which incorporates a reward signal to adapt the controller parameters, and therefore, optimality is considered while ensuring the control objectives are satisfied. The controllers based on RL methods are essentially direct adaptive controllers [59] in which the controller is designed without the model identification process, and the optimal control is learned online through the learning episodes by reinforcing the past control actions that give the maximum reward or minimum control utility.

1.2. Recent Challenges

While there have been significant developments in reinforcement learning based control algorithms over the past ten years, the design of reinforcement learning controllers have some challenges which need to be overcome to harness the full potential of these methods. The key difference between the classical reinforcement learning and reinforcement learning for dynamical systems is that in the latter, the system dynamics has to be taken into account. Unlike the traditional reinforcement learning applications such as AI games, the control of dynamical systems gives prime importance to the closed-loop system stability. Control algorithms are required to have guaranteed stability properties, which is a bare minimum requirement for feedback control. Control systems without performance, robustness and safety margins are not acceptable by the industry. As a result of these challenges, more rigorous results are needed that could provide sufficient stability and performance guarantees in order for these learning techniques to become mainstream in control applications. We highlight here some existing challenges being faced in applying reinforcement learning to control applications.

Reinforcement learning is a data-driven approach, and like other machine learning methods, it relies on the availability of sufficient data to make decisions. This data is obtained in real-time based on the

feedback of the system internal state, which is used to compute the reward signal and to apply the feedback control input to the system. This is referred to as state feedback, which requires as many sensors as is the order of the system. The difficulty with state feedback is that access to the complete internal state is generally not available in practice. The internal state of the system may not be a physically measurable phenomenon and a sensor may not be available to measure that state. Even when a sensor is available, it may not be feasible to install sensor to measure every component of the internal state owing to the cost and complexity of the resulting system. Furthermore, as the order of the system increases, this requirement becomes increasingly difficult to satisfy. In contrast to state feedback, control methods which employ system output feedback are more desirable. It is known that under certain observability assumptions the system input and output signals can be utilized to reconstruct the internal state of the system. However, output feedback control becomes quite challenging in the reinforcement learning paradigm because of the unavailability of the system model, which is needed to reconstruct the internal state. Thus, model-free output feedback methods should be sought to overcome the limitation of full state feedback.

An important requirement in reinforcement learning based methods is the exploration condition. In order to learn new policies, the state-space needs to be explored. However, there exists a trade-off between exploration of the new policies and exploitation of the current policies. During the exploration phase, the performance of the system may become undesirable. That is, the learning and the exploration itself incur a cost. On the other hand, if the state-space is not explored and only the exploitation of the already learned policies is carried out, then the improvement of the control policies is not possible. In practice, this exploration condition is met by injecting an exploration signal in the control which excites the system to learn the system behavior. This exploration requirement is similar to the excitation condition found in adaptive control [60]. However, in reinforcement learning based optimal adaptive control this condition is more important because the prime objective in RL controllers is to learn the optimal control parameters, and the convergence to these optimal parameters is only ensured when the state-space is well explored. Recent studies have shown that, under certain conditions, the excitation signal which is used to satisfy the exploration requirement may result in bias in the parameter estimates. Thus, excitation noise issue needs to be resolved to guarantee convergence to the optimal parameters.

In the original formulation of reinforcement learning for sequential decision problems, a discounting factor is employed in the cost function to ensure that the cost function is well defined. Discounted cost functions have also been used in the design of reinforcement learning controllers to address the issue

of excitation noise bias. However, it has been pointed out in the recent control literature [54] that the closed-loop system stability may be compromised due to the use of discounted cost functions. The issue stems from the need to make the long term cost of the control finite by assigning less weight to the future costs. However, this discounting factor masks the long term effect of the state energy in the cost function, and therefore, the convergence of the state is not guaranteed even when the cost becomes finite. Although the use of discounting factor is common in many sequential decision making problems, its application may not be feasible in control applications. Recent works tend to find a bound on this discounting factor which could still ensure the closed-loop stability. However, the computation of this bound requires the knowledge of the system model, which is unavailable in the case of model-free reinforcement learning control. Thus, undiscounted reinforcement learning controllers are sought upon.

Many reinforcement learning control algorithms require an initially stabilizing policy for their initialization and to guarantee convergence to the optimal stabilizing policy. This can be a restrictive assumption in some control applications because computing an initially stabilizing policy requires knowledge of system dynamics which is assumed to be unavailable in the case of reinforcement learning systems. For open-loop stable systems, the problem can be overcome by initializing the algorithm with an open-loop controller, however, for open-loop unstable systems, the selection of initially stabilizing policy is not straight forward.

In addition to the above challenges, practical control problems require disturbance rejection capability to counter the effect of external disturbances acting on the systems. The issue of actuator constraints is also of prime importance because actuator saturation can result in instabilities in the system. Time delays are also frequently encountered in control system, which need to be taken into account to ensure stable operation of the system. Thus, the overcoming of these challenges is essential for the applicability of RL control algorithms in practical applications.

1.3. Literature Survey

Optimal control problems rely on solving the HJB equation (for general nonlinear systems) or the ARE equation (for linear systems). Even when accurate system models are available, these equations are still difficult to solve analytically, and therefore, computational methods have been developed in the literature to solve these equations. Many of these methods are based on two computational techniques called policy iteration and value iteration, which were originally introduced in solving dynamic programming problems

for sequential decision processes such as MDP [25]. Instead of directly solving the HJB equation, these iterative algorithms recursively solve the Bellman equation that provides a recursive relationship between the current cost and the next step cost. In each iteration of the policy iteration algorithm, the Bellman equation is first solved by evaluating the cost or value of the current control policy and then the policy is improved based on its policy evaluation [39]. These two steps of policy evaluation and policy update are repeated until the algorithm sees no further updates in policy, and the final policy to said to be the optimal policy. The value iteration algorithm differs from the policy iteration algorithm only in the policy evaluation step, in which the value iteration evaluates a policy value based on the previous policy value. It should be noted that policy iteration generally requires solving a system of equations in each iteration, whereas, the value iteration simply performs a one-step recursion which is computationally economical. In contrast to policy iteration, the value iteration algorithm does not actually find the policy value corresponding to the current policy at each step but it takes only a step closer to that value. Also, the policy iteration algorithm must be suitably initialized to converge, i.e., the initial policy must be stabilizing. This is a limitation because such a policy may be difficult to obtain for complex systems. In contrast, the value iteration algorithm does not have this requirement. Although each step of value iteration is simpler than that of policy iteration, the policy iteration algorithm generally converges to the optimal solution in fewer iterations, as it does more work at each step. Another limitation of the value iteration is that only the final solution is guaranteed to be stabilizing while the iterative control sequences may not be stabilizing. The standard optimal control problem employs linear quadratic cost functions, which is referred to the linear quadratic regulation problem. Solving this problem requires the solution of a nonlinear Riccati equation [37]. To address the difficulty in solving these equations, a Newton iteration method based on the policy iteration algorithm was proposed in [33] to solve the ARE arising in the linear quadratic regulation problem of continuous-time linear systems. Later on, this method was extended to solve the discrete-time counterpart of the same problem in [24]. To overcome the requirement of an initially stabilizing policy and to reduce the computational complexity, a value iteration algorithm was presented in [34] to solve the ARE associated with the discrete-time linear quadratic regulation problem. Value iteration for continuous-time problems is, however, not straight forward owing to presence of differential matrix Riccati equations. Recently, an approximation based value iteration method was proposed towards solving the continuous-time ARE equation, where the requirement of initially stabilizing policy was removed. It should be noted that all of these design algorithms are model-based, and therefore, require

the complete knowledge of the system dynamics.

Reinforcement learning is an approach to solving optimal control problems without requiring full model information. The idea is to approximate the solution of HJB equation by performing some functional approximations using tools such as neural networks. Studies of RL based control methods generally consider one of the two main types of algorithms: actor-critic learning and Q-learning [59]. The actor-critic structure for RL control was introduced by Werbos [68]–[71], which he called approximate dynamic programming. The structure consists of a critic sub-system and an actor sub-system. The critic component assesses the cost of current action based on some optimality criteria similar to the policy evaluation step in PI and VI algorithms, while the actor component estimates an improved policy. However, in RL ADP, the critic and actor employ approximators such as neural networks to approximate the cost and control functions, respectively. Actor-critic algorithms employ value function approximation (VFA) to evaluate the value of the current policy. The use of function approximators instead of look-up tables overcomes the curse of dimensionality problem in traditional DP, which occurs when the state-space grows large. Actor-critic algorithms for the discrete-time and continuous-time LQR problems have been described in [40]. A partial knowledge of system dynamics (the input coupling term) is needed in these methods.

Q-learning is a simple yet powerful reinforcement learning algorithm which does not require any knowledge of system dynamics to design optimal controllers. This technique was introduced by Watkins [66] and is based on the idea of Q-function (Quality function), which is a function of state and control. The Q-function gives the cost of executing an arbitrary action in the current state and then following some specific policy. Since a Q-function contains information about control actions in every state, the best control action in each state can be selected by knowing only the Q-function. Thus, the main aim of the Q-learning algorithm is to estimate the optimal Q-function. Once the optimal Q-function is found, the optimal control can be readily obtained by finding the control action which minimizes or maximizes the optimal Q-function. The Q-function description is more comprehensive than the value function description as it spans the state-action space instead of the state-space. This enables Q-learning to employ only one function approximation structure instead of two separate approximators for critic (value function) and actor (control function). If the state-space is sufficiently explored than the Q-learning algorithm eventually converges to the optimal Q-function [67]. The success of Q-learning in controls is evident from the fact that it has provided model-free solution to the popular control problems. A Q-learning algorithm for the discrete-time linear quadratic regulation problem was proposed in [16]. In that work, the Q-learning

iterations were based on the policy iteration method, and therefore, the knowledge of an initially stabilizing policy was a requirement. It was shown that under certain excitation condition, the Q-learning algorithm converges to the optimal LQR controller. Later in [35], a value iteration based Q-learning LQR algorithm was presented and convergence to the optimal controller was shown. The requirement of an initially stabilizing policy was obviated in this work. Reinforcement learning has also been used to solve the robust control problem. A model-free Q-learning algorithm based on value iteration method was proposed for the robust H-infinity control problem [4] for discrete-time systems. In this work, the authors employed the framework of game theory and formulated the H-infinity problem as a zero-sum game. Q-learning has also been successfully used to design model-free tracking controllers. The objective in the model-free optimal tracking problem is to track a reference trajectory while minimizing certain cost without requiring access to the system model information. Recently, the authors in [31] employed Q-learning to solve the linear quadratic tracking (LQT) problem of discrete-time systems. A state augmentation approach was presented in this work, which combined the dynamics of reference trajectory generator with that of the original system to simultaneously learn the optimal feedback and feedforward terms. However, augmented system is not stabilizable because the reference generator is autonomous and neutrally stable. To overcome this difficulty, the authors in [31] had to resort to a discounting factor which makes the cost function well-defined even when the states of the reference system do not converge to zero. An appropriate value of this discounting factor is necessary for asymptotic tracking and closed-loop stability.

Most existing literature on RL control focuses on discrete-time systems as the continuous-time problem is more involved. This is due to the additional difficulty in the continuous-time Bellman equation, which requires the knowledge of the system dynamics [39] for the expression of the time derivative of the value function. This is in contrast to the discrete-time Bellman equation, where only the difference of the value function is involved in the Bellman equation. To address this problem, Vrabie [62] introduced integral reinforcement learning (IRL), in which the IRL Bellman equation does not involve system dynamics. In IRL, the reward signal is evaluated over some learning interval instead of discrete-time steps. As a result, the need of employing derivatives of the cost function and consequently invoking the system dynamics is circumvented. However, the knowledge of the input coupling matrix is needed in their work. A partially model-free policy iteration IRL algorithm was proposed in [64] to solve the continuous-time LQR problem, where the knowledge of an initially stabilizing policy and the input coupling matrix was needed. Later on, the authors of [27] proposed a completely model-free policy iteration IRL algorithm

to solve the same problem but the requirement of initially stabilizing policy was still present. This difficulty was recently overcome by the authors in [14], where the authors developed a model-free value iteration algorithm to solve the LQR problem, which did not require an initially stabilizing policy. IRL methods have also been proposed for optimal tracking problems. The authors in [47] presented a partially model-free IRL method to solve the LQT problem of continuous-time linear systems. However, similar to the discrete-time counterpart of this problem, the authors had to resort to discounted cost functions. Asymptotic tracking could not be ensured if the discounting factor is not carefully chosen. Furthermore, the discounted solution is sub-optimal as it does not correspond to the optimal solution obtained by solving the ARE. IRL methods have also been developed to solve the disturbance rejection problem using game theoretic methods. The continuous-time zero-sum game H-infinity problem has been solved using partially model-free [63] and completely model-free [41] IRL methods.

Interestingly, the RL notion of the agent and environment is also central to the multi-agent systems, and this makes RL a good candidate to solve control problems for multi-agent systems. RL has so far been quite successful in solving a wide range of multi-agent control problems without requiring the knowledge of system dynamics [17], [55]. Recently, a model-free reinforcement learning scheme was proposed in [61] to solve the optimal consensus problem of continuous-time multi-agent systems in which the leader and the follower agents were assumed to have identical dynamics. This work was later extended to the discrete-time setting [2]. The treatment of the heterogeneous multi-agent consensus problem, in the framework of output regulation, is quite challenging since it involves solving the output regulator equations for computing the feedforward term, which in turn requires the knowledge of the dynamics of both the leader and follower agents. This difficulty was recently overcome in [51], where the authors solved the continuous-time multi-agent consensus problem for heterogeneous agents using model-free reinforcement learning. Later on, a Q-learning based solution to the discrete-time version of this problem was proposed in [29]. Similar to the optimal tracking problem discussed earlier, [29], [51] employed a state augmentation approach to compute the feedback and feedforward terms simultaneously, and therefore, the authors had to resort to discounted cost functions. The resulting distributed control law is, however, not optimal as it differs from the one resulting from the original undiscounted Riccati equation. More importantly, the discounted controller has the potential to compromise closed-loop stability if the discounting factor is not carefully chosen. It has been shown in [29], [51] that the asymptotic output synchronization is achievable if the discounting factor satisfies a certain bound. However, determination

of this bound requires knowledge of the system dynamics, which is assumed to be unknown. Owing to these reasons, there exists a motivation to formulate a solution that eliminates the need of the discounting factor.

A common feature in the RL based control algorithms discussed so far is that they require the measurement of the complete internal state of the system. This is quite a restrictive assumption due to practical reasons such as cost and availability of sensors. Furthermore, the number of sensors increases with the dimensions of the system, which in turn increases the complexity of the control system. A technique called output feedback based control is often used in control systems to address this problem. Instead of requiring the measurement feedback of the internal state, the output feedback control makes use of only the system output measurements. If the system is observable, then the measurement feedback of the output together with the input can be used to reconstruct the system state. Thus, output feedback control eliminates the need of full state feedback and makes the design more practical, requiring fewer sensors and thereby making it cost-effective and reliable. Classical output feedback methods employ so called state observers to reconstruct the system state. However, these observers require the knowledge of the system model. As a result, output feedback in reinforcement learning is more challenging as the system dynamics is unknown.

Model-free output feedback approaches that have been reported in the RL control literature can be classified as neural network observers based and input-output data based methods. In [77], a suboptimal output feedback controller was presented for the LQR problem using a neural network observer with bounded estimation errors. In [19], [23], [43], [52], [72], [75], [76], neural networks were employed for state estimation, critic and actor networks to achieve near optimal solution for a class of nonlinear systems and ultimate boundedness of estimation errors was shown. In contrast to neural network based designs, the data based output feedback approach has attracted more attention recently [1], [21], [32], [38], [49]. This approach has the advantage that a separate state observer is not needed. As such, the results do not suffer from state estimation errors, resulting in an optimal control solution. In [1], an output feedback data based optimal controller was designed by the identification of Markov parameters using the input-output data. This idea was first used in the context of reinforcement learning LQR control in [38], where the VFA approach was taken to develop both PI and VI output feedback RL LQR algorithms. Following the VFA approach of [38], the authors of [32] solved the RL optimal linear quadratic tracking problem. Similarly, in [49], the VFA approach was used to solve the linear quadratic output feedback problem for

the continuous-time case. In all these works [32], [38], [49], a discounting factor is introduced in the cost function which helps to diminish the effect of excitation noise bias as studied in [38]. The end result is that the discounted controller is suboptimal and does not correspond to the solution of the Riccati equation. More importantly, this discounting factor has the potential to compromise system stability as reported recently in [49], [54]. The stability analysis carried out in [54] has shown that the discounted cost function cannot guarantee stability, in general. At best, stability can be achieved provided that the discounting factor is chosen above a lower bound. However, knowing this bound requires knowledge of the system dynamics, which is assumed to be unknown in RL ADP. Furthermore, the discounted controller loses optimality aspect of the control problem as it does not correspond to the optimal solution of the original ARE. In fact, the choice of this discounting factor can be too limited to achieve learning because of noise bias. Owing to these trade-offs, there is motivation to formulate a solution that does not require a discounting factor, in order to ensure stability and optimality.

Recently, some works have also employed RL to solve optimal control problems addressing more difficult challenges such as actuator saturation and time delays. The original work [3] employed the idea of nonquadratic cost functionals [45] to incorporate the actuator constraints and provided a model-based near optimal solution to the constraint HJB equation using neural networks. Later on, [28], [48], [50] extended this idea to solve the partially model-free constrained control problem online using reinforcement learning. A common feature in these works is that local stability could be guaranteed only in the form of uniform ultimate boundedness. Furthermore, the use of nonquadratic cost functional leads to nonlinear control laws even when the system dynamics are linear [44]. Time-delay is another frequently encountered difficulty in the control of dynamical systems. However, there are not much significant developments in the RL control framework on the topic of time-delays. Some recent methods that address the time-delay problem in the context of RL, require a special class of systems for which there exists a bicausal change of coordinates transformation, which is quite a restrictive assumption and is difficult to verify when the system model is unknown [74]. Therefore, more work is needed to address these practical challenges.

Based on the above literature survey, we present a list of the research objectives of this dissertation.

1) The development of model-free output feedback RL algorithms to relax the requirement of the measurement of the internal state.
2) Addressing the issue of estimation bias resulting from the use of exploration signals.
3) Uplifting the requirement of the discounting factor to ensure closed-loop stability and optimality

of the solution.

4) The design of model-free RL algorithms that can handle practical challenges such as disturbance rejection, actuator saturation and time delays.

1.4. Contributions

The aim of this research is to improve the applicability of the reinforcement learning algorithms by addressing some important practical challenges encountered in the design of feedback controllers for dynamical systems. The three main contributions of this research are,

1) This work will propose novel model-free output feedback optimal control algorithms to solve the optimal control problems for both discrete-time and continuous-time linear systems with completely unknown system dynamics.
2) The proposed algorithms will address the issues related to exploration bias and discounting factor, which have been the bottleneck in the earlier works in ensuring optimality and closed-loop stability.
3) This research will seek new algorithms which address some advanced control challenges such as disturbance rejection, actuator saturation and time delays in the setting of model-free optimal control.

Following is the list of publications upon which this dissertation has been drawn.

Book:

- S. A. A. Rizvi and Z. Lin, *Output Feedback Reinforcement Learning Control for Linear Systems.* Springer, *to be published in 2020*.

Book Chapter:

- S. A. A. Rizvi, Y. Wei and Z. Lin, "Reinforcement learning for optimal adaptive control of time delay systems," in *Handbook on Reinforcement Learning and Control.* D. H. Cansever, F. L. Lewis and Y. Wan, Eds. Springer, *to be published in 2020*.

Journal Papers:

- S. A. A. Rizvi and Z. Lin, "Output feedback adaptive dynamic programming for linear differential zero-sum games," *Automatica*, 2019, *under review*.
- S. A. A. Rizvi and Z. Lin, "Adaptive dynamic programming for model-free global stabilization of control constrained continuous-time systems," *IEEE Transactions on Cybernetics*, 2019, *under review*.

- S. A. A. Rizvi and Z. Lin, "An iterative Q-learning scheme for the global stabilization of discrete-time linear systems subject to actuator saturation," *International Journal of Robust and Nonlinear Control*, vol. 29, no. 9, pp. 2660–2672, 2019.
- S. A. A. Rizvi and Z. Lin, "Experience replay–based output feedback Q-learning scheme for optimal output tracking control of discrete-time linear systems," *International Journal of Adaptive Control and Signal Processing (Special Issue on Learning from Adaptive Control under Relaxed Excitation Conditions)*, vol. 33, no. 12, pp. 1825–1842, 2019.
- S. A. A. Rizvi and Z. Lin, "Output feedback reinforcement learning based optimal output synchronization of heterogeneous discrete-time multi-agent systems," *IET Control Theory & Applications (Special Issue on Distributed Optimisation and Learning for Networked Systems)*, vol. 13, no. 17, pp. 2866–2876, 2019.
- S. A. A. Rizvi and Z. Lin, "Reinforcement learning based linear quadratic regulation of continuous-time systems using dynamic output feedback," *IEEE Transactions on Cybernetics*, 2018, DOI: 10.1109/TCYB.2018.2886735.
- S. A. A. Rizvi and Z. Lin, "Output feedback reinforcement Q-learning control for the discrete-time linear quadratic regulator problem," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 5, pp. 1523–1536, 2018.
- S. A. A. Rizvi and Z. Lin, "Output feedback Q-learning for discrete-time linear zero-sum games with application to the H-infinity control," *Automatica*, vol. 95, pp. 213–221, 2018.

Conference Papers:

- S. A. A. Rizvi and Z. Lin, "Model-free global stabilization of continuous-time linear systems with saturating actuators using adaptive dynamic programming," *Proceedings of the 58th IEEE Conference on Decision and Control (CDC)*, pp. 145–150, 2019.
- S. A. A. Rizvi, Y. Wei and Z. Lin, "Model-free optimal stabilization of unknown time delay systems using adaptive dynamic programming," *Proceedings of the 58th IEEE Conference on Decision and Control (CDC)*, pp. 6536–6541, 2019.
- S. A. A. Rizvi and Z. Lin, "Model-free global stabilization of discrete-time linear systems with saturating actuators using reinforcement learning," *Proceedings of the 57th IEEE Conference on Decision and Control (CDC)*, pp. 5276–5281, 2018.
- S. A. A. Rizvi and Z. Lin, "Output feedback reinforcement learning control for the continuous-time

linear quadratic regulator problem," *Proceedings of the 2018 Annual American Control Conference (ACC)*, pp. 3417–3422, 2018.

- S. A. A. Rizvi and Z. Lin, "Output feedback optimal tracking control using reinforcement Q-Learning," *Proceedings of the 2018 Annual American Control Conference (ACC)*, pp. 3423–3428, 2018.
- S. A. A. Rizvi and Z. Lin, "Output feedback reinforcement Q-learning control for the discrete-time linear quadratic regulator problem," *Proceedings of the 56^{th} IEEE Conference on Decision and Control (CDC)*, pp. 1311–1316, 2017.

1.5. Dissertation Outline

This dissertation is divided into two parts. The first part presents the development of output feedback methods to solve the standard optimal control problems. We develop the dynamic output feedback learning equations for discrete-time and continuous-time linear dynamical systems. The focus of this part is to address the exploration bias issue and to develop output feedback methods that do not require discounted cost functions. These developments are essential in ensuring the closed-loop stability and optimality of the designed controller, and therefore, serve as the foundation of the more advanced output feedback control algorithms presented later in the dissertation. In the second part, we work towards developing output feedback algorithms that can handle various control challenges beyond the standard optimal control settings. These include the design of model-free controllers to tackle disturbances, actuator constraints and time delays in the feedback loop.

Chapter 2 presents the design of model-free output feedback RL algorithms to solve the LQR optimal stabilization problem. A Q-learning scheme is developed for discrete-time systems based on the parametrization of the system state in terms of the past input and output data. For the continuous-time systems, an integral reinforcement learning scheme is presented based on the filtered input and output data. The issues of discounting factor and exploration bias are addressed to overcome the difficulties with the existing works. It is shown that the proposed output feedback algorithms guarantee convergence to the optimal solution of the LQR Riccati equation.

Chapter 3 provides the design of model-free output feedback RL algorithms to solve the zero-sum game problem with application to the H-infinity control. The framework of game theory is employed to formulate the disturbance rejection problem as a two player zero-sum game. Output feedback Q-learning

and IRL algorithms are presented and convergence to the optimal solution is shown.

Chapter 4 deals with the design of model-free algorithms for the global stabilization of a class of systems subject to actuator saturation. The idea of gain-scheduled low gain feedback is applied to develop control laws that avoid saturation and achieve global stabilization. To design these control laws, we employ the framework of parameterized algebraic Riccati equations (AREs). Reinforcement learning techniques are developed for both continuous-time and discrete-time problems that find the solution of the parameterized ARE without requiring any knowledge of the system dynamics.

Chapter 5 focuses on the control problems involving time delays. The design of model-free RL controllers is presented based on an extended state augmentation approach. It is shown that discrete-time delay systems with input and/or state delays can be brought into a delay-free form. Extended state augmentation approach is presented to bring the system into a delay-free form. The controllability and observability conditions of the delay-free system are derived. Q-learning is then employed to learn the optimal control parameters for the extended dynamic system. Systems with arbitrary large unknown delays can be dealt with using the presented approach.

Chapter 6 concludes the dissertation and discusses some directions for future research.

2. MODEL-FREE OPTIMAL STABILIZATION

2.1. Introduction

Optimal stabilization refers to the problem of stabilizing a dynamical system based on some performance metric. For the case of linear systems, the cost function is traditionally described as a quadratic energy function reflecting the long term cost of the control and the states. For this reason, this problem is referred to as the linear quadratic regulation (LQR) problem.

In the following sections, we will present model-free reinforcement learning algorithms to solve the LQR optimal stabilization problem for linear systems using output feedback.

2.2. Q-learning Based Output Feedback LQR Control

Consider a discrete-time linear system given by the following state-space representation,

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k, \\ y_k &= Cx_k, \end{aligned} \tag{2.1}$$

where $x_k \in \mathbb{R}^n$ refers to the state, $u_k \in \mathbb{R}^m$ is the input, and $y_k \in \mathbb{R}^p$ is the output. Under the usual controllability and observability assumptions on (A, B) and (A, C), respectively, we would like to find the feedback control sequence $u_k = Kx_k$ that minimizes the long term cost,

$$J = \sum_{i=0}^{\infty} r(x_i, u_i), \tag{2.2}$$

where $r(x_k, u_k)$ is a quadratic function,

$$r(x_k, u_k) = x_k^{\mathrm{T}} C^{\mathrm{T}} Q_y C x_k + u_k^{\mathrm{T}} R u_k = y_k^{\mathrm{T}} Q_y y_k + u_k^{\mathrm{T}} R u_k, \tag{2.3}$$

where $Q_y \geq 0$ and $R > 0$ are the user-defined performance matrices. The optimal state feedback control that minimizes (2.2) is,

$$u_k^* = K^* x_k = -(R + B^{\mathrm{T}} P^* B)^{-1} B^T P^* A x_k, \tag{2.4}$$

and its associated value function is $V^*(x_k) = x_k^{\mathrm{T}} P^* x_k$, under the conditions of controllability of (A, B) and observability of $\left(A, \sqrt{Q}\right)$, where $\sqrt{Q}^{\mathrm{T}} \sqrt{Q} = Q$, $Q = C^{\mathrm{T}} Q_y C$. Here, $P^* = (P^*)^{\mathrm{T}}$ is the unique positive definite solution to the ARE,

$$A^{\mathrm{T}} P A - P + Q - A^{\mathrm{T}} P B (R + B^{\mathrm{T}} P B)^{-1} B^{\mathrm{T}} P A = 0. \tag{2.5}$$

In [38], a model-free output feedback solution to this problem has been proposed using the value function approximation (VFA) method based on the following Bellman learning equation corresponding to the value V_K of executing policy K,

$$V_K(x_k) = r(x_k, Kx_k) + V_K(x_{k+1}),$$

which holds when $u_k = Kx_k$. However, in order to keep the system excited, the control actually applied is $u_k = Kx_k + n_k$, where n_k is an exploration signal. The addition of this exploration signal violates this learning equation, and therefore, results in what is referred to as the exploration bias. This issue also exists in the output feedback counterpart of the above equation, in which the state x_k is replaced by input-output signals through a parametrization of the state [38]. To overcome this bias issue, generally a discounting factor $0 < \gamma < 1$ is introduced in the cost function as follows,

$$V_K(x_k) = \sum_{i=k}^{\infty} \gamma^{i-k} r(x_i, Kx_i). \tag{2.6}$$

This discounting factor helps diminish the long term effect of the exploration signal as shown in [38]. However, the use of the discounted cost function does not ensure closed-loop stability. Notice that for $\gamma < 1$, the discounted cost (2.6) is bounded,

$$\sum_{i=k}^{\infty} \gamma^{i-k} x_i^{\mathrm{T}} (Q + K^{\mathrm{T}} R K) x_i < \infty.$$

However, the boundedness of the cost does not ensure the convergence of $x_k \to 0$ as $k \to \infty$.

We now proceed to developing a Q-learning scheme without employing a discounted cost function. It will be shown that the controller parameters converge to the solution of the undiscounted ARE and the closed-loop stability is guaranteed. We will first present the concept of Q-functions which play a fundamental role in the design of the Q-learning algorithms that we will subsequently present. Consider the cost function given in (2.2). Under a feedback stabilizing control policy $u_k = Kx_k$ (not necessarily optimal), the total cost incurred when starting with any state x_k is quadratic in the state as given by,

$$V_K(x_k) = x_k^{\mathrm{T}} P x_k, P = P^{\mathrm{T}} > 0. \tag{2.7}$$

Motivated by the Bellman optimality principle, the undiscounted equation corresponding to (2.6) can be

written recursively as,

$$V_K(x_k) = r(x_k, Kx_k) + V_K(x_{k+1}), \tag{2.8}$$

where $V_K(x_{k+1})$ is the cost of following policy K in all future states.

Next, we use (2.8) to define a Q-function which is similar to the cost function but is explicit in both u_k and x_k as,

$$Q_K(x_k, u_k) = r(x_k, u_k) + V_K(x_{k+1}), \tag{2.9}$$

which is defined as the sum of the one-step cost of taking an arbitrary action u_k from state x_k plus the total cost that would incur if the policy K was followed from x_{k+1} and all subsequent states.

For the case of LQR, the Q-function can be represented as [16],

$$\begin{aligned} Q_K(x_k, u_k) &= x_k^{\mathrm{T}} Q x_k + u_k^{\mathrm{T}} R u_k + x_{k+1}^{\mathrm{T}} P x_{k+1} \\ &= x_k^{\mathrm{T}} Q x_k + u_k^{\mathrm{T}} R u_k + (Ax_k + Bu_k)^{\mathrm{T}} P (Ax_k + Bu_k), \end{aligned} \tag{2.10}$$

$$Q_K(x_k, u_k) = \begin{bmatrix} x_k \\ u_k \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} Q + A^{\mathrm{T}} P A & A^{\mathrm{T}} P B \\ B^{\mathrm{T}} P A & R + B^{\mathrm{T}} P B \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix}, \tag{2.11}$$

which is quadratic in both u_k and x_k.

Given the optimal cost V^*, we can compute K^*. To do so, we define the optimal Q-function as the cost of executing an arbitrary control u and then following the optimal policy K^* [38], [58], as given by,

$$Q^*(x_k, u_k) = r(x_k, u_k) + V^*(x_{k+1}), \tag{2.12}$$

and the optimal policy is given by,

$$K^* x_k = \arg\min_u (Q^*(x_k, u_k)).$$

The optimal LQR controller u_k^*, which minimizes the long term cost, can be obtained by solving $\frac{\partial}{\partial u_k} Q^* = 0$. The result is the same as given in (2.4), which was obtained by solving the ARE.

It can be seen that the Q-function that we obtained in (2.11) involves measurement of the internal state x_k which is not available in our problem setting. To tackle this difficulty, we now present some state reconstruction techniques that employ input and output data of the system to observe its internal state.

2.2.1. State Parametrization of Discrete-time MIMO Linear Systems:

Classical state estimation techniques make use of the model of the system to reconstruct the internal state by means of a state observer. A state observer is essentially a user-defined dynamical system which employs some error correction mechanism together with the knowledge of the system model and its input and output data to reconstruct the state. However, the problem of state estimation is not straight forward when the system model is not available.

A fundamental concept in adaptive control of unknown systems is to parameterize an equation in terms of the unknown parameters and the measurable variables, and using those measurable variables to estimate the unknown parameters [60]. The following result provides a parameterized expression of the state observer in terms of the measurable input and output data that will be used to derive the output feedback learning equations in the subsequent sections.

Theorem 2.1. *There exists a state parametrization,*

$$\bar{x}_k = W_u \sigma_k + W_y \omega_k, \tag{2.13}$$

that converges exponentially to the state x if the system is observable, where W_u and W_y are the system dependent matrices containing some transfer function coefficients, and σ_k and ω_k are the user-defined dynamics given by

$$\sigma_{k+1}^i = \mathcal{A}\sigma_k^i + \mathcal{B}u_k^i, \sigma_0^i = 0, i = 0, 1, \cdots, m$$
$$\omega_{k+1}^i = \mathcal{A}\omega_k^i + \mathcal{B}y_k^i, \omega_0^i = 0, i = 0, 1, \cdots, p,$$

for some Schur matrix $\mathcal{A}$ and input vector $\mathcal{B}$ of the form,

$$\mathcal{A} = \begin{bmatrix} -\alpha_{n-1} & -\alpha_{n-2} & \cdots & \cdots & -\alpha_0 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Proof: From the linear systems theory, it is known that if the pair (A, C) is observable then a full-state

observer can be constructed as,

$$\begin{aligned}\hat{x}_{k+1} &= A\hat{x}_k + Bu_k + L(y_k - C\hat{x}_k) \\ &= (A - LC)\hat{x}_k + Bu_k + Ly_k, \end{aligned} \tag{2.14}$$

where $\hat{x}_k$ is the estimate of the state x_k and L is the observer gain chosen such that the matrix $A - LC$ has all its eigenvalues strictly inside the unit circle. This observer can be considered a dynamic system driven by u_k and y_k with the dynamics matrix $A - LC$. This dynamic system can be written in the filter form by treating both u_k and y_k as inputs to the observer as given below,

$$\begin{aligned}\hat{x}_k &= (zI - A + LC)^{-1}B[u_k] + (zI - A + LC)^{-1}L[y_k] + (A - LC)^k\hat{x}_0 \\ &= \sum_{i=1}^{m} \frac{U_i(z)}{\Lambda(z)}[u_k^i] + \sum_{i=1}^{p} \frac{Y_i(z)}{\Lambda(z)}[y_k^i] + (A - LC)^k\hat{x}_0, \\ &= \frac{U(z)}{\Lambda(z)}[u_k] + \frac{Y(z)}{\Lambda(z)}[y_k] + (A - LC)^k\hat{x}_0, \end{aligned} \tag{2.15}$$

where u^i and y^i are the i^{th} input and output, respectively. Here, $U = \begin{bmatrix} U_1 & U_2 & \cdots & U_m \end{bmatrix}$, $Y = \begin{bmatrix} Y_1 & Y_2 & \cdots & Y_p \end{bmatrix}$, and $U_i(z)$ and $Y_i(z)$ are some n-dimensional polynomial vectors in the operator z, for example, $z^{-n}[u_k] = u_{k-n}$. Here the bracket notation $[u_k]$ is used to denote a time signal being operated by an operator represented by a transfer function to result in a new time signal. The resulting polynomial matrices $U(z)$ and $Y(z)$ depend on the quadruple (A, B, C, L). The characteristic polynomial $\Lambda(z)$ is given by

$$\begin{aligned}\Lambda(z) &= \det(zI - A + LC) \\ &= z^n + \alpha_{n-1}z^{n-1} + \alpha_{n-2}z^{n-2} + \cdots + \alpha_0. \end{aligned}$$

We now show that the filter form (2.15) can be linearly parameterized. Consider each input filter term $\frac{U_i(z)}{\Lambda(z)}[u_k^i]$ in (2.15),

$$\frac{U_i(z)}{\Lambda(z)}[u_k^i] = (zI - A + LC)^{-1}B_i[u_k^i],$$

where B_i is the i^{th} column of B and $\frac{U_i(z)}{\Lambda(z)}$ represents an n-dimensional polynomial vector, i.e., it contains

n filters each of degree n, which are applied to the input u^i. Therefore, we can express this term as,

$$\begin{aligned}\frac{U_i(z)}{\Lambda(z)}[u_k^i] &= \begin{bmatrix} \frac{a_{n-1}^{i1}z^{n-1}+a_{n-2}^{i1}z^{n-2}+\cdots+a_0^{i1}}{z^n+\alpha_{n-1}z^{n-1}+\alpha_{n-2}z^{n-2}+\cdots+\alpha_0} \\ \frac{a_{n-1}^{i2}z^{n-1}+a_{n-2}^{i2}z^{n-2}+\cdots+a_0^{i2}}{z^n+\alpha_{n-1}z^{n-1}+\alpha_{n-2}z^{n-2}+\cdots+\alpha_0} \\ \vdots \\ \frac{a_{n-1}^{in}z^{n-1}+a_{n-2}^{in}z^{n-2}+\cdots+a_0^{in}}{z^n+\alpha_{n-1}z^{n-1}+\alpha_{n-2}z^{n-2}+\cdots+\alpha_0} \end{bmatrix}[u_k^i] \\ &= \begin{bmatrix} a_{n-1}^{i1} & a_{n-2}^{i1} & \cdots & a_0^{i1} \\ a_{n-1}^{i2} & a_{n-2}^{i2} & \cdots & a_0^{i2} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n-1}^{in} & a_{n-2}^{in} & \cdots & a_0^{in} \end{bmatrix} \begin{bmatrix} \frac{z^{n-1}}{\Lambda(z)}[u_k^i] \\ \frac{z^{n-2}}{\Lambda(z)}[u_k^i] \\ \vdots \\ \frac{1}{\Lambda(z)}[u_k^i] \end{bmatrix} \\ &\triangleq W_u^i \sigma_k^i,\end{aligned}$$

where $W_u^i \in \mathbb{R}^{n\times n}$ is the parametric matrix corresponding to the coefficients of $U^i(z)$ for the input u^i. Notice that $\sigma_k^i \in \mathbb{R}^n$ contains the outputs of n filters when applied to the input u^i. Instead of applying n filters to u^i, we can also obtain σ^i using the n-dimensional state-space system driven by u^i as,

$$\sigma_{k+1}^i = \mathcal{A}\sigma_k^i + \mathcal{B}u_k^i,$$

where,

$$\mathcal{A} = \begin{bmatrix} -\alpha_{n-1} & -\alpha_{n-2} & \cdots & \cdots & -\alpha_0 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

This holds for every input u^i, $i = 0, 1, \cdots, m$, and thus, we can generate $\sigma_k = \begin{bmatrix} \sigma_k^1 & \sigma_k^2 & \cdots & \sigma_k^m \end{bmatrix}^T \in \mathbb{R}^{mn}$ with $W_u = \begin{bmatrix} W_u^1 & W_u^2 & \cdots & W_u^m \end{bmatrix} \in \mathbb{R}^{n\times mn}$. Similarly, we can repeat the same process for the output terms in (2.15). That is,

$$\omega_{k+1}^i = \mathcal{A}\omega_k^i + \mathcal{B}y_k^i,$$

which holds for every output y^i, $i = 0, 1, \cdots, p$, and thus, we can generate $\omega_k = \begin{bmatrix} \omega_k^1 & \omega_k^2 & \cdots & \omega_k^p \end{bmatrix}^T \in$

$\mathbb{R}^{pn}$ with $W_y = \begin{bmatrix} W_y^1 & W_y^2 & \cdots & W_y^p \end{bmatrix} \in \mathbb{R}^{n \times pn}$. We now show that the internal dynamics of σ_k and ω_k are asymptotically stable. The eigenvalues of $\mathcal{A}^i$ are the roots of its characteristic polynomial,

$$\det(zI - \mathcal{A}) = z^n + \alpha_{n-1} z^{n-1} + \alpha_{n-2} z^{n-2} + \cdots + \alpha_0.$$

Notice that the right-hand side equals $\Lambda(z)$, which is defined to be a stable polynomial. This implies that all the eigenvalues of $\mathcal{A}$ are strictly inside the unit circle, therefore, the internal dynamics of the σ_k and ω_k are asymptotically stable. Finally, by combining the input and output terms, we can write (2.15) as,

$$\hat{x}_k = W_u \sigma_k + W_y \omega_k + (A - LC)^k \hat{x}_0. \tag{2.16}$$

Note that $A - LC$ represents the observer error dynamics as known from the linear observer theory. That is,

$$e_k = x_k - \hat{x}_k = (A - LC)^k e_0,$$

which means that the original state x_k with the initial state x_0 can be parameterized as,

$$x_k = W_u \sigma_k + W_y \omega_k + (A - LC)^k x_0, \tag{2.17}$$

Since $A - LC$ is Schur stable, the term $(A - LC)^k$ in (2.16) and (2.17) vanishes. Thus, $\bar{x}_k - x_k \to 0$ as $k \to \infty$. This completes the proof. □

The state parametrization discussed in Theorem 2.1 is derived based on the Luenberger observer equation (2.14). Under this parametrization, the parameterized state (2.17) involves the observer error dynamics term $(A - LC)^k x_0$ which depends on the initial conditions. This observer error dynamics in turn depends on the choice of the design matrix L which is used to assign the eigenvalues of $A - LC$. The user-defined matrix $\mathcal{A}$ plays the role of the matrix $A - LC$ in the parametrization. Ideally, we would like the observer dynamics to be as fast as possible so that $\bar{x}_k$ converges to x_k quickly. For discrete-time systems, this can be achieved by placing all the eigenvalues of $A - LC$, or equivalently, the matrix $\mathcal{A}$ at 0. That is, the coefficients α_i of the matrix $\mathcal{A}$ are all chosen as zero. In this case, it can be verified that $(A - LC)^k x_0$ vanishes exactly in n time steps independent of the unknown initial condition x_0. Motivated by this property of discrete-time system, recently a special case of the above parametrization was presented in [38]. The special state parametrization result is recalled below.

Theorem 2.2. *Under the assumptions of the observability of the pair (A, C), the system state can be*

represented in terms of the measured input and output sequence as,

$$x_k = M_y \bar{y}_{k-1,k-N} + M_u \bar{u}_{k-1,k-N}, \tag{2.18}$$

where $N \leq n$ is an upper bound on the system's observability index, $\bar{u}_{k-1,k-N} \in \mathbb{R}^{mN}$ and $\bar{y}_{k-1,k-N} \in \mathbb{R}^{pN}$ are the past input and output data vectors defined as,

$$\bar{u}_{k-1,k-N} = \begin{bmatrix} u_{k-1}^T & u_{k-2}^T & \cdots & u_{k-N}^T \end{bmatrix}^T, \bar{y}_{k-1,k-N} = \begin{bmatrix} y_{k-1}^T & y_{k-2}^T & \cdots & y_{k-N}^T \end{bmatrix}^T,$$

$$M_y = A^N (V_N^T V_N)^{-1} V_N^T, M_u = U_N - A^N (V_N^T V_N)^{-1} V_N^T T_N,$$

with

$$V_N = \begin{bmatrix} (CA^{N-1})^T & \cdots & (CA)^T & C^T \end{bmatrix}^T, U_N = \begin{bmatrix} B & AB & \cdots & A^{N-1}B \end{bmatrix},$$

$$T_N = \begin{bmatrix} 0 & CB & CAB & \cdots & CA^{N-2}B \\ 0 & 0 & CB & \cdots & CA^{N-3}B \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & CB \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

We now proceed to apply the state parametrization (2.18) to describe the Q-function in (2.11) in terms of the system's inputs and outputs. It can be easily verified that the substitution of (2.18) in (2.11) results in,

$$Q_K = \begin{bmatrix} \bar{u}_{k-1,k-N} \\ \bar{y}_{k-1,k-N} \\ u_k \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} H_{\bar{u}\bar{u}} & H_{\bar{u}\bar{y}} & H_{\bar{u}u} \\ H_{\bar{y}\bar{u}} & H_{\bar{y}\bar{y}} & H_{\bar{y}u} \\ H_{u\bar{u}} & H_{u\bar{y}} & H_{uu} \end{bmatrix} \begin{bmatrix} \bar{u}_{k-1,k-N} \\ \bar{y}_{k-1,k-N} \\ u_k \end{bmatrix} \triangleq z_k^{\mathrm{T}} H z_k, \tag{2.19}$$

where $z_k = \begin{bmatrix} \bar{u}_{k-1,k-N}^{\mathrm{T}} & \bar{y}_{k-1,k-N}^{\mathrm{T}} & u_k^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}, H = H^{\mathrm{T}} \in \mathbb{R}^{(mN+pN+m)\times(mN+pN+m)}$ and the partitioned

matrices are defined as,

$$\begin{aligned}
H_{\bar{u}\bar{u}} &= M_u^{\mathrm{T}}(Q + A^{\mathrm{T}}PA)M_u \in \mathbb{R}^{mN\times mN},\\
H_{\bar{u}\bar{y}} &= M_u^{\mathrm{T}}(Q + A^{\mathrm{T}}PA)M_y \in \mathbb{R}^{mN\times pN},\\
H_{\bar{u}u} &= M_u^{\mathrm{T}}A^{\mathrm{T}}PB \in \mathbb{R}^{mN\times m},\\
H_{\bar{y}\bar{y}} &= M_y^{\mathrm{T}}(Q + A^{\mathrm{T}}PA)M_y \in \mathbb{R}^{pN\times pN}\\
H_{\bar{y}u} &= M_y^{\mathrm{T}}A^{\mathrm{T}}PB \in \mathbb{R}^{pN\times m},\\
H_{uu} &= R + B^{\mathrm{T}}PB \in \mathbb{R}^{m\times m}.
\end{aligned} \tag{2.20}$$

Given the optimal cost function V^* with the cost matrix P^*, we obtain the corresponding optimal output feedback matrix H^* by substituting $P = P^*$ in (2.20). Then the optimal output feedback Q-function is given by,

$$Q^* = z_k^T H^* z_k, \tag{2.21}$$

which provides a state-free representation of the LQR Q-function. Minimizing Q^* with respect to u_k results in the desired control law,

$$u_k^* = -(H_{uu}^*)^{-1}\left(H_{u\bar{u}}^*\bar{u}_{k-1,k-N} + H_{u\bar{y}}^*\bar{y}_{k-1,k-N}\right). \tag{2.22}$$

Lemma 2.1. *The output feedback controller given by (2.22) is the optimal controller that solves the LQR problem (2.1–2.3).*

Proof: By Theorem 2.2, we know that the state vector can be represented by input-output sequence as in (2.18). It can be easily seen that substituting (2.18) and (2.20) in the control law (2.22) results in the state feedback controller (2.4) which is the optimal LQR controller. So, the output feedback controller is the equivalent optimal controller that solves the LQR problem. This completes the proof. □

We seek an output feedback form of Q-learning to learn the optimal Q-function. The LQR Q-learning equation [16] is,

$$Q_K(x_k, u_k) = x_k^{\mathrm{T}}Qx_k + u_k^{\mathrm{T}}Ru_k + Q_K(x_{k+1}, Kx_{k+1}), \tag{2.23}$$

The parameterized output feedback Q-learning equation as follows,

$$\bar{H}^{\mathrm{T}}\bar{z}_k = y_k^{\mathrm{T}}Q_y y_k + u_k^{\mathrm{T}}Ru_k + \bar{H}^{\mathrm{T}}\bar{z}_{k+1}. \tag{2.24}$$

where $\bar{H} = \mathrm{vec}(H) \in \mathbb{R}^{l(l+1)/2} \triangleq [h_{11}, 2h_{12}, \cdots, 2h_{1l}, h_{22}, 2h_{23}, \cdots, 2h_{2l}, \cdots h_{ll}]^{\mathrm{T}}$,

$\bar{z} = \left[z_1^2, z_1 z_2, \cdots, z_1 z_l, z_2^2, z_2 z_3, \cdots, z_2 z_l, \cdots z_l^2\right]^{\mathrm{T}}$ and $l = mN + pN + m$.

Remark 2.1. *The parametrization matrices M_u and M_y in* (2.18) *are the same as W_u and W_y in* (2.13) *with $\bar{u}_{k-1,k-N} = \sigma_k$ and $\bar{y}_{k-1,k-N} = \omega_k$ when $\alpha_i's$ are zero and $N = n$. In this case, the subscripts $\bar{u}$ and $\bar{y}$ in the matrix H are replaced by σ and ω, respectively.*

We can now utilize reinforcement learning to learn our output feedback Q-function matrix.

Algorithm 1: Output Feedback Q-learning Value Iteration for LQR Control

Initialization. Select an arbitrary policy u_k^0 with $\bar{H}^0 = 0$. Then, for $j = 1, 2, \cdots$, perform until,

$$\left\|\bar{H}^j - \bar{H}^{j-1}\right\| < \varepsilon :$$

Value Update. Evaluate the cost of the current policy,

$$\left(\bar{H}^j\right)^{\mathrm{T}} \bar{z}_k = y_k^{\mathrm{T}} Q_y y_k + u_k^{\mathrm{T}} R u_k + \left(\bar{H}^{j-1}\right)^{\mathrm{T}} \bar{z}_{k+1}.$$

Policy Improvement. Determine an updated policy u_k^{j+1},

$$u_k^{j+1} = -\left(H_{uu}^j\right)^{-1} \left(H_{u\bar{u}}^j \bar{u}_{k-1,k-N} + H_{u\bar{y}}^j \bar{y}_{k-1,k-N}\right).$$

The above algorithm presents the VI algorithm for the output feedback Q-learning LQR, which is essentially a two-step procedure. In the value update step, we use the key equation (2.24) to solve for the unknown vector $\bar{H}$ in the least-squares sense by collecting $L \geq l(l+1)/2$ data samples of $u_k, y_k, \bar{u}_{k-1,k-N}, \bar{y}_{k-1,k-N}, \bar{u}_{k,k-N+1}$ and $\bar{y}_{k,k-N+1}$. Note that these L data samples refer to different points in time, k, which are used to form the data matrices $\Phi \in \mathbb{R}^{l(l+1)/2 \times L}$ and $\Upsilon^{j-1} \in \mathbb{R}^{L \times 1}$ defined by, the data matrices $\Phi \in \mathbb{R}^{l(l+1)/2 \times L}$ and $\Upsilon^{j-1} \in \mathbb{R}^{L \times 1}$ are defined by $\Phi = \left[\bar{z}_k^1, \bar{z}_k^2, \cdots, \bar{z}_k^L\right]$, $\Upsilon^{j-1} = \left[r^1(y_k, u_k) + \left(\bar{H}^{j-1}\right)^{\mathrm{T}} \bar{z}_{k+1}^1, \cdots, r^L(y_k, u_k) + \left(\bar{H}^{j-1}\right)^{\mathrm{T}} \bar{z}_{k+1}^L\right]^{\mathrm{T}}$.

Then, the least-squares solution of (2.24) is given by,

$$\bar{H}^j = (\Phi \Phi^{\mathrm{T}})^{-1} \Phi \Upsilon^{j-1}, \tag{2.25}$$

where $\bar{H}^j$ is the j^{th} estimate of $\bar{H}$ corresponding to the cost of the current policy u_k^j. In the policy improvement step, we compute the new policy u_k^{j+1} that is used in the next $j+1$ iteration. Notice in

(2.22) that the control u_k is linearly dependent on $\bar{u}_{k-1,k-N}$ and $\bar{y}_{k-1,k-N}$ which means that $\Phi\Phi^{\mathrm{T}}$ will not be invertible. To overcome this issue, one adds excitation noise in u_k which guarantees a unique solution to (2.25). In other words, the following condition needs to be satisfied,

$$\text{rank}(\Phi) = l(l+1)/2. \tag{2.26}$$

This excitation condition can be met in several ways such as adding sinusoidal noise of various frequencies, exponentially decaying noise and Gaussian noise.

The convergence criterion for the algorithm is to check on the updates in the estimated vector $\bar{H}^j$. This can be done by the following condition,

$$\left\|\bar{H}^j - \bar{H}^{j-1}\right\| < \varepsilon, \tag{2.27}$$

where $\varepsilon > 0$ is some small positive constant.

Theorem 2.3. *Suppose that (A, B) is controllable, $\left(A, \sqrt{Q_y}C\right)$ is observable. Then, the output feedback VI algorithm generates a sequence of controls $\{u_k^j, j = 1, 2, 3, ...\}$ that converges to the optimal output feedback controller given in (2.22) as $j \to \infty$ if the rank condition* (2.26) *holds.*

Proof: The convergence proof for VI based Q-learning LQR has been given in [35] for the state feedback case. Therefore, having shown the equivalence of the state and output feedback controller as discussed in Theorem 2.2, we can conclude that, under the unique solvability of the value update equation (ensured by the rank condition (2.26)), the output feedback Q-learning controller based on VI algorithm converges to the optimal controller as $j \to \infty$. This completes the proof. □

Exploration Bias Immunity of Algorithm 1:

We next show that the proposed Q-learning scheme does not lead to bias in the Q-function estimates.

Theorem 2.4. *The excitation noise required to satisfy the excitation condition does not result in any bias in Q-function estimates. This applies to both state feedback and output feedback Q-functions.*

Proof: Consider first the case of state feedback Q-learning. Under the excitation noise, we can write the Q-function as

$$Q_K(x_k, \hat{u}_k) = r(x_k, \hat{u}_k) + V_K(x_{k+1}),$$

where $\hat{u}_k = u_k + n_k$ with n_k being the excitation noise. Let $\hat{H}'$ be the estimate of H' obtained using the input $\hat{u}$. It then follows from (2.11) that

$$\begin{bmatrix} x_k \\ \hat{u}_k \end{bmatrix}^{\mathrm{T}} \hat{H}' \begin{bmatrix} x_k \\ \hat{u}_k \end{bmatrix} = r(x_k, \hat{u}_k) + (Ax_k + B\hat{u}_k)^{\mathrm{T}} P (Ax_k + B\hat{u}_k).$$

Upon further expansion, we have,

$$\begin{bmatrix} x_k \\ u_k \end{bmatrix}^{\mathrm{T}} \hat{H}' \begin{bmatrix} x_k \\ u_k \end{bmatrix} + x_k^{\mathrm{T}} A^{\mathrm{T}} P B n_k + n_k^{\mathrm{T}} B^{\mathrm{T}} P A x_k + n_k^{\mathrm{T}} (R + B^{\mathrm{T}} P B) u_k + u_k^{\mathrm{T}} (R + B^{\mathrm{T}} P B) n_k + n_k^{\mathrm{T}} (R + B^{\mathrm{T}} P B) n_k$$
$$= x_k^{\mathrm{T}} Q x_k + u_k^{\mathrm{T}} R u_k + n_k^{\mathrm{T}} R n_k + n_k^{\mathrm{T}} R u_k + u_k^{\mathrm{T}} R n_k + (Ax_k + Bu_k)^{\mathrm{T}} P (Ax_k + Bu_k) + (Ax_k + Bu_k)^{\mathrm{T}} P B n_k$$
$$+ (Bn_k)^{\mathrm{T}} P (Ax_k + Bu_k) + n_k B^{\mathrm{T}} P B n_k.$$

It can be easily verified that all the terms involving n_k get canceled on both sides of the equation, and we are left with

$$\begin{bmatrix} x_k \\ u_k \end{bmatrix}^{\mathrm{T}} \hat{H}' \begin{bmatrix} x_k \\ u_k \end{bmatrix} = r(x_k, u_k) + (Ax_k + Bu_k)^{\mathrm{T}} P (Ax_k + Bu_k).$$

Comparing the above equation with (2.10), we have $\hat{H}' = H'$, that is,

$$Q_K(x_k, u_k) = r(x_k, u_k) + V_K(x_{k+1}).$$

In view of (2.23) and (2.19), we have,

$$z_k^{\mathrm{T}} H z_k = y_k^{\mathrm{T}} Q_y y_k + u_k^{\mathrm{T}} R u_k + z_{k+1}^{\mathrm{T}} H z_{k+1}.$$

Hence, we get the bias free output feedback Bellman equation. This completes the proof. □

2.3. Integral Reinforcement Learning Based Output Feedback LQR Control

The discussion so far focused on the discrete-time dynamical systems as reinforcement learning algorithms were originally developed for such systems. The treatment of continuous-time problems in reinforcement learning is quite different from the discrete-time problems discussed in the previous section. One of the notable difficulties in the continuous-time case arise due to the involvement of the derivatives of the value function in the Bellman equation, which requires knowledge of the system dynamics, as

given by the following equation,

$$0 = r(x(t), Kx(t)) + (\Delta V)^{\mathrm{T}}(Ax(t) + Bu(t)).$$

In contrast, the Bellman equation for the discrete-time problems (2.8) is a recursion between two consecutive values of the value function and does not involve the system dynamics. Recently, the idea of integral reinforcement learning (IRL) [64] has been used to overcome this difficulty. The IRL Bellman equation is,

$$V(x(t)) = \int_t^{t+T} r(x(\tau), Kx(\tau))d\tau + V(x(t+T)).$$

The idea of using interval integrals in the learning equation has been successfully used to design RL and ADP based model-free control algorithms [14], [27] and will be adopted in our work to solve continuous-time control problems.[1]

To this end, we now work towards designing output feedback RL based LQR controller for continuous-time systems. Consider a continuous-time linear time-invariant system in the state-space form,

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t), \end{aligned} \tag{2.28}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the control input, and $y \in \mathbb{R}^p$ is the output. Under the usual controllability and observability assumptions on (A, B) and (A, C), respectively, we are to solve the LQR problem by dynamic output feedback. We will adopt an observer based controller structure. When the full state is available for feedback, the control law takes the form,

$$u^*(t) = K^* x(t) \tag{2.29}$$

that minimizes the cost function of the form

$$J = \int_0^{\infty} r(x(\tau), u(\tau))d\tau \tag{2.30}$$

where $r(x, u)$ is the utility function. In our LQR problem, the utility function $r(x, u)$ takes a quadratic form

$$r = x^{\mathrm{T}}(t)C^{\mathrm{T}}Q_y Cx(t) + u^{\mathrm{T}}(t)Ru(t) = y^{\mathrm{T}}(t)Q_y y(t) + u^{\mathrm{T}}(t)Ru(t), \tag{2.31}$$

[1] Although the terminology integral (interval) reinforcement learning is often used to refer to the partially model-free algorithms based on the work [64], we will use the same terminology for the completely model-free algorithms presented in this work.

where $Q_y \geq 0$ and $R > 0$ are the user-defined weighting matrices. Under a stabilizing policy $u = Kx$ (not necessarily optimal), the value function is quadratic in the state,

$$V(x) = x^{\mathrm{T}}(t)Px(t), \tag{2.32}$$

where $P = P^{\mathrm{T}} > 0$. Under the conditions of controllability of (A, B) and observability of $\left(A, \sqrt{Q}\right)$, where $\sqrt{Q}^{\mathrm{T}}\sqrt{Q} = Q$, $Q = C^{\mathrm{T}}Q_yC$, there exists a unique optimal feedback gain given by,

$$K^* = -R^{-1}B^{\mathrm{T}}P^*, \tag{2.33}$$

where $P^* = P^{*\mathrm{T}} > 0$ is the unique positive definite solution to the ARE [37],

$$A^{\mathrm{T}}P + PA + Q - PBR^{-1}B^{\mathrm{T}}P = 0. \tag{2.34}$$

A model-free output feedback solution for the continuous-time LQR problem was recently proposed in [49], where the authors used the N-sample observability theorem [65] to reconstruct the state of a continuous-time system by sampling delayed output measurements. It has been shown in [49] that if the control is of the full state feedback form, $u = Kx$, then the state of the closed-loop system $\dot{x} = (A + BK)x$ is related to the delayed output as,

$$y(t - iT) = Ce^{-iT(A+BK)}x(t),$$

where $i = 0, 1, \cdots, N-1$ and T is the sample interval. Here, N is the minimum number of samples needed to observe the system in some interval (t_i, t_f), which depends on the system dynamics. Thus, N delayed output measurements can be used to reconstruct the state as,

$$x(t) = (G^{\mathrm{T}}G)^{-1}G^{\mathrm{T}}\bar{y}(t),$$

where,

$$\begin{aligned} \bar{y}(t) &= \Big[y^{\mathrm{T}}(t)\; y^{\mathrm{T}}(t-T) \cdots y^{\mathrm{T}}(t-(N-1)T)\Big]^{\mathrm{T}}, \\ G &= \Big[C^{\mathrm{T}}\; (e^{-T(A+BK)})^{\mathrm{T}}C^{\mathrm{T}} \cdots (e^{-(N-1)T(A+BK)})^{\mathrm{T}}C^{\mathrm{T}}\Big]^{\mathrm{T}}. \end{aligned}$$

The method is elegant, however, it assumes that $u = Kx$, and therefore, it does not consider the excitation noise in the input. Thus, introducing excitation noise violates the above relations, which ultimately leads

to bias in the parameter estimates. As a result, the estimated control parameters converge to sub-optimal parameters [49]. To address this problem, a discounting factor γ is typically introduced in the cost function [38], [49], which helps to suppress the noise bias. In the continuous-time case, the discounted value function is given by,

$$V(x) = \int_t^\infty e^{-\gamma(\tau-t)} r(x(\tau), u(\tau)) d\tau. \tag{2.35}$$

The addition of the discounting factor γ, however, changes the solution of the Riccati equation, and the resulting discounted control is no longer optimal. More importantly, the introduction of the discounting factor does not guarantee closed-loop stability as the original optimal control (2.29) would. To see this, we note that for $\gamma > 0$, $\int_t^\infty e^{-\gamma(\tau-t)} x^{\mathrm{T}}(\tau)(Q + K^{\mathrm{T}}RK)x(\tau)d\tau < \infty$ does not guarantee $x(t) \to 0$ as $t \to \infty$.

In the following, we will develop output feedback algorithms that do not suffer from the exploration bias problem and as a result do not require discounted cost functions. The development of such output feedback algorithms will enable to guarantee optimality and close-loop stability of the controller. To this end, we will present a different parametrization of the state of a continuous-time system that can take into account the exploration component of the input signal.

2.3.1. State Parametrization of Continuous-time MIMO Linear Systems:

In the discrete-time case, we were fortunate to obtain an exact parametrization of the state in terms of the past input-output data, as given in Theorem 2.2. This was made possible by placing the eigenvalues of the observer at the origin. However, the direct translation of this discrete-time result into the continuous-time case would invoke the derivatives of the input and output measurements, which is generally prohibitive in practice. To overcome this difficulty, [49] proposed the parametrization based on delayed output measurements, which does not include the effect of exploration noise component present in the input, as shown in the previous section. Interestingly, the general MIMO parametrization result derived based on the Luenberger observer theory in Theorem 2.1 the for the discrete-time case can be extended to continuous-time case. This was the motivation of introducing the general parametrization (2.13) before its special case (2.17). In the following, we present a state parametrization procedure to represent the state in terms of the filtered inputs and outputs for general continuous-time MIMO linear systems.

Theorem 2.5. *There exists a state parametrization*

$$\bar{x}(t) = M_u \zeta_u(t) + M_y \zeta_y(t), \tag{2.36}$$

that converges exponentially to the state x if the system is observable, where M_u and M_y are the system dependent matrices containing some transfer function coefficients, and ζ_u and ζ_y are the user-defined dynamics given by

$$\dot{\zeta_u^i}(t) = \mathcal{A}\zeta_u^i(t) + \mathcal{B}u_i(t),\ \zeta_u^i(0) = 0, \forall i = 1, 2, \cdots, m$$
$$\dot{\zeta_y^i}(t) = \mathcal{A}\zeta_y^i(t) + \mathcal{B}y_i(t),\ \zeta_y^i(0) = 0, \forall i = 1, 2, \cdots, p,$$

for some Hurwitz matrix $\mathcal{A}$ and input vector $\mathcal{B}$ of the form,

$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -\alpha_0 & -\alpha_1 & \cdots & \cdots & -\alpha_{n-1} \end{bmatrix},\ \mathcal{B} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

Proof: Under the observability of (A, C), the estimate of the state given by $\hat{x}$ can be obtained based on the following observer,

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) + L(y(t) - C\hat{x}(t)) \\ &= (A - LC)\hat{x}(t) + Bu(t) + Ly(t), \end{aligned} \tag{2.37}$$

where L is the user-defined observer gain selected such that the matrix $A - LC$ is Hurwitz. The system inputs and outputs serve as the inputs to the dynamical system $A - LC$ of the observer, which can be written in a filter notation as follows,

$$\begin{aligned} \hat{x}(t) &= (sI - A + LC)^{-1}B[u] + (sI - A + LC)^{-1}L[y] \\ &\quad + e^{(A-LC)t}\hat{x}(0) \\ &= \sum_{i=1}^{m} \frac{U_i(s)}{\Lambda(s)}[u_i] + \sum_{i=1}^{p} \frac{Y_i(s)}{\Lambda(s)}[y_i] + e^{(A-LC)t}\hat{x}(0), \\ &= \frac{U(s)}{\Lambda(s)}[u] + \frac{Y(s)}{\Lambda(s)}[y] + e^{(A-LC)t}\hat{x}(0), \end{aligned} \tag{2.38}$$

where u_i and y_i are the components of the input and output vectors, respectively. Moreover, $U = \begin{bmatrix} U_1 & U_2 & \cdots & U_m \end{bmatrix}$, $Y = \begin{bmatrix} Y_1 & Y_2 & \cdots & Y_p \end{bmatrix}$, and $U_i(s)$ and $Y_i(s)$ are some n-dimensional polynomial vectors in the differential operator s, for example, $s[u] = \frac{d}{dt}u$. We represent $[u]$ as a time signal acted upon by some transfer function, which results in another time signal. Note that $U(s)$ and $Y(s)$ are dependent on (A, B, C, L). We define $\Lambda(s)$ as the characteristic polynomial corresponding to $A - LC$ as,

$$\begin{aligned}\Lambda(s) &= \det(sI - A + LC) \\ &= s^n + \alpha_{n-1}s^{n-1} + \alpha_{n-2}s^{n-2} + \cdots + \alpha_1 s + \alpha_0.\end{aligned}$$

Note that $\frac{U_i(s)}{\Lambda(s)}[u_i]$ in (2.38) is given by,

$$\frac{U_i(s)}{\Lambda(s)}[u_i] = (sI - A + LC)^{-1}B_i[u_i],$$

where B_i is the i^{th} column of B. The above equation can be further expanded as follows,

$$\begin{aligned}\frac{U_i(s)}{\Lambda(s)}[u_i] &= \begin{bmatrix} \frac{a_{n-1}^{i1}s^{n-1}+a_{n-2}^{i1}s^{n-2}+\cdots+a_0^{i1}}{s^n+\alpha_{n-1}s^{n-1}+\alpha_{n-2}s^{n-2}+\cdots+\alpha_0} \\ \frac{a_{n-1}^{i2}s^{n-1}+a_{n-2}^{i2}s^{n-2}+\cdots+a_0^{i2}}{s^n+\alpha_{n-1}s^{n-1}+\alpha_{n-2}s^{n-2}+\cdots+\alpha_0} \\ \vdots \\ \frac{a_{n-1}^{in}s^{n-1}+a_{n-2}^{in}s^{n-2}+\cdots+a_0^{in}}{s^n+\alpha_{n-1}s^{n-1}+\alpha_{n-2}s^{n-2}+\cdots+\alpha_0} \end{bmatrix}[u_i] \\ &= \begin{bmatrix} a_0^{i1} & a_1^{i1} & \cdots & a_{n-1}^{i1} \\ a_0^{i2} & a_1^{i2} & \cdots & a_{n-1}^{i2} \\ \vdots & \vdots & \ddots & \vdots \\ a_0^{in} & a_1^{in} & \cdots & a_{n-1}^{in} \end{bmatrix} \begin{bmatrix} \frac{1}{\Lambda(s)}[u_i] \\ \frac{s}{\Lambda(s)}[u_i] \\ \vdots \\ \frac{s^{n-1}}{\Lambda(s)}[u_i] \end{bmatrix} \\ &\triangleq M_u^i \zeta_u^i(t),\end{aligned}$$

where the parametrization matrix $M_u^i \in \mathbb{R}^{n\times n}$ contains the coefficients of the polynomial vector $U^i(s)$. $\zeta_u^i \in \mathbb{R}^n$ is the result of n filtering operations on the signal u_i, which can also be obtained using the following the dynamical system,

$$\dot{\zeta_u^i}(t) = \mathcal{A}\zeta_u^i(t) + \mathcal{B}u_i(t),$$

where,

$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -\alpha_0 & -\alpha_1 & \cdots & \cdots & -\alpha_{n-1} \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

Thus, for the complete input vector u, we can obtain $\zeta_u = \begin{bmatrix} (\zeta_u^1)^{\mathrm{T}} & (\zeta_u^2)^{\mathrm{T}} & \cdots & (\zeta_u^m)^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{mn}$ with $M_u = \begin{bmatrix} M_u^1 & M_u^2 & \cdots & M_u^m \end{bmatrix} \in \mathbb{R}^{n \times mn}$. The same procedure can also be applied for the output vector, as follows,

$$\dot{\zeta}_y^i(t) = \mathcal{A}\zeta_y^i(t) + \mathcal{B}y_i(t),$$

Note that the characteristic polynomial of $\mathcal{A}$,

$$\det(sI - \mathcal{A}) = s^n + \alpha_{n-1}s^{n-1} + \alpha_{n-2}s^{n-2} + \cdots + \alpha_0.$$

is the same as $\Lambda(s)$, which is defined to be a stable polynomial, and thus, the above dynamical systems driven by u and y are also asymptotically stable. Based on the above relations, (2.38) can be written as,

$$\hat{x}(t) = M_u\zeta_u(t) + M_y\zeta_y(t) + e^{(A-LC)t}\hat{x}(0). \tag{2.39}$$

It should be noted that $A - LC$ represents the error dynamics of the observer,

$$e(t) = x(t) - \hat{x}(t) = e^{(A-LC)t}e(0),$$

and therefore x can be written as,

$$x(t) = M_u\zeta_u(t) + M_y\zeta_y(t) + e^{(A-LC)t}x(0). \tag{2.40}$$

It can be seen that $e^{(A-LC)t}$ in (2.39) and (2.40) converges to zero because $A - LC$ is Hurwitz stable. Hence, $\bar{x} - x \to 0$ as $t \to \infty$. This completes the proof. □

Remark 2.2. *The parametrization matrices M_u and M_y depend on the quadruple (A, B, C, L), and therefore, require the knowledge of system dynamics. It will be shown that by embedding these matrices in the learning equation, we will be able to directly learn the optimal output feedback policy without*

actually computing these matrices, which makes the design model-free.

Next, we aim to use this state parametrization to describe the value function in (2.32). Substitution of (2.36) in (2.32) results in

$$V = \begin{bmatrix} \zeta_u \\ \zeta_y \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} M_u & M_y \end{bmatrix}^{\mathrm{T}} P \begin{bmatrix} M_u & M_y \end{bmatrix} \begin{bmatrix} \zeta_u \\ \zeta_y \end{bmatrix} \triangleq z^{\mathrm{T}} \bar{P} z, \tag{2.41}$$

where

$$z = \begin{bmatrix} \zeta_u^{\mathrm{T}} & \zeta_y^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{(mn+pn)},$$
$$\bar{P} = \bar{P}^{\mathrm{T}} = \begin{bmatrix} M_u^{\mathrm{T}} P M_u & M_u^{\mathrm{T}} P M_y \\ M_y^{\mathrm{T}} P M_u & M_y^{\mathrm{T}} P M_y \end{bmatrix} \in \mathbb{R}^{(mn+pn)\times(mn+pn)}.$$

By (2.41) we have obtained an approximation of the value function in terms of the filtered inputs and outputs of the system. The corresponding output feedback controller is given by,

$$u = Kx = K \begin{bmatrix} M_u & M_y \end{bmatrix} \begin{bmatrix} \zeta_u^{\mathrm{T}} & \zeta_y^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \triangleq \bar{K} z, \tag{2.42}$$

where $\bar{K} = K \begin{bmatrix} M_u & M_y \end{bmatrix} \in \mathbb{R}^{m\times(mn+pn)}$. Therefore, the optimal cost matrix is given by $\bar{P}^* = (\bar{P}^*)^{\mathrm{T}}$ and the corresponding optimal output feedback controller is

$$u^* = \bar{K}^* z. \tag{2.43}$$

Lemma 2.2. *The output feedback controller given by (2.43) converges to the optimal LQR controller.*

Proof: Consider the optimal output feedback LQR controller (2.43). Substituting $\bar{K}^* = K^* \begin{bmatrix} M_u & M_y \end{bmatrix}$ in (2.43) results in

$$u^*(z) = K^*(M_u \zeta_u(t) + M_y \zeta_y(t)).$$

By Theorem 2.5, we have the convergence of $\bar{x}(t) = M_u \zeta_u(t) + M_y \zeta_y(t)$ to $x(t)$, and hence, (2.43) converges to

$$u^*(x) = K^* x(t).$$

This completes the proof. □

The discussion so far focused on finding the nominal solution of the continuous-time LQR problem

when the system model is known. In the following, we will derive an output feedback learning equation that will allow us to learn the optimal output feedback LQR controller based on the filtered input and output data instead of using full state feedback. To this end, consider the following Lyapunov function candidate,

$$V = x^{\mathrm{T}} P x. \tag{2.44}$$

Evaluating the derivative of (2.44) along the system trajectories,

$$\begin{aligned}\frac{d}{dt}(x^{\mathrm{T}}(t)Px(t)) &= (Ax(t)+Bu(t))^{\mathrm{T}}Px(t) + x^{\mathrm{T}}(t)P(Ax(t)+Bu(t)) \\ &= x^{\mathrm{T}}(t)Hx(t) - 2(Ru(t))^{\mathrm{T}}Kx(t),\end{aligned}$$

where $H = A^{\mathrm{T}}P + P^{\mathrm{T}}A$ and $K = -R^{-1}B^{\mathrm{T}}P$. Performing finite window integrals of length $T > 0$ on both sides results in the following equation,

$$x^{\mathrm{T}}(t)Px(t) - x^{\mathrm{T}}(t-T)Px(t-T) = \int_{t-T}^{t} x^{\mathrm{T}}(\tau)Hx(\tau)d\tau - 2\int_{t-T}^{t}(Ru(\tau))^{\mathrm{T}}Kx(\tau)d\tau, \tag{2.45}$$

Adding $\int_{t-T}^{t} y^{\mathrm{T}}(\tau)Q_y y(\tau)d\tau$ on both sides of this equation and substituting $y(t) = Cx(t)$ and $H = A^{\mathrm{T}}P + PA$ on the right-hand side, we have,

$$\begin{aligned}& x^{\mathrm{T}}(t)Px(t) - x^{\mathrm{T}}(t-T)Px(t-T) + \int_{t-T}^{t} y^{\mathrm{T}}(\tau)Q_y y(\tau)d\tau \\ = \quad & \int_{t-T}^{t} x^{\mathrm{T}}(\tau)(A^{\mathrm{T}}P + PA + C^{\mathrm{T}}Q_y C)x(\tau)d\tau - 2\int_{t-T}^{t}(Ru(\tau))^{\mathrm{T}}Kx(\tau)d\tau.\end{aligned}$$

Next, we use our state parametrization (2.36) and substitute (2.41) and (2.42) in the above equation, which results in,

$$\begin{aligned}& z^{\mathrm{T}}(t)\bar{P}z(t) - z^{\mathrm{T}}(t-T)\bar{P}z(t-T) + \int_{t-T}^{t} y^{\mathrm{T}}(\tau)Q_y y(\tau)d\tau \\ = \quad & \int_{t-T}^{t} z^{\mathrm{T}}(\tau)M^{\mathrm{T}}(A^{\mathrm{T}}P + PA + C^{\mathrm{T}}Q_y C)Mz(\tau)d\tau - 2\int_{t-T}^{t}(Ru(\tau))^{\mathrm{T}}\bar{K}z(\tau)d\tau,\end{aligned}$$

or more compactly as,

$$\begin{aligned}& z^{\mathrm{T}}(t)\bar{P}z(t) - z^{\mathrm{T}}(t-T)\bar{P}z(t-T) + \int_{t-T}^{t} y^{\mathrm{T}}(\tau)Q_y y(\tau)d\tau \\ = \quad & \int_{t-T}^{t} z^{\mathrm{T}}(\tau)\bar{H}z(\tau)d\tau - 2\int_{t-T}^{t}(Ru(\tau))^{\mathrm{T}}\bar{K}z(\tau)d\tau,\end{aligned} \tag{2.46}$$

where $M \triangleq \begin{bmatrix} M_u & M_y \end{bmatrix}$ and $\bar{H} \triangleq M^{\mathrm{T}}(A^{\mathrm{T}}P + PA + C^{\mathrm{T}}Q_yC)M$. This new equation (2.46) serves as the key equation for our proposed output feedback based value iteration algorithm. Note that (2.46) is a scalar equation, which is linear in terms of the unknowns $\bar{H}$ and $\bar{K}$. These matrices are the output feedback counterparts of the matrices H and K. As there are more unknowns than the number of equations, we develop a system of l number of such equations by performing l finite window integrals each of length T. To solve this linear system of equations, we define the following data matrices,

$$\begin{aligned}
\delta_{zz} &= \Big[z(\tau)\otimes(\tau)z|_{t_0}^{t_1}, z(\tau)\otimes z(\tau)|_{t_1}^{t_2}, \cdots, z(\tau)\otimes z(\tau)|_{t_{l-1}}^{t_l}\Big]^{\mathrm{T}},\\
I_{zu} &= \Big[\int_{t_0}^{t_1} z(\tau)\otimes Ru(\tau)d\tau, \int_{t_1}^{t_2} z(\tau)\otimes Ru(\tau)d\tau, \cdots, \int_{t_{l-1}}^{t_l} z(\tau)\otimes Ru(\tau)d\tau\Big]^{\mathrm{T}},\\
I_{zz} &= \Big[\int_{t_0}^{t_1} \bar{z}^{\mathrm{T}}(\tau)d\tau, \int_{t_1}^{t_2} \bar{z}^{\mathrm{T}}(\tau)d\tau, \cdots, \int_{t_{l-1}}^{t_l} \bar{z}^{\mathrm{T}}(\tau)d\tau\Big]^{\mathrm{T}},\\
I_{yy} &= \Big[\int_{t_0}^{t_1} y(\tau)\otimes y(\tau)d\tau, \int_{t_1}^{t_2} y(\tau)\otimes y(\tau)d\tau, \cdots, \int_{t_{l-1}}^{t_l} y(\tau)\otimes y(\tau)d\tau\Big]^{\mathrm{T}}.
\end{aligned}$$

We rewrite (2.46) as the following matrix equation,

$$\begin{bmatrix} I_{zz}, 2I_{zu} \end{bmatrix} \begin{bmatrix} \mathrm{vecs}(\bar{H}) \\ \mathrm{vec}(\bar{K}) \end{bmatrix} = \delta_{zz}\mathrm{vec}(\bar{P}) + I_{yy}\mathrm{vec}(Q_y),$$

whose least-squares solution is given by,

$$\begin{bmatrix} \mathrm{vecs}(\bar{H}) \\ \mathrm{vec}(\bar{K}) \end{bmatrix} = \Big(\begin{bmatrix} I_{zz},\ 2I_{zu} \end{bmatrix}^{\mathrm{T}}\begin{bmatrix} I_{zz},\ 2I_{zu} \end{bmatrix}\Big)^{-1}\begin{bmatrix} I_{zz},\ 2I_{zu} \end{bmatrix}^{\mathrm{T}}\big(\delta_{xx}\mathrm{vec}(\bar{P}) + I_{yy}\mathrm{vec}(Q_y)\big) \tag{2.47}$$

We recall the following definitions before introducing the next algorithm. Let $\{B_q\}_{q=0}^{\infty}$ be some bounded nonempty sets that satisfy $B_q \subseteq B_{q+1}$, $q \in \mathbb{Z}_+$ and $\lim_{q\to\infty} B_q = \mathcal{P}_+$, where $\mathcal{P}_+$ is a set of positive semidefinite matrices. Also, let ϵ_k be the step size sequence satisfying $\epsilon_k > 0$, $\sum_{k=0}^{\infty}\epsilon_k = \infty$ and $\lim_{k\to\infty}\epsilon_k = 0$. With these definitions, Algorithm 2 presents the continuous-time model-free output feedback LQR algorithm.

Algorithm 2: Output Feedback IRL Value Iteration for LQR Control

Initialize. Select any initial control policy $u^0 = \bar{K}_0 z + \nu$ with ν being the exploration signal and $\bar{P}_0 \geq 0$, and set $k \leftarrow 0$, $q \leftarrow 0$.

Data Collection. Apply u^0 to the system to collect online data for $t \in [t_0, t_l]$, where $t_l = t_0 + lT$ and T is the interval length. Based on this data, perform the following iterations for $k = 0, 1, \cdots$,

loop:

Find the solution, $\bar{H}_k$ and $\bar{K}_k$, of the following learning equation

$$z^{\mathrm{T}}(t)\bar{P}_k z(t) - z^{\mathrm{T}}(t-T)\bar{P}_k z(t-T) + \int_{t-T}^{t} y^{\mathrm{T}}(\tau) Q_y y(\tau) d\tau$$
$$= \int_{t-T}^{t} z^{\mathrm{T}}(\tau)\bar{H}_k z(\tau) d\tau - 2\int_{t-T}^{t} (Ru(\tau))^{\mathrm{T}} \bar{K}_k z(\tau) d\tau.$$

Update $\tilde{\bar{P}}$ as,

$$\tilde{\bar{P}}_{k+1} \leftarrow \bar{P}_k + \epsilon_k(\bar{H}_k - \bar{K}_k^{\mathrm{T}} R \bar{K}_k).$$

if $\tilde{\bar{P}}_{k+1} \notin B_q$ **then**

$\bar{P}_{k+1} \leftarrow \bar{P}_0$ and $q \leftarrow q + 1$.

else if $\left\| \tilde{\bar{P}}_{k+1} - \bar{P}_k \right\| / \epsilon_k < \varepsilon$ **then**

return $\bar{P}_k$ as an estimate of $\bar{P}^*$

else

$\bar{P}_{k+1} \leftarrow \tilde{\bar{P}}_{k+1}$

end if

$k \leftarrow k + 1$

end loop.

Remark 2.3. *It can be seen in* (2.42) *that the control linearly depends on z, which means that the least-squares problem* (2.47) *will not have a unique solution as the data matrices will have linearly dependent columns. To overcome this difficulty, we need to apply an exploration input ν in the data collection phase*

and collect $l \geq (mn+pn)(mn+pn+1)/2 + m(mn+pn)$ *data samples to solve the least-squares problem* (2.47)*. That is, the following rank condition needs to be satisfied,*

$$\textit{rank}\left(\left[I_{zz},\ 2I_{zu}\right]\right) = \frac{(mn+pn)(mn+pn+1)}{2} + m(mn+pn). \tag{2.48}$$

Theorem 2.6. *Under the conditions of the controllability of* (A, B)*, the observability of* $\left(A, \sqrt{Q_y}C\right)$*, and the rank condition* (2.48)*, Algorithm 2 generates a control sequence which asymptotically converges to the optimal dynamic output feedback control law (2.43) as* $t \to \infty$*.*

Proof: The proof of convergence of the proposed output feedback algorithm follows along the same lines of the convergence proof of the state feedback model-free VI [14] algorithm. From Theorem 2.5, we have the convergence of the proposed state parametrization based on the filtered input and output signals. It follows that the proposed dynamic output feedback learning equation (2.46) converges to its state feedback counterpart (2.45), which in turn converges to the optimal solution if the rank condition (2.48) holds. This implies that the proposed output feedback algorithm also converges to the optimal solution as $t \to \infty$. This completes the proof. □

Remark 2.4. *In comparison with the previous output feedback LQR works based on RL [49], [77], the proposed output feedback value iteration (VI) algorithm does not require an initially stabilizing policy.*

Exploration Bias Immunity of Algorithm 2:

We now discuss the parameter estimation bias problem resulting from the use of exploration signals. For this purpose, the following result is presented.

Theorem 2.7. *The proposed output feedback algorithm Algorithm 2 is immune to the exploration bias problem.*

Proof: Consider the learning equation (2.46) with the excited input $\hat{u}$. Let $\hat{\bar{P}}$, $\hat{\bar{H}}$ and $\hat{\bar{K}}$ be the parameter estimates obtained as a result of these excited inputs. We have,

$$\begin{aligned} & z^{\mathrm{T}}(t)\hat{\bar{P}}z(t) - z^{\mathrm{T}}(t-T)\hat{\bar{P}}z(t-T) + \int_{t-T}^{t} y^{\mathrm{T}}(\tau)Q_y y(\tau)d\tau \\ = \quad & \int_{t-T}^{t} z^{\mathrm{T}}(\tau)\hat{\bar{H}}z(\tau)d\tau - 2\int_{t-T}^{t} (R\hat{u}(\tau))^{\mathrm{T}}\hat{\bar{K}}z(\tau)d\tau, \end{aligned}$$

Taking time derivative results in

$$\begin{aligned} & 2z^{\mathrm{T}}(t)\hat{\bar{P}}\dot{z}(t) - 2z^{\mathrm{T}}(t\text{–}T)\hat{\bar{P}}\dot{z}(t\text{–}T) + y^{\mathrm{T}}(t)Q_y y(t) - y^{\mathrm{T}}(t\text{–}T)Q_y y(t\text{–}T) \\ = \quad & z^{\mathrm{T}}(t)\hat{\bar{H}}z(t) - z^{\mathrm{T}}(t\text{–}T)\hat{\bar{H}}z(t\text{–}T) - 2(R\hat{u}(t))^{\mathrm{T}}\hat{\bar{K}}z(t) + 2(R\hat{u}(t\text{–}T))^{\mathrm{T}}\hat{\bar{K}}z(t\text{–}T). \end{aligned}$$

Further expansion yields,

$$\begin{aligned} & 2z^{\mathrm{T}}(t)\hat{\bar{P}}(\bar{\mathcal{A}}z(t) + \bar{\mathcal{B}}_1 u(t) + \bar{\mathcal{B}}_2 y(t)) + 2z^{\mathrm{T}}(t)\hat{M}^{\mathrm{T}}\hat{P}\hat{M}\bar{\mathcal{B}}_1\nu(t) - 2z^{\mathrm{T}}(t-T)\hat{\bar{P}}(\bar{\mathcal{A}}z(t-T) \\ & +\bar{\mathcal{B}}_1 u(t-T) + \bar{\mathcal{B}}_2 y(t-T)) - 2z^{\mathrm{T}}(t-T)\hat{M}^{\mathrm{T}}\hat{P}\hat{M}\bar{\mathcal{B}}_1\nu(t-T) + y^{\mathrm{T}}(t)Q_y y(t) - y^{\mathrm{T}}(t\text{–}T)Q_y y(t\text{–}T) \\ = \quad & z^{\mathrm{T}}(t)\hat{\bar{H}}z(t) - z^{\mathrm{T}}(t\text{–}T)\hat{\bar{H}}z(t\text{–}T) - 2(Ru(t))^{\mathrm{T}}\hat{\bar{K}}z(t) + 2(Ru(t\text{–}T))^{\mathrm{T}}\hat{\bar{K}}z(t\text{–}T) - 2(R\nu(t))^{\mathrm{T}}\hat{\bar{K}}z(t) \\ & +2(R\nu(t\text{–}T))^{\mathrm{T}}\hat{\bar{K}}z(t\text{–}T) \end{aligned}$$

where we have combined the dynamics of z based on the input-output dynamics of the observer in which $\bar{\mathcal{A}}$ and $\bar{\mathcal{B}}_i$ represent the combined system dynamics and input matrices corresponding to the $\zeta' s$. Using the fact that $M\bar{\mathcal{B}}_1 = B$, we have $2z^{\mathrm{T}}\hat{M}^{\mathrm{T}}\hat{P}\hat{M}\bar{\mathcal{B}}_1\nu = -2(R\nu)^{\mathrm{T}}\hat{\bar{K}}z$, thereby cancelling the delayed and non-delayed ν terms. Thus, we have,

$$\begin{aligned} & 2z^{\mathrm{T}}(t)\hat{\bar{P}}(\bar{\mathcal{A}}z(t) + \bar{\mathcal{B}}_1 u(t) + \bar{\mathcal{B}}_2 y(t)) - 2z^{\mathrm{T}}(t-T)\hat{\bar{P}}(\bar{\mathcal{A}}z(t-T) \\ & +\bar{\mathcal{B}}_1 u(t-T) + \bar{\mathcal{B}}_2 y(t-T)) + y^{\mathrm{T}}(t)Q_y y(t) - y^{\mathrm{T}}(t\text{–}T)Q_y y(t\text{–}T) \\ = \quad & z^{\mathrm{T}}(t)\hat{\bar{H}}z(t) - z^{\mathrm{T}}(t\text{–}T)\hat{\bar{H}}z(t\text{–}T) - 2(Ru(t))^{\mathrm{T}}\hat{\bar{K}}z(t) + 2Ru(t\text{–}T)^{\mathrm{T}}\hat{\bar{K}}z(t\text{–}T). \end{aligned}$$

Reversing the previous operations and performing integration result in

$$\begin{aligned} & z^{\mathrm{T}}(t)\hat{\bar{P}}z(t) - z^{\mathrm{T}}(t\text{–}T)\hat{\bar{P}}z(t\text{–}T) + \int_{t-T}^{t} y^{\mathrm{T}}(\tau)Q_y y(\tau)d\tau \\ = \quad & \int_{t-T}^{t} z^{\mathrm{T}}(\tau)\hat{\bar{H}}z(\tau)d\tau + 2\int_{t-T}^{t}(Ru(\tau))^{\mathrm{T}}\hat{\bar{K}}z(\tau)d\tau. \end{aligned}$$

Comparing the above equation with (2.46), we have $\hat{\bar{P}} = \bar{P}$, $\hat{\bar{H}} = \bar{H}$ and $\hat{\bar{K}} = \bar{K}$. This establishes the bias-free property of Algorithm 2. This completes the proof. □

2.4. Results

Example 1: A Discrete-time Unstable System:

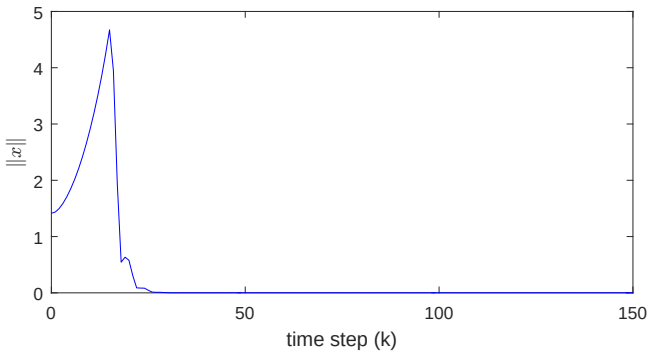

Fig. 2.1: Example 1: State trajectory of the closed-loop system under the state feedback Q-learning algorithm [40].

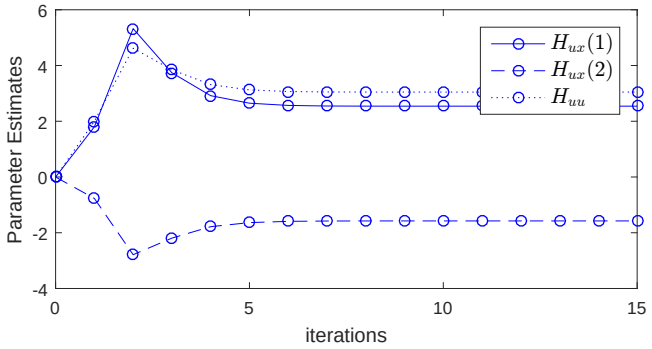

Fig. 2.2: Example 1: Convergence of the parameter estimates under state feedback Q-learning algorithm [40].

Consider system (2.1) with

$$A = \begin{bmatrix} 1.8 & -0.77 \\ 1 & 0 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, C = \begin{bmatrix} 1 & -0.5 \end{bmatrix}.$$

Let $Q_y = 1$ and $R = 1$ be the performance indices. The eigenvalues of the open-loop system are 0.7 and 1.1, and hence, the system is unstable. Let the characteristic polynomial of the observer be $\Lambda(z) = z^2$. We compare here the state feedback Q-learning algorithm and our proposed Q-learning based output feedback algorithm. By solving the Riccati equation (2.5), we obtain the optimal control matrices for the state feedback controller as

$$H_{ux}^* = \begin{bmatrix} 2.5416 & -1.5759 \end{bmatrix}, H_{uu}^* = 3.0467,$$

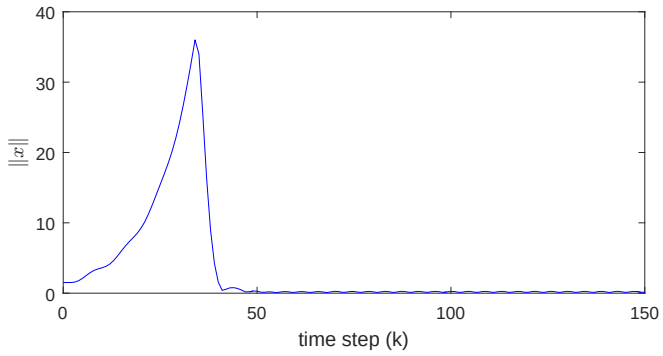

Fig. 2.3: Example 1: State trajectory of the closed-loop system under the proposed output feedback Q-learning Algorithm 1.

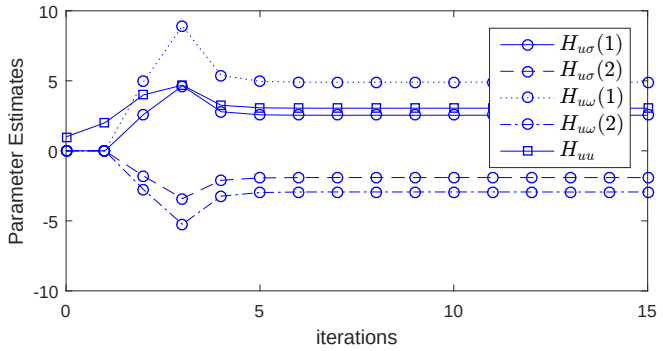

Fig. 2.4: Example 1: Convergence of the parameter estimates under the proposed output feedback Q-learning Algorithm 1.

and for our proposed Q-learning based output feedback algorithm as

$$H^*_{u\sigma} = \begin{bmatrix} 2.5416 & -1.9065 \end{bmatrix}, H^*_{u\omega} = \begin{bmatrix} 4.9055 & -2.9360 \end{bmatrix}, H^*_{uu} = 3.0467.$$

We use the VI algorithm as the system is unstable. The excitation condition is ensured by adding sinusoidal probing noises. The convergence criteria of $\varepsilon = 0.01$ was chosen for both the state feedback and output feedback algorithms. Seven data samples were collected for the state feedback algorithm, that is, $L = 7$, whereas $L = 18$ for the output feedback algorithm. The state trajectories under the state feedback Q-learning algorithm and our proposed output feedback Q-learning method are shown in Figs. 2.1 and 2.3, respectively. The convergence of the parameter estimates under these two methods are shown in Figs. 2.2 and 2.4, respectively. The final parameter estimates obtained for the state feedback algorithm are

$$\hat{H}_{ux} = \begin{bmatrix} 2.5416 & -1.5759 \end{bmatrix}, \hat{H}_{uu} = 3.0466,$$

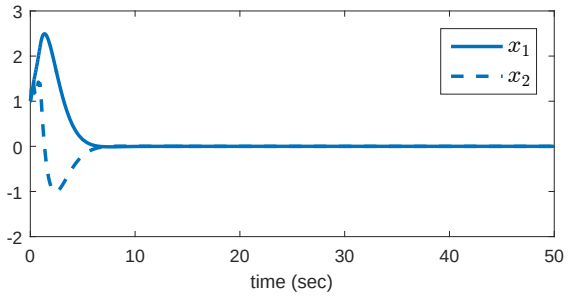

Fig. 2.5: Example 2: State trajectories of the closed-loop system under the state feedback [14].

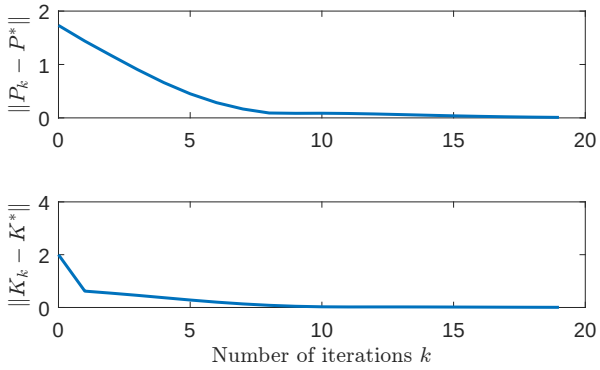

Fig. 2.6: Example 2: Convergence of the parameter estimates under the state feedback IRL algorithm [14].

and for our proposed output feedback Q-learning algorithm are

$$\hat{H}_{u\sigma} = \begin{bmatrix} 2.4641 & -1.9371 \end{bmatrix}, \hat{H}_{u\omega} = \begin{bmatrix} 4.8020 & -2.9832 \end{bmatrix}, \hat{H}_{uu} = 3.0396.$$

It can be seen that both the state feedback Q-learning algorithm and our proposed output feedback Q-learning algorithm converge to the solution of the ARE given by Equation (2.5).

Example 2: A Continuous-time Unstable System:

We now test the proposed output feedback IRL algorithm on a continuous-time unstable system of the form (2.28). For this purpose, we consider the double integrator system with

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

Both state feedback and output feedback IRL VI algorithms are evaluated. We choose the performance

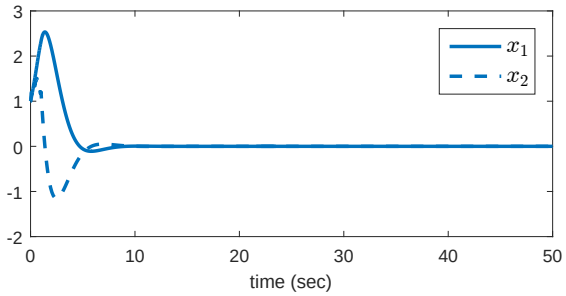

Fig. 2.7: Example 2: State trajectories of the closed-loop system under the output feedback IRL Algorithm 2.

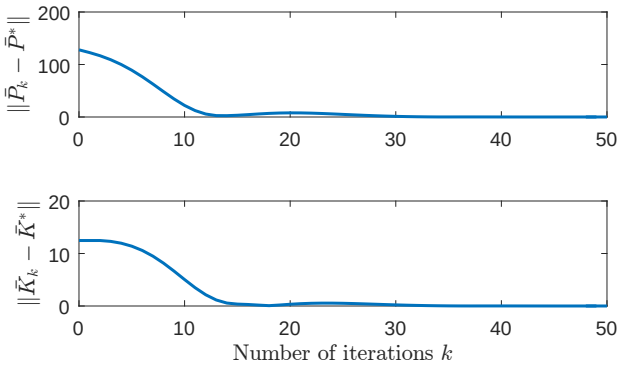

Fig. 2.8: Example 2: Convergence of the parameter estimates under the output feedback IRL Algorithm 2.

index parameters as $Q_y = 1$ and $R = 1$. The eigenvalues of the observer matrix $\mathcal{A}$ are placed at -2. The optimal control parameters as found by solving the ARE (2.34) are,

$$P^* = \begin{bmatrix} 1.4142 & 1.0000 \\ 1.0000 & 1.4142 \end{bmatrix}, K^* = \begin{bmatrix} 1.0000 & 1.4142 \end{bmatrix},$$

$$M_u = \begin{bmatrix} 1 & 0 \\ 4 & 1 \end{bmatrix}, M_y = \begin{bmatrix} 4 & 4 \\ 0 & 4 \end{bmatrix}.$$

The initial controller parameters are set to zero. We choose 20 learning intervals of period $T = 0.05$s. Step size $\epsilon_k = (k^{0.2} + 5)^{-1}, k = 0, 1, 2, \ldots$ and the set $B_q = \left\{\bar{P} \in \mathcal{P}_+^4 : |\bar{P}| \leq 800(q+1)\right\}, q = 0, 1, 2, \ldots$. The exploration condition is met by injecting sinusoidal signals of different frequencies in the control. We compare here both the state feedback VI algorithm proposed in [14] and the proposed output feedback VI algorithm (Algorithm 2). The state feedback results are shown in Figs. 2.5 and 2.6, and the proposed output feedback results are shown in Figs. 2.7 and 2.8. It can be seen that similar to

the state feedback results, the output feedback parameters also converge to their nominal values with performance quite close to that of the state feedback case along with the advantage that the output feedback algorithm does not require the measurement of the system state. However, it should be noted that the proposed output feedback learning equation contains more unknown terms as compared to the state feedback learning equation. As a result, the output feedback algorithm can take longer to converge. Furthermore, the exploration noise can be safely removed after the initial data collection phase.

2.5. Summary

In this chapter, we have presented model-free output feedback RL algorithms to solve the LQR optimal stabilization problem. For the discrete-time case, we have developed an output feedback Q-function description for LQR using a parametrization of the state given by the past input and output data. A Q-learning scheme based on value iteration has been developed, which learns the optimal output feedback Q-function and optimal output feedback control parameters. For the continuous-time problem, we have derived an output feedback integral reinforcement learning equation based on the parametrization of the state given in terms of the filtered input and output data. Then, an integral reinforcement learning scheme based on value iteration has been presented which learns the optimal value function matrix and the optimal output feedback control parameters. It has been shown that the exploration noise does not result in bias in the parameter estimates. As a result, the need of employing a discounting factor in the cost function is eliminated and closed-loop stability is preserved. Hence, optimal stabilization of the system is achieved without requiring the knowledge of the system dynamics or the internal state of the system. Simulation results have been presented that confirm the effectiveness of the proposed schemes.

3. Model-Free Zero-Sum Game and Disturbance Rejection Control

3.1. Introduction

H-infinity control offers both robust performance and stabilization guarantees [18], which make it a good candidate in problems involving external and cross-coupling disturbances. It finds many important applications such as rotorcrafts [53], VSTOL aircrafts [26], guided projectiles [57], satellites [20] and power systems [5]. It has been shown that the H-infinity problem is strongly related with the zero-sum game problem in game theory [7]. The solution of the linear H-infinity control problem is obtained by solving a game algebraic Riccati equation (GARE). Computational methods to solve this equation have been developed but they require perfect knowledge of the system dynamics. Model-free reinforcement learning has been used to solve the zero-sum game and the associated H-infinity control problem in [4], [30], [41], [63]. However, these methods solve the full information H-infinity problem, which require access to both the internal state and disturbance channel. In this work, we will solve the partial information H-infinity control problem by using output feedback.

3.2. Q-learning Based Output Feedback Zero-Sum Game

Consider a discrete-time linear time-invariant system in the state-space form,

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + Ew_k, \\ y_k &= Cx_k, \end{aligned} \tag{3.1}$$

where $x_k \in \mathbb{R}^n$ is the system state vector, $u_k \in \mathbb{R}^{m_1}$ is the control input vector, $w_k \in \mathbb{R}^{m_2}$ is the disturbance input vector, and $y_k \in \mathbb{R}^p$ is the output vector. The zero-sum problem game can be formulated as a minimax problem with the optimal value function of the form [7],

$$V^*(x_k) = \min_{u_i} \max_{w_i} \sum_{i=k}^{\infty} r(x_i, u_i, w_i), \tag{3.2}$$

where the control input u acts as the minimizing player and the disturbance input w acts as the maximizing player. For our H-infinity problem, the utility function r_k takes the quadratic form,

$$r_k = y_k^{\mathrm{T}} Q_y y_k + u_k^{\mathrm{T}} R u_k - \gamma^2 w_k^{\mathrm{T}} w_k, \tag{3.3}$$

where $Q_y \geq 0$ and $R > 0$ are the user-defined weighting matrices and γ is an upper bound on the desired L_2 gain from the disturbance to the performance output [7]. The aim of the H-infinity control is to find

the optimal control policy u_k^* such that the closed-loop system is asymptotically stable with $w_k = 0$ and it satisfies the H_∞ constraint with the following disturbance attenuation condition [30],

$$\sum_{i=0}^{\infty} (y_i^{\mathrm{T}} Q_y y_i + u_i^{\mathrm{T}} R u_i) \leq \gamma^2 \sum_{i=0}^{\infty} w_i^{\mathrm{T}} w_i, \tag{3.4}$$

for some user-defined L_2 disturbance attenuation level $\gamma \geq \gamma^* > 0$, for all $w_k \in L_2[0, \infty)$, under the usual controllability and observability assumptions on (A, B) and (A, C), respectively.

When the system dynamics is completely known and full-state feedback x_k is available, there exist a unique optimal stabilizing controller $u_k^* = L^* x_k$ and a unique worst-case disturbance $w_k^* = K^* x_k$, where

$$L^* = \Big(R + B^{\mathrm{T}} P B - B^{\mathrm{T}} P E (E^{\mathrm{T}} P E - \gamma^2 I)^{-1} E^{\mathrm{T}} P B\Big)^{-1} \Big(B^{\mathrm{T}} P E (E^{\mathrm{T}} P E - \gamma^2 I)^{-1} E^{\mathrm{T}} P A - B^{\mathrm{T}} P A\Big), \tag{3.5}$$

$$K^* = \Big(E^{\mathrm{T}} P E - \gamma^2 I - E^{\mathrm{T}} P B (R + B^{\mathrm{T}} P B)^{-1} B^{\mathrm{T}} P E\Big)^{-1} \Big(E^{\mathrm{T}} B (R + B^{\mathrm{T}} P B)^{-1} B^{\mathrm{T}} P A - E^{\mathrm{T}} P A\Big), \tag{3.6}$$

under the conditions of controllability of (A, B) and observability of $\big(A, \sqrt{Q}\big)$, where $\sqrt{Q}^{\mathrm{T}} \sqrt{Q} = Q$, $Q = C^{\mathrm{T}} Q_y C$ [37], [38]. Here, $P = P^{\mathrm{T}} > 0$ is the unique positive definite solution to the game algebraic Riccati equation (GARE),

$$P = A^{\mathrm{T}} P A + Q - \begin{bmatrix} A^{\mathrm{T}} P B & A^{\mathrm{T}} P E \end{bmatrix} \begin{bmatrix} R + B^{\mathrm{T}} P B & B^{\mathrm{T}} P E \\ E^{\mathrm{T}} P B & E^{\mathrm{T}} P E - \gamma^2 I \end{bmatrix}^{-1} \begin{bmatrix} B^{\mathrm{T}} P A \\ E^{\mathrm{T}} P A \end{bmatrix}. \tag{3.7}$$

Moreover, the following inequality must be satisfied [7],

$$I - \gamma^{-2} E^{\mathrm{T}} P E > 0. \tag{3.8}$$

We seek to solve the given problem by using only the input-output data. No knowledge of system dynamics (A, B, C, E) and no measurement of state x_k are available. For the H-infinity problem, we have a quadratic cost and so the associated Q-function can be expressed as

$$Q(x_k, u_k, w_k) = x_k^{\mathrm{T}} Q x_k + u_k^{\mathrm{T}} R u_k - \gamma^2 w_k^{\mathrm{T}} w_k + (A x_k + B u_k + E w_k)^{\mathrm{T}} P (A x_k + B u_k + E w_k) \tag{3.9}$$

Now, we would like to express the Q-function in (3.9) in terms of inputs and outputs samples instead of state x_k. We refer to the resulting expression as output feedback Q-function. Based on Theorem 2.1, we

have

$$x_k = M_y \bar{y}_{k-1,k-N} + M_u \bar{u}_{k-1,k-N} + M_w \bar{w}_{k-1,k-N}, \tag{3.10}$$

It can be easily verified that substitution of (3.10) in (3.9) results in,

$$Q = \begin{bmatrix} \bar{u}_{k-1,k-N} \\ \bar{w}_{k-1,k-N} \\ \bar{y}_{k-1,k-N} \\ u_k \\ w_k \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} H_{\bar{u}\bar{u}} & H_{\bar{u}\bar{w}} & H_{\bar{u}\bar{y}} & H_{\bar{u}u} & H_{\bar{u}w} \\ H_{\bar{w}\bar{u}} & H_{\bar{w}\bar{w}} & H_{\bar{w}\bar{y}} & H_{\bar{w}u} & H_{\bar{w}w} \\ H_{\bar{y}\bar{u}} & H_{\bar{y}\bar{w}} & H_{\bar{y}\bar{y}} & H_{\bar{y}u} & H_{\bar{y}w} \\ H_{u\bar{u}} & H_{u\bar{w}} & H_{u\bar{y}} & H_{uu} & H_{uw} \\ H_{w\bar{u}} & H_{w\bar{w}} & H_{w\bar{y}} & H_{wu} & H_{ww} \end{bmatrix} \begin{bmatrix} \bar{u}_{k-1,k-N} \\ \bar{w}_{k-1,k-N} \\ \bar{y}_{k-1,k-N} \\ u_k \\ w_k \end{bmatrix} \triangleq z_k^{\mathrm{T}} H z_k,$$

where $z_k = \begin{bmatrix} \bar{u}_{k-1,k-N}^{\mathrm{T}} & \bar{w}_{k-1,k-N}^{\mathrm{T}} & \bar{y}_{k-1,k-N}^{\mathrm{T}} & u_k^{\mathrm{T}} & w_k^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$, $H = H^{\mathrm{T}} \in \mathbb{R}^{l \times l}$, and $l = m_1 N + m_2 N + pN + m_1 + m_2$. The partitioned matrices are defined as

$$\begin{aligned}
H_{\bar{u}\bar{u}} &= M_u^{\mathrm{T}}(Q + A^{\mathrm{T}} P A) M_u \in \mathbb{R}^{m_1 N \times m_1 N}, \\
H_{\bar{u}\bar{w}} &= M_u^{\mathrm{T}}(Q + A^{\mathrm{T}} P A) M_w \in \mathbb{R}^{m_1 N \times m_2 N}, \\
H_{\bar{u}\bar{y}} &= M_u^{\mathrm{T}}(Q + A^{\mathrm{T}} P A) M_y \in \mathbb{R}^{m_1 N \times pN}, \\
H_{\bar{u}u} &= M_u^{\mathrm{T}} A^{\mathrm{T}} P B \in \mathbb{R}^{m_1 N \times m_1}, \\
H_{\bar{u}w} &= M_u^{\mathrm{T}} A^{\mathrm{T}} P E \in \mathbb{R}^{m_1 N \times m_2}, \\
H_{\bar{w}\bar{u}} &= M_w^{\mathrm{T}}(Q + A^{\mathrm{T}} P A) M_u \in \mathbb{R}^{m_2 N \times m_1 N}, \\
H_{\bar{w}\bar{w}} &= M_w^{\mathrm{T}}(Q + A^{\mathrm{T}} P A) M_w \in \mathbb{R}^{m_2 N \times m_2 N}, \\
H_{\bar{w}\bar{y}} &= M_w^{\mathrm{T}}(Q + A^{\mathrm{T}} P A) M_y \in \mathbb{R}^{m_2 N \times pN}, \\
H_{\bar{w}u} &= M_w^{\mathrm{T}} A^{\mathrm{T}} P B \in \mathbb{R}^{m_2 N \times m_1}, \\
H_{\bar{w}w} &= M_w^{\mathrm{T}} A^{\mathrm{T}} P E \in \mathbb{R}^{m_2 N \times m_2}, \\
H_{\bar{y}\bar{u}} &= M_y^{\mathrm{T}}(Q + A^{\mathrm{T}} P A) M_u \in \mathbb{R}^{pN \times m_1 N}, \\
H_{\bar{y}\bar{w}} &= M_y^{\mathrm{T}}(Q + A^{\mathrm{T}} P A) M_w \in \mathbb{R}^{pN \times m_2 N}, \\
H_{\bar{y}\bar{y}} &= M_y^{\mathrm{T}}(Q + A^{\mathrm{T}} P A) M_y \in \mathbb{R}^{pN \times pN}, \\
H_{\bar{y}u} &= M_y^{\mathrm{T}} A^{\mathrm{T}} P B \in \mathbb{R}^{pN \times m_1}, \\
H_{\bar{y}w} &= M_y^{\mathrm{T}} A^{\mathrm{T}} P E \in \mathbb{R}^{pN \times m_2},
\end{aligned} \tag{3.11}$$

$$H_{u\bar{u}} = B^{\mathrm{T}}PAM_u \in \mathbb{R}^{m_1\times m_1 N},$$

$$H_{u\bar{w}} = B^{\mathrm{T}}PAM_w \in \mathbb{R}^{m_1\times m_2 N},$$

$$H_{u\bar{y}} = B^{\mathrm{T}}PAM_y \in \mathbb{R}^{m_1\times pN},$$

$$H_{w\bar{u}} = E^{\mathrm{T}}PAM_u \in \mathbb{R}^{m_2\times m_1 N},$$

$$H_{w\bar{w}} = E^{\mathrm{T}}PAM_w \in \mathbb{R}^{m_2\times m_2 N},$$

$$H_{w\bar{y}} = E^{\mathrm{T}}PAM_y \in \mathbb{R}^{m_2\times pN},$$

$$H_{uu} = B^{\mathrm{T}}PB + R \in \mathbb{R}^{m_1\times m_1},$$

$$H_{uw} = B^{\mathrm{T}}PE \in \mathbb{R}^{m_1\times m_2},$$

$$H_{wu} = E^{\mathrm{T}}PB \in \mathbb{R}^{m_2\times m_1},$$

$$H_{ww} = E^{\mathrm{T}}PE - \gamma^2 I \in \mathbb{R}^{m_2\times m_2}.$$

The output feedback optimal control u_k^* and the worst-case disturbance w_k^* are obtained by solving simultaneously $\frac{\partial}{\partial u_k}Q^* = 0$ and $\frac{\partial}{\partial w_k}Q^* = 0$ for u_k and w_k. This results in,

$$\begin{aligned} u_k^* = (H_{uu}^* - H_{uw}^* {H_{ww}^*}^{-1}(H_{wu}^*))^{-1}\Big(H_{uw}^* {H_{ww}^*}^{-1}(H_{w\bar{u}}\bar{u}_{k-1,k-N} + H_{w\bar{w}}^*\bar{w}_{k-1,k-N} + H_{w\bar{y}}^*\bar{y}_{k-1,k-N}) \\ -(H_{u\bar{u}}^*\bar{u}_{k-1,k-N} + H_{u\bar{w}}^*\bar{w}_{k-1,k-N} + H_{u\bar{y}}^*\bar{y}_{k-1,k-N})\Big), \end{aligned} \tag{3.12}$$

$$\begin{aligned} w_k^* = (H_{ww}^* - H_{wu}^* {H_{uu}^*}^{-1}H_{uw}^*)^{-1}\Big(H_{wu}^* {H_{uu}^*}^{-1}(H_{u\bar{u}}^*\bar{u}_{k-1,k-N} + H_{u\bar{w}}^*\bar{w}_{k-1,k-N} + H_{u\bar{y}}^*\bar{y}_{k-1,k-N}) \\ -(H_{w\bar{u}}^*\bar{u}_{k-1,k-N} + H_{w\bar{w}}^*\bar{w}_{k-1,k-N} + H_{w\bar{y}}^*\bar{y}_{k-1,k-N})\Big). \end{aligned} \tag{3.13}$$

Lemma 3.1. *The output feedback policies given by (3.12) and (3.13) converge to the state feedback policies (3.5) and (3.6), respectively, that solve the zero-sum game problem (3.1)–(3.3).*

Proof: We know that the state vector can be represented by the input-output data sequence as in (3.10). It can be easily verified that substituting (3.10) and (3.11) in (3.12) and (3.13) results in the state feedback policies (3.5) and (3.6), which solve the zero-sum game problem. So, the output feedback policies (3.12) and (3.13) are the equivalent policies that also solve the zero-sum game problem (3.1)–(3.3). This completes the proof. □

Using the state parametrization (3.10), we have the following Q-learning equation,

$$\bar{H}^{\mathrm{T}}\bar{z}_k = y_k^{\mathrm{T}}Q_y y_k + u_k^{\mathrm{T}}Ru_k - \gamma^2 w_k^{\mathrm{T}}w_k + \bar{H}^{\mathrm{T}}\bar{z}_{k+1}. \tag{3.14}$$

where $\bar{H}$ is the vector containing the column wise components of H and $\bar{z}$ is the quadratic basis, as defined in Section 2.2. We can now utilize the Q-learning technique to learn our output feedback Q-function matrix.

Algorithm 1: Output Feedback Q-learning Value Iteration Algorithm for Zero-Sum Game

Initialization. Start with $\bar{H}^0 = 0$, $u_k^0 = n_k$ and $w_k^0 = v_k$ with n_k and v_k being the exploration signals. Then, for $j = 1, 2, \cdots$, perform until convergence,

$$\left\|\bar{H}^j - \bar{H}^{j-1}\right\| < \varepsilon :$$

Value Update. Determine the least-squares solution of

$$\left(\bar{H}^j\right)^{\mathrm{T}} \bar{z}_k = y_k^{\mathrm{T}} Q_y y_k + u_k^{\mathrm{T}} R u_k - \gamma^2 w_k^{\mathrm{T}} w_k + \left(\bar{H}^{j-1}\right)^{\mathrm{T}} \bar{z}_{k+1}.$$

Policy Improvement. Determine an improved policy for next iteration,

$$\begin{aligned}
u_k^{j+1} &= (H_{uu}^j - H_{uw}^j (H_{ww}^j)^{-1} H_{wu}^j)^{-1} \Big(H_{uw}^j (H_{ww}^j)^{-1} (H_{w\bar{u}}^j \bar{u}_{k-1,k-N} + H_{w\bar{w}}^j \bar{w}_{k-1,k-N} + H_{w\bar{y}}^j \bar{y}_{k-1,k-N}) \\
&\quad - (H_{u\bar{u}}^j \bar{u}_{k-1,k-N} + H_{u\bar{w}}^j \bar{w}_{k-1,k-N} + H_{u\bar{y}}^j \bar{y}_{k-1,k-N}) \Big), \\
w_k^{j+1} &= (H_{ww}^j - H_{wu}^j (H_{uu}^j)^{-1} H_{uw}^j)^{-1} \Big(H_{wu}^j (H_{uu}^j)^{-1} (H_{u\bar{u}}^j \bar{u}_{k-1,k-N} + H_{u\bar{w}}^j \bar{w}_{k-1,k-N} + H_{u\bar{y}}^j \bar{y}_{k-1,k-N}) \\
&\quad - (H_{w\bar{u}}^j \bar{u}_{k-1,k-N} + H_{w\bar{w}}^j \bar{w}_{k-1,k-N} + H_{w\bar{y}}^j \bar{y}_{k-1,k-N}) \Big).
\end{aligned}$$

The above Q-learning iterative algorithm consists of two steps. In the policy evaluation step, we use the key equation (3.14) recursively to solve for the unknown vector $\bar{H}$ in the least-squares sense by collecting $L \geq l(l+1)/2$ data samples of u_k, w_k, y_k, $\bar{u}_{k-1,k-N}$, $\bar{w}_{k-1,k-N}$, $\bar{y}_{k-1,k-N}$, $\bar{u}_{k,k-N+1}$, $\bar{w}_{k,k-N+1}$ and $\bar{y}_{k,k-N+1}$ to form the data matrices $\Phi \in \mathbb{R}^{l(l+1)/2 \times L}$ and $\Upsilon^{j-1} \in \mathbb{R}^{L \times 1}$ defined by

$$\begin{aligned}
\Phi &= [\bar{z}_k^1, \bar{z}_k^2, \cdots, \bar{z}_k^L], \\
\Upsilon^{j-1} &= [r_k^1 + (\bar{H}^{j-1})^{\mathrm{T}} \bar{z}_{k+1}^1, r_k^2 + (\bar{H}^{j-1})^{\mathrm{T}} \bar{z}_{k+1}^2, \cdots, r_k^L + (\bar{H}^{j-1})^{\mathrm{T}} \bar{z}_{k+1}^L]^{\mathrm{T}}.
\end{aligned}$$

Then, the least-squares solution of (3.14) is given by

$$\bar{H}^j = (\Phi \Phi^{\mathrm{T}})^{-1} \Phi \Upsilon^{j-1}, \tag{3.15}$$

Notice in (3.12) and (3.13) that u_k and w_k are linearly dependent on $\bar{u}_{k-1,k-N}$, $\bar{w}_{k-1,k-N}$ and $\bar{y}_{k-1,k-N}$, which means that $\Phi\Phi^{\mathrm{T}}$ will not be invertible. To overcome this issue, one adds excitation noise in u_k and w_k to guarantee a unique solution to (3.15). In other words, the following condition needs to be satisfied,

$$\text{rank}(\Phi) = l(l+1)/2. \tag{3.16}$$

This excitation condition can be met in several ways such as adding sinusoidal noise of various frequencies, exponentially decaying noise and Gaussian noise.

Theorem 3.1. *Assume that the linear quadratic zero-sum game is solvable. Then, the output feedback Q-learning algorithm generates a sequence of policies* $\{u_k^j, j = 1, 2, 3, ...\}$ *and* $\{w_k^j, j = 1, 2, 3, ...\}$ *that converge to the optimal output feedback policies given in (3.12) and (3.13) as* $j \to \infty$ *if the rank condition* (3.16) *holds.*

Proof: The proof follows from [4], where it is shown that under sufficient excitation, the state feedback Q-learning iterative algorithm generates a sequence of policies $\{u_k^j, j = 1, 2, 3, ...\}$ and $\{w_k^j, j = 1, 2, 3, ...\}$ that converge to the optimal state feedback policies (3.25) and (3.26). By the state parametrization (3.10), we see that the state feedback and output feedback Q-functions are equivalent, and by Lemma 3.1, the output feedback policies (3.12) and (3.13) are equivalent to (3.5) and (3.6), respectively. Therefore, following the result in [4], we can conclude that, under sufficient excitation such that the rank condition (3.16) holds, the output feedback Q-learning algorithm generates a sequence of policies that converge to the optimal output feedback policies as $j \to \infty$. This completes the proof. □

Exploration Bias Immunity of Algorithm 1:

We next show that the proposed Q-learning scheme does not lead to bias in the Q-function estimates.

Theorem 3.2. *The excitation noise required to satisfy the excitation condition does not result in any bias in the Q-function estimates.*

Proof: Based on the state parametrization (3.10), we know that the output feedback Q-function in (3.11) is equivalent to the original state feedback Q-function (3.9). Under the excitation noise, we can write the Q-function as

$$Q(x_k, \hat{u}_k, \hat{w}_k) = r_k(x_k, \hat{u}_k, \hat{w}_k) + V(x_{k+1}),$$

where $\hat{u}_k = u_k + n_k$ and $\hat{w}_k = w_k + v_k$ with n_k and v_k being the excitation noise signals.

Let $\hat{H}$ be the estimate of H obtained using $\hat{u}_k$ and $\hat{w}$. It then follows from (3.9) that

$$\begin{bmatrix} x_k \\ \hat{u}_k \\ \hat{w}_k \end{bmatrix}^{\mathrm{T}} \hat{H} \begin{bmatrix} x_k \\ \hat{u}_k \\ \hat{w}_k \end{bmatrix} = r_k(x_k, \hat{u}_k, \hat{w}_k) + (Ax_k + B\hat{u}_k + E\hat{w}_k)^{\mathrm{T}} P (Ax_k + B\hat{u}_k + E\hat{w}_k).$$

Upon further expansion, we have,

$$\begin{aligned}
& \begin{bmatrix} x_k \\ u_k \\ w_k \end{bmatrix}^{\mathrm{T}} \hat{H} \begin{bmatrix} x_k \\ u_k \\ w_k \end{bmatrix} + x_k^{\mathrm{T}} A^{\mathrm{T}} P B n_k + n_k^{\mathrm{T}} B^{\mathrm{T}} P A x_k \\
& + n_k^{\mathrm{T}}(R + B^{\mathrm{T}} P B) u_k + n_k^{\mathrm{T}} B^{\mathrm{T}} P E w_k + v_k^{\mathrm{T}} E^{\mathrm{T}} P A x_k \\
& + u_k^{\mathrm{T}}(R + B^{\mathrm{T}} P B) n_k + n_k^{\mathrm{T}}(R + B^{\mathrm{T}} P B) n_k \\
& + x_k^{\mathrm{T}} A^{\mathrm{T}} P E v_k + w_k^{\mathrm{T}}(E^{\mathrm{T}} P E - \gamma^2 I) v_k + n_k^{\mathrm{T}} B^{\mathrm{T}} P E v_k \\
& + v_k^{\mathrm{T}}(E^{\mathrm{T}} P E - \gamma^2 I) v_k + u_k^{\mathrm{T}} B^{\mathrm{T}} P E v_k + v_k^{\mathrm{T}} E^{\mathrm{T}} P B u_k \\
& + w_k^{\mathrm{T}} E^{\mathrm{T}} P B n_k + v_k^{\mathrm{T}} E^{\mathrm{T}} P B n_k + v_k^{\mathrm{T}}(E^{\mathrm{T}} P E - \gamma^2 I) w_k \\
= \; & x_k^{\mathrm{T}} Q x_k + u_k^{\mathrm{T}} R u_k - \gamma^2 w_k^{\mathrm{T}} w_k + n_k^{\mathrm{T}} R n_k + n_k^{\mathrm{T}} R u_k \\
& + u_k^{\mathrm{T}} R n_k - \gamma^2 w_k^{\mathrm{T}} v_k - \gamma^2 v_k^{\mathrm{T}} w_k - \gamma^2 v_k^{\mathrm{T}} v_k \\
& + (Ax_k + Bu_k + Ew_k)^{\mathrm{T}} P (Ax_k + Bu_k + Ew_k) \\
& + (Ax_k + Bu_k + Ew_k)^{\mathrm{T}} P B n_k + n_k^{\mathrm{T}} B^{\mathrm{T}} P B n_k \\
& + (Bn_k)^{\mathrm{T}} P (Ax_k + Bu_k + Ew_k) + n_k^{\mathrm{T}} B^{\mathrm{T}} P E v_k \\
& + v_k^{\mathrm{T}} E^{\mathrm{T}} P B n_k + (Ax_k + Bu_k + Ew_k)^{\mathrm{T}} P E v_k \\
& + (Ev_k)^{\mathrm{T}} P (Ax_k + Bu_k + Ew_k) + v_k^{\mathrm{T}} E^{\mathrm{T}} P E v_k.
\end{aligned}$$

It can be easily verified that all the terms involving n_k and v_k get canceled on both sides of the equation,

and we are left with

$$\begin{bmatrix} x_k \\ u_k \\ w_k \end{bmatrix}^{\mathrm{T}} \hat{H} \begin{bmatrix} x_k \\ u_k \\ w_k \end{bmatrix} = r_k(x_k, u_k, w_k) + (Ax_k + Bu_k + Ew_k)^{\mathrm{T}} P (Ax_k + Bu_k + Ew_k).$$

Comparing the above equation with (3.9), we have $\hat{H} = H$, that is

$$Q(x_k, u_k, w_k) = r_k(x_k, u_k, w_k) + V(x_{k+1}).$$

In view of (3.11), we have

$$z_k^{\mathrm{T}} H z_k = y_k^{\mathrm{T}} Q_y y_k + u_k^{\mathrm{T}} R u_k - \gamma^2 {w_k}^{\mathrm{T}} w_k + z_{k+1}^{\mathrm{T}} H z_{k+1}.$$

That is, we have obtained the Bellman equation in the absence of excitation noise as given in (3.14). This completes the proof. □

3.3. Integral Reinforcement Learning Based Output Feedback Zero-Sum Game

Consider a continuous-time linear time-invariant system given by the following state space representation,

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + Ew(t), \\ y(t) &= Cx(t), \end{aligned} \tag{3.17}$$

where $x \in \mathbb{R}^n$ refers to the state vector, $u \in \mathbb{R}^{m_1}$ corresponds to the input vector of Player 1, $w \in \mathbb{R}^{m_2}$ corresponds to input vector of the opposing Player 2, and $y \in \mathbb{R}^p$ is the output vector. Let us define the infinite horizon cost function as,

$$J(x(0), u, w) = \int_0^\infty r(x(\tau), u(\tau), w(\tau)) d\tau. \tag{3.18}$$

We consider a linear quadratic differential game in which the utility function $r(x, u, w)$ takes the following quadratic form,

$$r = y^{\mathrm{T}}(t) Q_y y(t) + u^{\mathrm{T}}(t) u(t) - \gamma^2 w^{\mathrm{T}}(t) w(t), \tag{3.19}$$

where $Q_y \geq 0$ is the user-defined weighting matrix and $\gamma \geq \gamma^* > 0$ is an upper bound on the desired L_2 gain from the player w to the performance output [7], that is,

$$\int_0^\infty (y^{\mathrm{T}}(\tau)Q_y y(\tau) + u^{\mathrm{T}}(\tau)u(\tau))d\tau \leq \gamma^2 \int_0^\infty w^{\mathrm{T}}(\tau)w(\tau)d\tau, \tag{3.20}$$

$\forall w \in L_2[0,\infty)$. Since we are interested in feedback policies of the form $u(x) = Kx$ and $w(x) = Hx$, we introduce a value function that gives the cost of executing these policies, as given by,

$$V(x) = \int_t^\infty y^{\mathrm{T}}(\tau)Q_y y(\tau) + u^{\mathrm{T}}(\tau)u(\tau) - \gamma^2 w^{\mathrm{T}}(\tau)w(\tau)d\tau. \tag{3.21}$$

Furthermore, if the feedback polices are such that the resulting closed-system is asymptotically stable, then $V(x)$ is quadratic in the state [7], as follows,

$$V(x) = x^{\mathrm{T}}Px, \tag{3.22}$$

for $P > 0$. The optimal value function associated with the zero-sum game is of the form,

$$V^*(x) = \min_u \max_w \int_t^\infty y^{\mathrm{T}}(\tau)Q_y y(\tau) + u^{\mathrm{T}}(\tau)u(\tau) - \gamma^2 w^{\mathrm{T}}(\tau)w(\tau)d\tau, \tag{3.23}$$

where the input u acts as the minimizing player and the input w acts as the maximizing player. The aim of the zero-sum game problem is to find the saddle point solution (u^*, w^*) that satisfies the following pair of Nash equilibrium inequalities,

$$J(x(0), u^*, w) \leq J(x(0), u^*, w^*) \leq J(x(0), u, w^*), \tag{3.24}$$

for any feedback policies $u(x)$ and $w(x)$. For the special case of the H-infinity control problem, the maximizing player w can be regarded as an $L_2[0,\infty)$ disturbance, and achieving the Nash equilibrium amounts to the stabilization of the system at the equilibrium point $x = 0$.

There exist the unique state feedback policies $u^* = K^*x$ and $w^* = H^*x$ that achieve the objective (3.2) under the conditions of controllability of (A, B) and observability of $\left(A, \sqrt{Q}\right)$, with $\sqrt{Q}^{\mathrm{T}}\sqrt{Q} = Q$ and $Q = C^{\mathrm{T}}Q_y C$ [37], [38], where

$$K^* = -B^{\mathrm{T}}P^*, \tag{3.25}$$

$$H^* = \gamma^{-2}E^{\mathrm{T}}P^*. \tag{3.26}$$

Here, $P^* = (P^*)^{\mathrm{T}} > 0$ is the unique positive definite solution to the game algebraic Riccati equation (GARE),

$$A^{\mathrm{T}}P + PA + Q - P(BB^{\mathrm{T}} - \gamma^{-2}EE^{\mathrm{T}})P = 0. \tag{3.27}$$

The above solution is, however, model-based and requires full state feedback. In this section we will present a design using dynamic output feedback towards solving the model-free linear differential zero-sum game and the asssociated H-infinity control problem. To this end, we first parameterize the state based on Theorem 2.5 as follows,

$$\bar{x} = M_u\zeta_u + M_w\zeta_w + M_y\zeta_y \tag{3.28}$$

Next, we use the state parametrization to describe the cost function in (3.22). It can be easily verified that substitution of (3.28) in (3.22) results in,

$$\begin{aligned} V &= \begin{bmatrix} \zeta_u \\ \zeta_w \\ \zeta_y \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} M_u & M_w & M_y \end{bmatrix}^{\mathrm{T}} P \begin{bmatrix} M_u & M_w & M_y \end{bmatrix} \begin{bmatrix} \zeta_u \\ \zeta_w \\ \zeta_y \end{bmatrix} \\ &\triangleq z^{\mathrm{T}}\bar{P}z, \end{aligned} \tag{3.29}$$

where,

$$\begin{aligned} z &= \begin{bmatrix} \zeta_u^{\mathrm{T}} & \zeta_w^{\mathrm{T}} & \zeta_y^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{(m_1n+m_2n+pn)}, \\ M &= \begin{bmatrix} M_u & M_w & M_y \end{bmatrix} \in \mathbb{R}^{n\times(m_1n+m_2n+pn)}, \\ \bar{P} &= \bar{P}^{\mathrm{T}} \in \mathbb{R}^{(m_1n+m_2n+pn)\times(m_1n+m_2n+pn)} \\ &= \begin{bmatrix} M_u^{\mathrm{T}}PM_u & M_u^{\mathrm{T}}PM_w & M_u^{\mathrm{T}}PM_y \\ M_w^{\mathrm{T}}PM_u & M_w^{\mathrm{T}}PM_w & M_w^{\mathrm{T}}PM_y \\ M_y^{\mathrm{T}}PM_u & M_y^{\mathrm{T}}PM_w & M_y^{\mathrm{T}}PM_y \end{bmatrix}. \end{aligned}$$

By (3.29) we have obtained a new description of the cost function in terms of inputs and outputs of the system. The corresponding output feedback policies are given by,

$$u = K\begin{bmatrix} M_u & M_w & M_y \end{bmatrix}\begin{bmatrix} \zeta_u^{\mathrm{T}} & \zeta_w^{\mathrm{T}} & \zeta_y^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \triangleq \bar{K}z, \tag{3.30}$$

$$w = H\begin{bmatrix} M_u & M_w & M_y \end{bmatrix}\begin{bmatrix} \zeta_u^{\mathrm{T}} & \zeta_w^{\mathrm{T}} & \zeta_y^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \triangleq \bar{H}z, \tag{3.31}$$

where $\bar{K} = K\begin{bmatrix} M_u & M_w & M_y \end{bmatrix} \in \mathbb{R}^{m_1 \times (m_1 n + m_2 n + pn)}$ and $\bar{H} = H\begin{bmatrix} M_u & M_w & M_y \end{bmatrix} \in \mathbb{R}^{m_2 \times (m_1 n + m_2 n + pn)}$. Therefore, the optimal cost matrix is given by $\bar{P}^* = (\bar{P}^*)^{\mathrm{T}}$ and the corresponding optimal output feedback policies are,

$$u^* = \bar{K}^* z, \tag{3.32}$$

$$w^* = \bar{H}^* z. \tag{3.33}$$

Lemma 3.2. *The output feedback policies given by (3.32) and (3.33) converge to the optimal policies (3.25) and (3.26), respectively, which solve the zero-sum game problem.*

Proof: We can write the output feedback strategies as,

$$\bar{u}^* = \bar{K}^* z = K^* M z,$$

$$\bar{w}^* = \bar{L}^* z = L^* M z.$$

Applying the result in Theorem 2.5, we have,

$$\bar{x} = M z,$$

which converges exponentially to x, and thus,

$$\bar{u}^* = K^* x,$$

$$\bar{w}^* = L^* x,$$

which are the optimal strategies of the differential zero-sum game. □

The discussion so far focused on finding the nominal solution of the continuous-time H-infinity problem when the system model is known. In the following, we will derive an output feedback learning equation that will allow us to learn the optimal output feedback controller based on the input and output data instead of using full state feedback. To this end, consider the following Lyapunov function candidate,

$$V = x^{\mathrm{T}} P^i x. \tag{3.34}$$

Evaluating the derivative of (3.34) along the system trajectories and taking finite window integrals on

both sides, results in,

$$\begin{aligned} & x^{\mathrm{T}}(t)P^i x(t) - x^{\mathrm{T}}(t-T)P^i x(t-T) \\ = & \int_{t-T}^{t} x^{\mathrm{T}}(\tau)(A^{\mathrm{T}}P^i + P^i A)x(\tau)d\tau + 2\int_{t-T}^{t} u^{\mathrm{T}}(\tau)B^{\mathrm{T}}P^i x(\tau)d\tau + 2\int_{t-T}^{t} w^{\mathrm{T}}(\tau)E^{\mathrm{T}}P^i x(\tau)d\tau. \end{aligned} \tag{3.35}$$

We add $\int_{t-T}^{t} y^{\mathrm{T}}(\tau)Q_y y(\tau)d\tau$ on both sides of (3.35) so that Q can lumped up together with the unknown H^i. Then, we substitute $y(t) = Cx(t)$ and $H^i = A^{\mathrm{T}}P^i + P^i A$ on the right-hand side of the resulting equation, which gives us,

$$\begin{aligned} & x^{\mathrm{T}}(t)P^i x(t) - x^{\mathrm{T}}(t-T)P^i x(t-T) + \int_{t-T}^{t} y^{\mathrm{T}}(\tau)Q_y y(\tau)d\tau \\ = & \int_{t-T}^{t} x^{\mathrm{T}}(\tau)(A^{\mathrm{T}}P^i + P^i A + C^{\mathrm{T}}QC)x(\tau)d\tau + 2\int_{t-T}^{t} u^{\mathrm{T}}(\tau)B^{\mathrm{T}}P^i x(\tau)d\tau + 2\int_{t-T}^{t} w^{\mathrm{T}}(\tau)E^{\mathrm{T}}P^i x(\tau)d\tau. \end{aligned}$$

Next, we use the state parametrization (3.28) and substitute (3.29) in the above equation, which results in,

$$\begin{aligned} & z^{\mathrm{T}}(t)\bar{P}^i z(t) - z^{\mathrm{T}}(t-T)\bar{P}^i z(t-T) + \int_{t-T}^{t} y^{\mathrm{T}}(\tau)Q_y y(\tau)d\tau \\ = & \int_{t-T}^{t} z^{\mathrm{T}}(\tau)M^{\mathrm{T}}(A^{\mathrm{T}}P^i + P^i A + C^{\mathrm{T}}Q_y C)Mz(\tau)d\tau \\ & + 2\int_{t-T}^{t} u^{\mathrm{T}}(\tau)B^{\mathrm{T}}P^i Mz(\tau)d\tau + 2\int_{t-T}^{t} w^{\mathrm{T}}(\tau)E^{\mathrm{T}}P^i Mz(\tau)d\tau \end{aligned}$$

or more compactly,

$$\begin{aligned} & z^{\mathrm{T}}(t)\bar{P}^i z(t) - z^{\mathrm{T}}(t-T)\bar{P}^i z(t-T) + \int_{t-T}^{t} y^{\mathrm{T}}(\tau)Q_y y(\tau)d\tau \\ = & \int_{t-T}^{t} z^{\mathrm{T}}(\tau)\bar{H}^i z(\tau)d\tau + 2\int_{t-T}^{t} u^{\mathrm{T}}(\tau)B^{\mathrm{T}}P^i Mz(\tau)d\tau + 2\int_{t-T}^{t} w^{\mathrm{T}}(\tau)E^{\mathrm{T}}P^i Mz(\tau)d\tau, \end{aligned} \tag{3.36}$$

where,

$$\bar{H}^i \triangleq M^{\mathrm{T}}(A^{\mathrm{T}}P^i + P^i A + C^{\mathrm{T}}Q_y C)M.$$

To solve this linear system of equations, we define the following data matrices,

$$\begin{aligned} \delta_{zz} &= \left[z \otimes z|_{t_0}^{t_1}, z \otimes z|_{t_1}^{t_2}, \cdots, z \otimes z|_{t_{l-1}}^{t_l}\right]^{\mathrm{T}}, \\ I_{zu} &= \left[\int_{t_0}^{t_1} z \otimes u d\tau, \int_{t_1}^{t_2} z \otimes u d\tau, \cdots, \int_{t_{l-1}}^{t_l} z \otimes u d\tau\right]^{\mathrm{T}}, \\ I_{zw} &= \left[\int_{t_0}^{t_1} z \otimes w d\tau, \int_{t_1}^{t_2} z \otimes w d\tau, \cdots, \int_{t_{l-1}}^{t_l} z \otimes w d\tau\right]^{\mathrm{T}}, \end{aligned}$$

$$I_{zz} = \left[\int_{t_0}^{t_1} \bar{z}^{\mathrm{T}} d\tau, \int_{t_1}^{t_2} \bar{z}^{\mathrm{T}} d\tau, \cdots, \int_{t_{l-1}}^{t_l} \bar{z}^{\mathrm{T}} d\tau\right]^{\mathrm{T}},$$
$$I_{yy} = \left[\int_{t_0}^{t_1} y \otimes y d\tau, \int_{t_1}^{t_2} y \otimes y d\tau, \cdots, \int_{t_{l-1}}^{t_l} y \otimes y d\tau\right]^{\mathrm{T}}.$$

We can write (3.36) more compactly as,

$$\Psi \begin{bmatrix} \mathrm{vecs}(\bar{H}^i) \\ \mathrm{vec}(B^{\mathrm{T}} P^i M) \\ \mathrm{vec}(E^{\mathrm{T}} P^i M) \end{bmatrix} = \Gamma^i,$$

where the data matrices are given by,

$$\begin{aligned} \Psi &= [I_{zz}, 2I_{zu}, 2I_{zw}], \in \mathbb{R}^{l \times \left(\frac{N(N+1)}{2} + (m_1+m_2)N\right)}, \quad N = m_1 n + m_2 n + pn, \\ \Gamma^i &= \delta_{zz} \mathrm{vec}(\bar{P}^i) + I_{yy} \mathrm{vec}(Q_y) \in \mathbb{R}^l. \end{aligned}$$

The least-squares solution is given by,

$$\begin{bmatrix} \mathrm{vecs}(\bar{H}^i) \\ \mathrm{vec}(B^{\mathrm{T}} P^i M) \\ \mathrm{vec}(E^{\mathrm{T}} P^i M) \end{bmatrix} = ((\Psi)^{\mathrm{T}} \Psi)^{-1} (\Psi)^{\mathrm{T}} \Gamma^i. \tag{3.37}$$

Before introducing the output feedback value iteration algorithm, we recall the following definitions. Let $\{B_q\}_{q=0}^{\infty}$ be a collection of sets of norm limited $n \times n$ positive semidefinite matrices with nonempty interiors that satisfy $B_q \subseteq B_{q+1}$, $q \in \mathbb{Z}_+$ and $\lim_{q\to\infty} B_q = \mathcal{P}_+$, where $\mathcal{P}_+$ is a set of positive semidefinite matrices. We now propose the following output feedback based value iteration algorithm for the differential zero-sum game.

Algorithm 2: Output Feedback IRL Value Iteration Algorithm for Zero-Sum Game

Initialize. Choose $u^0 = \nu_1$ and $w^0 = \nu_2$. Initialize $\bar{P}_0 \geq 0$, and set $i \leftarrow 0$, $q \leftarrow 0$.

Acquire Data. Apply u^0 and w^0 during $t \in [t_0, t_l]$. Collect the filtered input-output data for each interval. Iterate for $i = 0, 1, \cdots,$

loop:

Solve the following learning equation for $\bar{H}^i$, $B^{\mathrm{T}}P^iM$ and $E^{\mathrm{T}}P^iM$,

$$\begin{aligned} & z^{\mathrm{T}}(t)\bar{P}^i z(t) - z^{\mathrm{T}}(t-T)\bar{P}^i z(t-T) + \int_{t-T}^{t} y^{\mathrm{T}}(\tau)Q_y y(\tau)d\tau \\ = \quad & \int_{t-T}^{t} z^{\mathrm{T}}(\tau)\bar{H}^i z(\tau)d\tau + 2\int_{t-T}^{t} u^{\mathrm{T}}(\tau)B^{\mathrm{T}}P^iMz(\tau)d\tau + 2\int_{t-T}^{t} w^{\mathrm{T}}(\tau)E^{\mathrm{T}}P^iMz(\tau)d\tau, \end{aligned}$$

Perform the following recursion,

$$\tilde{\bar{P}}^{i+1} \leftarrow \bar{P}^i + \epsilon_i(\bar{H}^i - M^{\mathrm{T}}P^iBB^{\mathrm{T}}P^iM + \gamma^{-2}M^{\mathrm{T}}P^iEE^{\mathrm{T}}P^iM).$$

if $\tilde{\bar{P}}^{i+1} \notin B_q$ **then**

$\bar{P}^{i+1} \leftarrow \bar{P}_0$ and $q \leftarrow q+1$.

else if $\left\|\tilde{\bar{P}}^{i+1} - \bar{P}^i\right\| / \epsilon_i < \varepsilon$ **then**

return $\bar{P}^i$ as the estimate of $\bar{P}^*$

else

$\bar{P}^{i+1} \leftarrow \tilde{\bar{P}}^{i+1}$

end if

$i \leftarrow i+1$

end loop.

It can be seen that Algorithm 2 does not require a stabilizing control policy for its initialization. The updates in the difference Riccati equation are performed with varying step size ϵ_i that satisfies $\epsilon_i > 0$, $\sum_{i=0}^{\infty}\epsilon_i = \infty$ and $\lim_{i\to\infty}\epsilon_i = 0$. We need to apply an exploration input in the data collection phase and collect $l \geq N(N+1)/2 + (m_1+m_1)N$ data samples to solve the least-squares problem (3.37). That is, the following rank condition needs to be satisfied,

$$\text{rank}\,(\Psi) = \frac{N(N+1)}{2} + (m_1+m_2)N. \tag{3.38}$$

Theorem 3.3. *Let (A,B) be controllable and $\left(A,\sqrt{Q_y}C\right)$ be observable. Then, the proposed output feedback algorithm Algorithm 2 generates a sequence $\bar{P}^i$ that converges to the optimal output feedback solution $\bar{P}^*$ as $i\to\infty$, provided that the rank condition* (3.38) *holds.*

Proof: Consider the following recursion in Algorithm 2,

$$\tilde{\bar{P}}^{i+1} = \quad \bar{P}^i + \epsilon_i(\bar{H}^i - M^{\mathrm{T}}P^iBB^{\mathrm{T}}P^iM + \gamma^{-2}M^{\mathrm{T}}P^iEE^{\mathrm{T}}P^iM).$$

Using the definitions of $\bar{P}$ and $\bar{H}$, we have,

$$\begin{aligned} M^{\mathrm{T}}\tilde{P}^{i+1}M \;=\; & M^{\mathrm{T}}P^iM + \epsilon_i(M^{\mathrm{T}}H^iM + M^{\mathrm{T}}C^{\mathrm{T}}Q_yCM \\ & -M^{\mathrm{T}}P^iBB^{\mathrm{T}}P^iM+\gamma^{-2}M^{\mathrm{T}}P^iEE^{\mathrm{T}}P^iM). \end{aligned}$$

Since (A, C) is observable, the state x can be uniquely determined from ω_i using the parametrization (3.28). Then, M must have full row rank. This reduces the above equation to

$$\tilde{P}^{i+1} = P^i + \epsilon_i(H^i + Q - P^iBB^{\mathrm{T}}P^i + \gamma^{-2}P^iEE^{\mathrm{T}}P^i),$$

where $Q = C^{\mathrm{T}}Q_yC$. The recursions on above equation converge to the GARE solution P^* under the controllability and observability conditions if H^i, $B^{\mathrm{T}}P^i$ and $E^{\mathrm{T}}P^i$ are the unique solution to the state feedback learning equation as shown in [13], which in turn requires the unique solution $\bar{H}^i$, $B^{\mathrm{T}}P^iM$ and $E^{\mathrm{T}}P^iM$ of the least-squares problem (3.37) associated with (3.36). The existence of the unique solution of (3.37) is guaranteed under the rank condition (3.38). This completes the proof. □

Exploration Bias Immunity of Algorithm 2:

We next show that the proposed IRL scheme does not lead to bias in the parameter estimates.

Theorem 3.4. *The excitation noise required to satisfy the excitation condition does not result in any bias in the parameter estimates.*

Proof: Consider the VI learning equation (3.36) with the excited inputs $\hat{u}$ and $\hat{w}$. Let $\hat{\bar{P}}^i$, $\hat{\bar{H}}^i$, $\hat{B}^{\mathrm{T}}\hat{P}^i\hat{M}$ and $\hat{E}^{\mathrm{T}}\hat{P}^i\hat{M}$ be the parameter estimates obtained as a result of these excited inputs. We have,

$$\begin{aligned} & z^{\mathrm{T}}(t)\hat{\bar{P}}^iz(t) - z^{\mathrm{T}}(t-T)\hat{\bar{P}}^iz(t-T) + \int_{t-T}^{t} y^{\mathrm{T}}(\tau)Q_yy(\tau)d\tau \\ = \quad & \int_{t-T}^{t} z^{\mathrm{T}}(\tau)\hat{\bar{H}}^iz(\tau)d\tau + 2\int_{t-T}^{t}\hat{u}^{\mathrm{T}}(\tau)\hat{B}^{\mathrm{T}}\hat{P}^i\hat{M}z(\tau)d\tau + 2\int_{t-T}^{t}\hat{w}^{\mathrm{T}}(\tau)\hat{E}^{\mathrm{T}}\hat{P}^i\hat{M}z(\tau)d\tau, \end{aligned}$$

Taking time derivative results in

$$\begin{aligned}
& 2z^{\mathrm{T}}(t)\hat{\bar{P}}^i\dot{z}(t) - 2z^{\mathrm{T}}(t\text{–}T)\hat{\bar{P}}^i\dot{z}(t\text{–}T) + y^{\mathrm{T}}(t)Q_y y(t) - y^{\mathrm{T}}(t\text{–}T)Q_y y(t\text{–}T) \\
= \quad & z^{\mathrm{T}}(t)\hat{\bar{L}}^i z(t) - z^{\mathrm{T}}(t\text{–}T)\hat{\bar{L}}z(t\text{–}T) + 2\hat{u}(t)\hat{B}^{\mathrm{T}}\hat{P}^i\hat{M}z(t) + \hat{w}(t)\hat{E}^{\mathrm{T}}\hat{P}^i\hat{M}z(t) \\
& +2\hat{u}(t\text{–}T)\hat{B}^{\mathrm{T}}\hat{P}^i\hat{M}z(t\text{–}T) + \hat{w}(t\text{–}T)\hat{E}^{\mathrm{T}}\hat{P}^i\hat{M}z(t\text{–}T)
\end{aligned}$$

Further expansion yields,

$$\begin{aligned}
& 2z^{\mathrm{T}}(t)\hat{\bar{P}}^i(\bar{\mathcal{A}}z(t) + \bar{\mathcal{B}}_1 u(t) + \bar{\mathcal{B}}_2 w(t) + \bar{\mathcal{B}}_3 y(t)) + 2z^{\mathrm{T}}(t)\hat{M}^{\mathrm{T}}\hat{P}^i\hat{M}\bar{\mathcal{B}}_1\nu_1(t) + 2z^{\mathrm{T}}(t)\hat{M}^{\mathrm{T}}\hat{P}^i\hat{M}\bar{\mathcal{B}}_2\nu_2(t) \\
& -2z^{\mathrm{T}}(t-T)\hat{\bar{P}}^i(\bar{\mathcal{A}}z(t-T) + \bar{\mathcal{B}}_1 u(t-T) + \bar{\mathcal{B}}_2 w(t-T) + \bar{\mathcal{B}}_3 y(t-T)) \\
& -2z^{\mathrm{T}}(t\text{–}T)\hat{M}^{\mathrm{T}}\hat{P}^i\hat{M}\bar{\mathcal{B}}_1\nu_1(t\text{–}T)\text{–}2z^{\mathrm{T}}(t\text{–}T)\hat{M}^{\mathrm{T}}\hat{P}^i\hat{M}\bar{\mathcal{B}}_2\nu_2(t\text{–}T) + y^{\mathrm{T}}(t)Q_y y(t) - y^{\mathrm{T}}(t\text{–}T)Q_y y(t\text{–}T) \\
= \quad & z^{\mathrm{T}}(t)\hat{\bar{H}}^i z(t) - z^{\mathrm{T}}(t\text{–}T)\hat{\bar{H}}z(t\text{–}T) + 2u^{\mathrm{T}}(t)\hat{B}^{\mathrm{T}}\hat{P}^i\hat{M}z(t) + w(t)\hat{E}^{\mathrm{T}}\hat{P}^i\hat{M}z(t) \\
& -2u^{\mathrm{T}}(t\text{–}T)\hat{B}^{\mathrm{T}}\hat{P}^i\hat{M}z(t\text{–}T) - w(t\text{–}T)\hat{E}^{\mathrm{T}}\hat{P}^i\hat{M}z(t\text{–}T) + 2\nu_1^{\mathrm{T}}(t)\hat{B}^{\mathrm{T}}\hat{P}^i\hat{M}z(t) + 2\nu_2(t)^{\mathrm{T}}\hat{E}^{\mathrm{T}}\hat{P}^i\hat{M}z(t) \\
& -2\nu_1^{\mathrm{T}}(t\text{–}T)\hat{B}^{\mathrm{T}}\hat{P}^i\hat{M}z(t\text{–}T) - 2\nu_2^{\mathrm{T}}(t\text{–}T)\hat{E}^{\mathrm{T}}\hat{P}^i\hat{M}z(t\text{–}T)
\end{aligned}$$

where we have combined the dynamics of z based on the input-output dynamics of the observer in which $\bar{\mathcal{A}}$ and $\bar{\mathcal{B}}_i$ represent the combined system dynamics and input matrices, respectively, corresponding to the dynamics of the filtering variables $\zeta' s$. Using the fact that $M\bar{\mathcal{B}}_1 = B$ and $M\bar{\mathcal{B}}_2 = E$, we have $2z^{\mathrm{T}}\hat{M}^{\mathrm{T}}\hat{P}^i\hat{M}\bar{\mathcal{B}}_1\nu_1 = 2\nu_1^{\mathrm{T}}\hat{B}^{\mathrm{T}}\hat{P}^i\hat{M}z$ and $2z^{\mathrm{T}}\hat{M}^{\mathrm{T}}\hat{P}^i\hat{M}\bar{\mathcal{B}}_2\nu_2 = 2\nu_2^{\mathrm{T}}\hat{E}^{\mathrm{T}}\hat{P}^i\hat{M}z$, thereby canceling the delayed and non-delayed ν_i terms. Thus, we have,

$$\begin{aligned}
& 2z^{\mathrm{T}}(t)\hat{\bar{P}}^i(\bar{\mathcal{A}}z(t) + \bar{\mathcal{B}}_1 u(t) + \bar{\mathcal{B}}_2 w(t) + \bar{\mathcal{B}}_3 y(t)) \\
& -2z^{\mathrm{T}}(t-T)\hat{\bar{P}}^i(\bar{\mathcal{A}}z(t-T) + \bar{\mathcal{B}}_1 u(t-T) + \bar{\mathcal{B}}_2 w(t-T) + \bar{\mathcal{B}}_3 y(t-T)) \\
& +y^{\mathrm{T}}(t)Q_y y(t) - y^{\mathrm{T}}(t\text{–}T)Q_y y(t\text{–}T) \\
= \quad & z^{\mathrm{T}}(t)\hat{\bar{H}}^i z(t) - z^{\mathrm{T}}(t\text{–}T)\hat{\bar{H}}z(t\text{–}T) + 2u^{\mathrm{T}}(t)\hat{B}^{\mathrm{T}}\hat{P}^i\hat{M}z(t) + w(t)\hat{E}^{\mathrm{T}}\hat{P}^i\hat{M}z(t) \\
& -2u^{\mathrm{T}}(t\text{–}T)\hat{B}^{\mathrm{T}}\hat{P}^i\hat{M}z(t\text{–}T) - w(t\text{–}T)\hat{E}^{\mathrm{T}}\hat{P}^i\hat{M}z(t\text{–}T)
\end{aligned}$$

Reversing the previous operations and performing integration result in

$$z^{\mathrm{T}}(t)\hat{\bar{P}}^i z(t) - z^{\mathrm{T}}(t-T)\hat{\bar{P}}^i z(t-T) + \int_{t-T}^{t} y^{\mathrm{T}}(\tau)Q_y y(\tau)d\tau$$

$$= \int_{t-T}^{t} z^{\mathrm{T}}(\tau)\hat{\bar{H}}^i z(\tau)d\tau + 2\int_{t-T}^{t} u^{\mathrm{T}}(\tau)\hat{B}^{\mathrm{T}}\hat{P}^i\hat{M}z(\tau)d\tau + 2\int_{t-T}^{t} w^{\mathrm{T}}(\tau)\hat{E}^{\mathrm{T}}\hat{P}^i\hat{M}z(\tau)d\tau.$$

Comparing the above equation with (3.36), we have $\hat{\bar{P}}^i = \bar{P}^i$, $\hat{\bar{H}}^i = \bar{H}^i$, $\hat{B}^{\mathrm{T}}\hat{P}^i\hat{M} = B^{\mathrm{T}}P^iM$ and $\hat{E}^{\mathrm{T}}\hat{P}^i\hat{M} = E^{\mathrm{T}}P^iM$. This establishes the bias-free property of Algorithm 2. □

3.4. Results

3.4.1. Example 1: H-infinity Control of F-16 Autopilot System:

In this section, we test the proposed design by numerical simulations of a model-free H-infinity autopilot controller for the F-16 aircraft.

Consider the discrete-time model of the F-16 aircraft autopilot from [4], which is in the form of (3.1) with

$$A = \begin{bmatrix} 0.906488 & 0.0816012 & -0.0005 \\ 0.0741349 & 0.90121 & -0.0007083 \\ 0 & 0 & 0.132655 \end{bmatrix}, B = \begin{bmatrix} -0.00150808 \\ -0.0096 \\ 0.867345 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, E = \begin{bmatrix} 0.00951892 \\ 0.00038373 \\ 0 \end{bmatrix}.$$

The three states are given by $x = [x_1, x_2, x_3]$, where x_1 is the angle of attack, x_2 is the rate of pitch, and x_3 is the elevator angle of deflection. The initial state of the system is $x = [10, 5, -2]$. The algorithm is initialized with $u_k^0 = n_k$ and $w_k^0 = v_k$. The PE condition is ensured by adding sinusoidal noise n_k and v_k of different frequencies and amplitudes in the inputs u_k and w_k, respectively. The system order is 3, so $N = 3$ is selected. We first compare the state feedback and output feedback model-free Q-learning schemes. The state feedback case has a smaller H matrix, but it requires the complete state feedback. On the other hand, the output feedback case has a larger H matrix, whose nominal values are

$$\begin{aligned}
H_{u\bar{u}}^* &= \begin{bmatrix} 0.0009 & -0.0006 & -0.0002 \end{bmatrix}, \\
H_{u\bar{w}}^* &= \begin{bmatrix} -0.0008 & 0.0087 & -0.0011 \end{bmatrix}, \\
H_{u\bar{y}}^* &= \begin{bmatrix} -0.9987 & 0.9443 & -0.1077 \end{bmatrix}, \\
H_{w\bar{u}}^* &= \begin{bmatrix} -0.0008 & 0.0005 & 0.0002 \end{bmatrix}, \\
H_{w\bar{w}}^* &= \begin{bmatrix} 0.0009 & -0.0084 & 0.0011 \end{bmatrix}, \\
H_{w\bar{y}}^* &= \begin{bmatrix} 0.9813 & -0.9132 & 0.1039 \end{bmatrix}, \\
H_{uu}^* &= 1.0009, H_{uw}^* = -0.0008, H_{ww}^* = -0.9990.
\end{aligned}$$

Consequently, the total time for output feedback parameters convergence is larger. Fig. 3.1 and Fig. 3.2 show the closed-loop state response and parameters convergence, respectively, under the state feedback Q-learning. The corresponding results for the proposed output feedback Q-learning scheme are shown in Fig. 3.3 and Fig. 3.4. In the state feedback Q-learning, we used 20 data samples in each iteration, while 70 data samples were used for the output feedback case, in order to satisfy the requirement of minimum number of data samples needed to satisfy the rank condition (3.16). It can be seen that the proposed output feedback Q-learning algorithm is able to maintain closed-loop stability and the controller parameters also converge to the optimal parameters as given below,

$$\begin{aligned}
\hat{H}_{u\bar{u}} &= \begin{bmatrix} 0.0009 & -0.0006 & -0.0002 \end{bmatrix}, \\
\hat{H}_{u\bar{w}} &= \begin{bmatrix} -0.0008 & 0.0087 & -0.0011 \end{bmatrix}, \\
\hat{H}_{u\bar{y}} &= \begin{bmatrix} -0.9980 & 0.9436 & -0.1076 \end{bmatrix}, \\
\hat{H}_{w\bar{u}} &= \begin{bmatrix} -0.0008 & 0.0005 & 0.0002 \end{bmatrix}, \\
\hat{H}_{w\bar{w}} &= \begin{bmatrix} 0.0009 & -0.0084 & 0.0011 \end{bmatrix}, \\
\hat{H}_{w\bar{y}} &= \begin{bmatrix} 0.9811 & -0.9130 & 0.1039 \end{bmatrix}, \\
\hat{H}_{uu} &= 1.0009, \hat{H}_{uw} = -0.0008, \hat{H}_{ww} = -0.9990.
\end{aligned}$$

The convergence criterion was chosen as $\varepsilon = 0.1$. Notice that no discounting factor was employed and the results correspond to those obtained by solving the game algebraic Riccati equation. Furthermore, the excitation noise did not introduce any bias in the estimates, which is an advantage of the proposed scheme. Moreover, the excitation noise is removed after the convergence of parameters estimates at $k = 3000$ for the state feedback case and $k = 14000$ for the output feedback case as shown in Fig. 3.1 and Fig. 3.3, respectively.

3.4.2. Example 2: Double Integrator with Disturbance:

We now test the IRL algorithm on a continuous-time system. For this, we consider the double integrator system with

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, E = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

The double integrator model represents a large class of practical systems including satellite attitude control

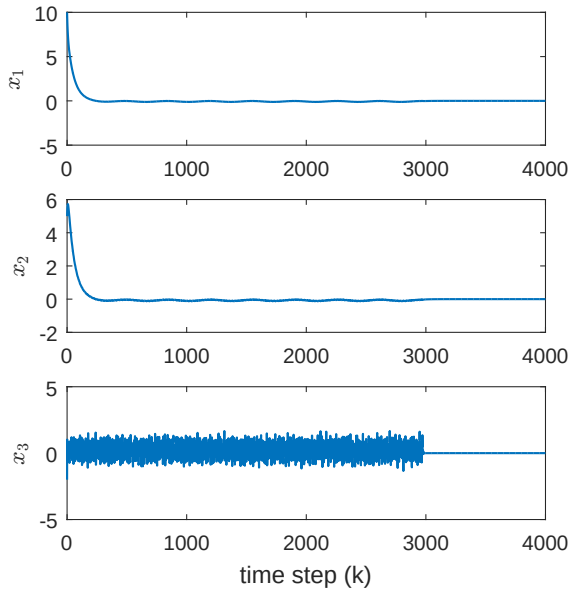

Fig. 3.1: Example 1: State trajectory of the closed-loop system under the state feedback Q-learning algorithm [4].

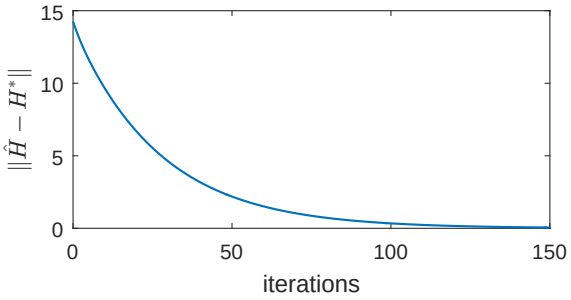

Fig. 3.2: Example 1: Convergence of the parameter estimates under the state feedback Q-learning algorithm [4].

and rigid body motion. Here u is the stabilizing control and w is a perturbation in both states. Both the state feedback and the proposed output feedback algorithms are evaluated. We choose the performance index parameters as $Q_y = 1$ and $\gamma = 3$. The eigenvalues of the observer matrix $\mathcal{A}$ are placed at -2. The optimal control parameters as found by solving the ARE (3.27) are,

$$P^* = \begin{bmatrix} 1.7997 & 1.4821 \\ 1.4821 & 2.0941 \end{bmatrix}, K^* = \begin{bmatrix} 1.0000 & 1.4142 \end{bmatrix},$$

$$M_u = \begin{bmatrix} 1 & 0 \\ 4 & 1 \end{bmatrix}, M_w = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, M_y = \begin{bmatrix} 4 & 4 \\ 0 & 4 \end{bmatrix}.$$

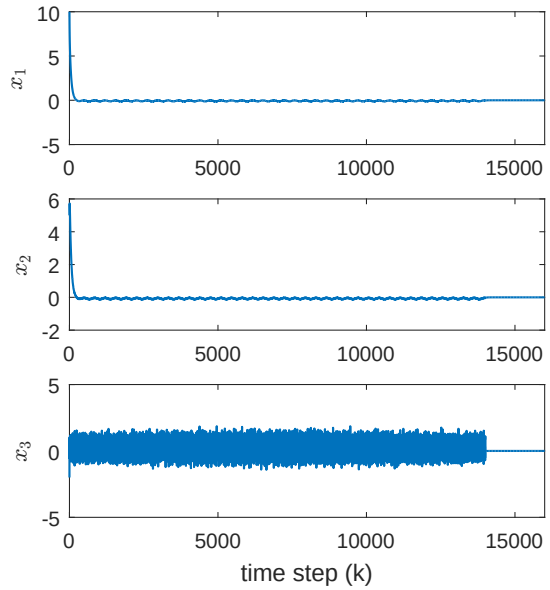

Fig. 3.3: Example 1: State trajectory of the closed-loop system under the output feedback Q-learning Algorithm 1.

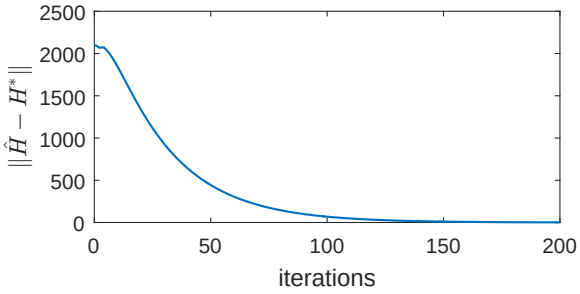

Fig. 3.4: Example 1: Convergence of the parameter estimates under the output feedback Q-learning Algorithm 1.

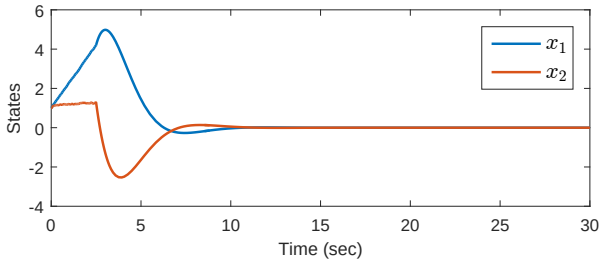

Fig. 3.5: Example 2: State trajectory of the closed-loop system under the state feedback IRL algorithm [13].

The initial controller parameters are set to zero. The learning period is $T = 0.05$s with a total of $l = 20$

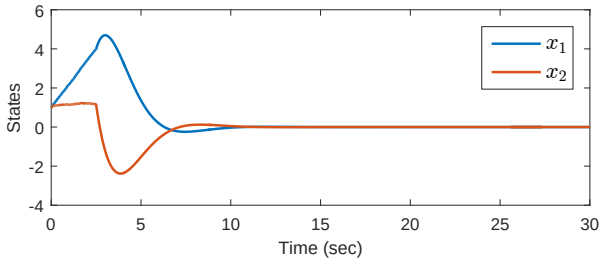

Fig. 3.6: Example 2: State trajectory of the closed-loop system under the output feedback IRL Algorithm 2.

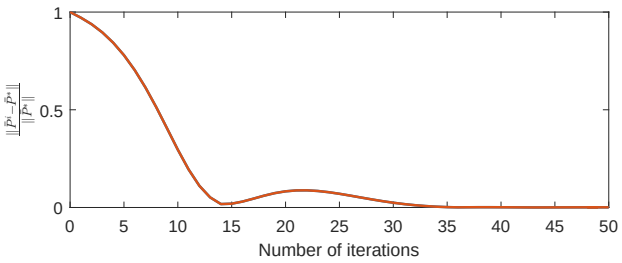

Fig. 3.7: Example 2: Convergence of the output feedback cost matrix $\bar{P}$ under the output feedback IRL Algorithm 2.

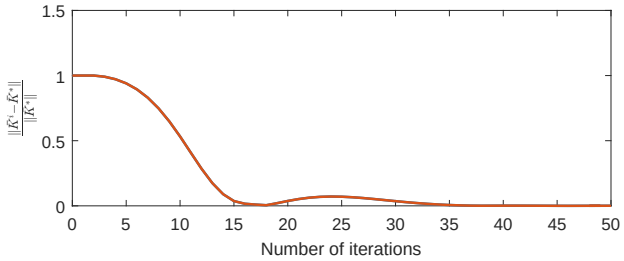

Fig. 3.8: Example 2: Convergence of the Player 1 output feedback policy $\bar{K}$ under the output feedback IRL Algorithm 2.

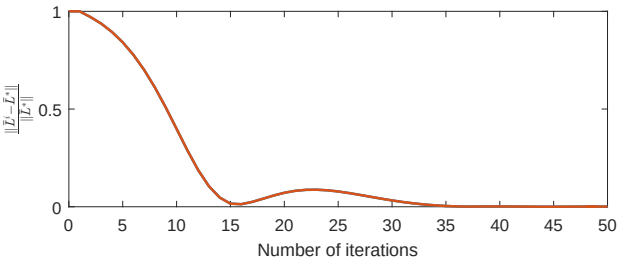

Fig. 3.9: Example 2: Convergence of the Player 2 output feedback policy $\bar{L}$ under the output feedback IRL Algorithm 2.

intervals. The GARE recursions are performed with the step size $\epsilon_i = (i^{0.2}+5)^{-1}, i = 0, 1, 2, \ldots$ and the set $B_q = \left\{\bar{P} \in \mathcal{P}_+^4 : |\bar{P}| \leq 200(q+1)\right\}, q = 0, 1, 2, \ldots$. Sinusoids of different frequencies were used to satisfy the rank condition (3.38). The state feedback results are shown in Fig. 3.5 and the proposed output feedback results are shown in Fig. 3.6. For comparison, we used the same learning period and applied the same excitation during the learning of both the state feedback and output feedback algorithms. It can be seen that the proposed output feedback algorithm gives performance quite close to that of its state feedback counterpart but without requiring the measurement of the system state. Furthermore, the VI algorithm does not require an initially stabilizing policy.

3.5. Summary

In this chapter, we have presented model-free output feedback RL algorithms to solve the zero-sum problem with application to the H-infinity control. We have developed an output feedback Q-function description for the linear quadratic zero-sum game using a parametrization of the state in terms of the past input, disturbance and output data. A Q-learning scheme based on value iteration is then developed, which learns the optimal output feedback Q-function and optimal output feedback policies for both players. For the continuous-time problem, we derive an output feedback learning equation corresponding to the linear quadratic zero-sum game based on integral reinforcement learning using a parametrization of the state given by the filtered input, disturbance and output data. Then, an integral reinforcement learning algorithm based on policy iteration is presented, which learns the optimal value function matrix and the optimal output feedback control parameters. It has been shown that the exploration noise does not result in bias in the parameter estimates. As a result, the need of employing a discounting factor in the cost function is eliminated and closed-loop stability is preserved. Hence, optimal stabilization under external disturbance is achieved without requiring the knowledge of the system dynamics or the internal state of the system. Simulation results have been presented that confirm the effectiveness of the proposed method.

4. Model-Free Stabilization Under Actuator Constraints

4.1. Introduction

A practical problem in designing control systems is to take into account the actuation capabilities of the actuators which apply the control inputs to the physical system. Actuator limitations are generally ignored in the design phase because many traditional control results do not remain applicable under such constraints. While some good progress has been made in the control of systems with perfectly known models and in the presence of actuator saturation, the problem becomes increasingly challenging in a model-free scenario. As such, model-free stabilization under actuator constraints still remains a largely unexplored area.

4.2. Q-learning Based State Feedback and Output Feedback Stabilization Under Actuator Constraints

Consider a discrete-time time-invariant system subject to actuator saturation, given by the following state-space representation,

$$\begin{aligned} x_{k+1} &= Ax_k + B\sigma(u_k), \\ y_k &= Cx_k, \end{aligned} \tag{4.1}$$

where $x_k = \left[x_k^1, x_k^2, \cdots, x_k^n\right]^{\mathrm{T}} \in \mathbb{R}^n$ refers to the state, $u_k = \left[u_k^1, u_k^2, \cdots, u_k^m\right]^{\mathrm{T}} \in \mathbb{R}^m$ is the input, and $y_k = \left[y_k^1, y_k^2, \cdots, y_k^p\right]^{\mathrm{T}} \in \mathbb{R}^p$ is the output. Without loss of generality, we assume that $\sigma : \mathbb{R}^m \to \mathbb{R}^m$ is a standard saturation function, that is, for $i = 1, 2, \cdots, m$,

$$\sigma(u_k^i) = \begin{cases} -b & \text{if } u_k^i < -b, \\ u_k^i & \text{if } -b \le u_k^i \le b, \\ b & \text{if } u_k^i > b, \end{cases} \tag{4.2}$$

with b being the saturation limit.

Assumption 4.1. *The pair (A, B) is asymptotically null controllable with bounded controls (ANCBC), i.e.,*

1) *(A, B) is stabilizable.*
2) *All eigenvalues of A are on or inside the unit circle.*

The objective in this problem is to find a stabilizing feedback control law that avoids saturation, and furthermore, the control law approaches a linear stabilizing one. Formally speaking, we would like to achieve global stabilization of the system subject to actuator saturation.

A model based solution to this problem was proposed in our previous work [42] that employs a parameterized algebraic Riccati equation (ARE) based approach to finding the low gain feedback matrix. The design process involves solving the following parameterized ARE,

$$A^{\mathrm{T}}P(\varepsilon)A - P(\varepsilon) + \varepsilon I - A^{\mathrm{T}}P(\varepsilon)B(B^{\mathrm{T}}P(\varepsilon)B + I)^{-1}B^{\mathrm{T}}P(\varepsilon)A = 0. \tag{4.3}$$

Based on the solution of the ARE (4.3), the following family of low gain feedback laws can be developed,

$$u_k = K^*(\varepsilon)x_k, \tag{4.4}$$

where $K^*(\varepsilon) = -(B^{\mathrm{T}}P^*(\varepsilon)B + I)^{-1}B^{\mathrm{T}}P^*(\varepsilon)A$ and $P^*(\varepsilon) > 0$ is the unique solution of the ARE, parameterized in the parameter $\varepsilon \in (0, 1]$ called the low gain parameter.

We recall the following lemma from [42],

Lemma 4.1. *Let Assumption 4.1 hold. Then, for each $\varepsilon \in (0, 1]$, there exists a unique positive definite matrix $P^*(\varepsilon)$ that solves the ARE (4.3). Moreover, such a $P^*(\varepsilon)$ satisfies,*

1) $\lim_{\varepsilon \to 0} P^*(\varepsilon) = 0$.
2) *There exists an $\varepsilon^* \in (0, 1]$ such that,*

$$\|P^*(\varepsilon)^{\frac{1}{2}}AP^*(\varepsilon)^{-\frac{1}{2}}\|_2 \leq \sqrt{2}, \varepsilon \in (0, \varepsilon^*].$$

Here $\|\cdot\|_2$ denotes the 2-norm of a matrix.

In [42], it has been shown that the family of low gain feedback control laws (4.4) achieve semi-global exponential stabilization of the system (4.1) without saturating the actuators. To this end, we recall the following result,

Theorem 4.1. *Consider the system (4.1). Under Assumption 4.1, for any* a priori *given (arbitrarily large) bounded set $\mathcal{W}$, there exists an $\varepsilon^* \in (0, 1]$ such that for any $\varepsilon \in (0, \varepsilon^*]$, the low gain feedback control law (4.4) renders the closed-loop system exponentially stable with $\mathcal{W}$ contained in the domain of attraction.*

The above low gain design technique is a model-based approach as it relies on solving the parameterized

ARE (4.3) which requires complete knowledge of the system dynamics. We will employ the method of reinforcement learning to learn the control law (4.4) without solving the ARE. Specifically, we will develop an iterative Q-learning scheme towards finding the low gain feedback controller. Both state feedback and output feedback algorithms will be presented.

In order to solve this problem under the framework of reinforcement learning, we will first formulate this constrained control problem as an optimal control problem so that we can apply the Bellman optimality principle that plays an instrumental role in reinforcement learning control. To this end, we define a quadratic utility function $r(x_k, u_k, \varepsilon) = \varepsilon x_k^{\mathrm{T}} x_k + u_k^{\mathrm{T}} u_k$. Next we define the following value function that we would like to minimize,

$$V = \sum_{i=k}^{\infty} r_i. \tag{4.5}$$

The parameterized Q-function can be expressed as

$$Q_K(z_k, \varepsilon) = z_k^{\mathrm{T}} H(\varepsilon) z_k, \tag{4.6}$$

where $z_k = (x_k^{\mathrm{T}}, u_k^{\mathrm{T}})^{\mathrm{T}}$ and the matrix $H(\varepsilon)$ is defined as

$$H(\varepsilon) \triangleq \begin{bmatrix} H_{xx} & H_{xu} \\ H_{ux} & H_{uu} \end{bmatrix} = \begin{bmatrix} \varepsilon I + A^{\mathrm{T}} P(\varepsilon) A & A^{\mathrm{T}} P(\varepsilon) B \\ B^{\mathrm{T}} P(\varepsilon) A & B^{\mathrm{T}} P(\varepsilon) B + I \end{bmatrix}.$$

The Q-learning objective is to estimate the parameterized Q-function matrix $H(\varepsilon)$. Once we have an estimate of $H(\varepsilon)$, we can compute the low gain control matrix $K(\varepsilon)$. We now propose an iterative Q-learning algorithm to find a control law that globally stabilizes the system and prevents saturation.

Algorithm 1: Iterative Q-learning Under Constrained Control Using State Feedback

1. Initialization. Start with any arbitrary policy $u_k = \nu_k$ with ν_k being the exploration signal and u_k satisfying the control constraint $\|u_k\|_\infty \leq b$, where $\|\cdot\|_\infty$ denotes the ∞-norm of a vector. Set $\varepsilon = 1$ and $H^0(\varepsilon) = 0$.

2. Collect Online Data. Apply the initial policy to collect L datasets of (x_k, u_k) for $k \in [0, L-1]$ along with their quadratic terms, where

$$L \geq (n+m)(n+m+1)/2.$$

Then for each iteration $i = 1, 2, \cdots$, perform:

3. Value Update. Evaluate the cost of the policy $K^i(\varepsilon)$ by solving for $H^i(\varepsilon)$ using the Bellman equation,

$$(\bar{H}^i(\varepsilon))^{\mathrm{T}} \bar{z}_k = \varepsilon x_k^{\mathrm{T}} x_k + u_k^{\mathrm{T}} u_k + (\bar{H}^{i-1}(\varepsilon))^{\mathrm{T}} \bar{z}_{k+1}. \tag{4.7}$$

4. Update Policy. Determine an updated policy $K^{i+1}(\varepsilon)$ using,

$$K^{i+1}(\varepsilon) = -(H_{uu}^i)^{-1} H_{ux}^i. \tag{4.8}$$

5. Repeat. Iterate on Steps 3 and 4 until the following convergence criterion is met,

$$\left\| K^{i+1}(\varepsilon) - K^i(\varepsilon) \right\|_2 < \delta,$$

for some small constant $\delta > 0$. Once the criterion is met, the resulting $\hat{K}(\varepsilon)$ is an approximation of $K^*(\varepsilon)$ for a given ε.

6. Control Saturation Check. For each $k = L, L+1, \cdots$, check the following saturation condition,

$$\|\hat{K}(\varepsilon) x_k\|_\infty \leq b.$$

If for any $k = L, L+1, \cdots$, the saturation condition is violated, reduce ε, reset $i = 1$, and repeat Steps 3 to 5 with a new ε.

7. Terminate. Stop when the control policy is no longer saturating.

The above algorithm presents a Q-learning value iteration algorithm for finding the low gain feedback control law that prevents saturation and achieves global stabilization. We start with a choice of low gain parameter $\varepsilon \in (0, 1]$ and select any arbitrary initial policy (not necessarily stabilizing), which contains an exploration signal and satisfies the saturation condition. This initial policy is referred to as the behavioral policy which is used to collect system data to be used in Step 3. The data collection is performed only once and we use the same dataset repeatedly to learn all future control policies. In the third step, we use the collected data to solve the Bellman Q-learning equation. Note that we use the previous $H^{i-1}(\varepsilon)$ in (4.7) and then update the policy in Step 4. These two steps are repeated for multiple iterations i until we no longer see updates on the policy, which leads to an optimal policy $K^*(\varepsilon)$ for a given ε. We then check this optimal policy $K^*(\varepsilon)$ for saturation. If the saturation condition holds good, that is, the control

is not saturating at $k = L$ then we apply $u_k = K^*(\varepsilon)x_k$ to the system, otherwise we reduce ε and repeat Steps 3 to 5 to continue search for a smaller control policy. Similar to Q-learning algorithms discussed in previous chapters, the following rank condition needs to be satisfied,

$$\text{rank}(\Phi) = (n+m)(n+m+1)/2. \tag{4.9}$$

This excitation condition can be met in several ways such as adding sinusoidal noise of various frequencies, exponentially decaying noise and gaussian noise. Theorem 4.2 states the convergence of the proposed algorithm.

Theorem 4.2. *Under the Assumption 4.1 and the rank condition (4.9), the proposed iterative Q-learning scheme globally stabilizes the system.*

The proof is carried out in two steps which correspond to the iteration on the low gain parameter ε and the value iteration on the matrix $H(\varepsilon)$. Let j be the iteration index corresponding to the iterations on ε. Then, we have $\varepsilon_j < \varepsilon_{j-1}$ with $\varepsilon_j \in (0, 1]$. By Lemma 4.1, there exists a unique $P^*(\varepsilon_j) > 0$ that satisfies the parameterized ARE (4.3) and by definition (4.6) there exists a unique $H^*(\varepsilon_j)$. By Theorem 4.1, for any given initial condition, there exists a $\varepsilon^* \in (0, 1]$ such that for all $\varepsilon_j \in (0, \varepsilon^*]$, the closed-loop system is exponentially stable. As j increases, we have $\varepsilon_j \in (0, \varepsilon^*]$. Under the stabilizability assumption on (A, B) and rank condition (2.26), the value iteration Steps 3 and 4 converge to the optimal H^* as shown in [35]. Therefore, $H^i(\varepsilon_j)$ converges to $H^*(\varepsilon_j)$, $\varepsilon_j \in (0, \varepsilon^*]$, as $i \to \infty$. Finally, by (4.8), we have the convergence of $K^i(\varepsilon_j)$ to $K^*(\varepsilon_j)$, $\varepsilon_j \in (0, \varepsilon^*]$. This completes the proof. □

Remark 4.1. *The proposed iterative scheme does not seek to find ε^*, rather it searches for an $\varepsilon \in (0, \varepsilon^*]$, which suffices to ensure closed-loop stability without saturating the actuators. However, better closed-loop performance could be obtained if the final ε is close to ε^*, which can be achieved by applying small decrements to ε in each iteration.*

Algorithm 1 solves the model-free global stabilization problem under actuator constraints using state feedback. In many applications, the complete state vector is not available for measurement, and therefore, output feedback methods are more desirable as they decrease the number of sensors and make the design more reliable and cost effective. We extend Algorithm 1 to the output feedback case by the method of state parametrization (2.17).

The parameterized output feedback Q-function is

$$\begin{aligned} Q_K(\zeta_k,\varepsilon) &= \begin{bmatrix} \bar{u}_{k-1,k-N} \\ \bar{y}_{k-1,k-N} \\ u_k \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathcal{H}_{\bar{u}\bar{u}} & \mathcal{H}_{\bar{u}\bar{y}} & \mathcal{H}_{\bar{u}u} \\ \mathcal{H}_{\bar{y}\bar{u}} & \mathcal{H}_{\bar{y}\bar{y}} & \mathcal{H}_{\bar{y}u} \\ \mathcal{H}_{u\bar{u}} & \mathcal{H}_{u\bar{y}} & \mathcal{H}_{uu} \end{bmatrix} \begin{bmatrix} \bar{u}_{k-1,k-N} \\ \bar{y}_{k-1,k-N} \\ u_k \end{bmatrix} \\ &\triangleq \zeta_k^{\mathrm{T}} \mathcal{H} \zeta_k, \end{aligned} \tag{4.10}$$

where,

$$\begin{aligned} \zeta_k &= \begin{bmatrix} \bar{u}_{k-1,k-N}^{\mathrm{T}} & \bar{y}_{k-1,k-N}^{\mathrm{T}} & u_k^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}, \\ \mathcal{H} &= \mathcal{H}^{\mathrm{T}} \in \mathbb{R}^{(mN+pN+m)\times(mN+pN+m)}, \end{aligned}$$

and the submatrices are given as

$$\begin{aligned} \mathcal{H}_{\bar{u}\bar{u}} &= M_u^{\mathrm{T}}(Q + A^{\mathrm{T}}P(\varepsilon)A)M_u \in \mathbb{R}^{mN\times mN}, \\ \mathcal{H}_{\bar{u}\bar{y}} &= M_u^{\mathrm{T}}(Q + A^{\mathrm{T}}P(\varepsilon)A)M_y \in \mathbb{R}^{mN\times pN}, \\ \mathcal{H}_{\bar{u}u} &= M_u^{\mathrm{T}}A^{\mathrm{T}}P(\varepsilon)B \in \mathbb{R}^{mN\times m}, \\ \mathcal{H}_{\bar{y}\bar{y}} &= M_y^{\mathrm{T}}(Q + A^{\mathrm{T}}P(\varepsilon)A)M_y \in \mathbb{R}^{pN\times pN} \\ \mathcal{H}_{\bar{y}u} &= M_y^{\mathrm{T}}A^{\mathrm{T}}P(\varepsilon)B \in \mathbb{R}^{pN\times m}, \\ \mathcal{H}_{uu} &= R + B^{\mathrm{T}}P(\varepsilon)B \in \mathbb{R}^{m\times m}. \end{aligned} \tag{4.11}$$

Let $\mathcal{H}^*$ be the optimal matrix corresponding to $P^*(\varepsilon)$ for a given low gain parameter ε. Then the optimal output feedback controller is given by

$$\begin{aligned} u_k^* &= -(\mathcal{H}_{uu}^*)^{-1}\left(\mathcal{H}_{u\bar{u}}^*\bar{u}_{k-1,k-N} + \mathcal{H}_{u\bar{y}}^*\bar{y}_{k-1,k-N}\right) \\ &\triangleq \mathcal{K}^*(\varepsilon)\begin{bmatrix} \bar{u}_{k-1,k-N}^{\mathrm{T}} & \bar{y}_{k-1,k-N}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \end{aligned} \tag{4.12}$$

We now proceed to develop an output feedback Q-learning equation which enables us to learn the output feedback controller online. Using the output feedback Q-function (4.10) in the Q-learning equation (4.13) results in,

$$\zeta_k^{\mathrm{T}}\mathcal{H}(\varepsilon)\zeta_k = \varepsilon y_k^{\mathrm{T}}y_k + u_k^{\mathrm{T}}u_k + \zeta_{k+1}^{\mathrm{T}}\mathcal{H}(\varepsilon)\zeta_{k+1}, \tag{4.13}$$

which can be linearly parameterized as

$$Q_K = \bar{\mathcal{H}}^{\mathrm{T}}(\varepsilon)\bar{\zeta}_k, \tag{4.14}$$

where $\bar{\mathcal{H}}(\varepsilon) = \text{vec}(\mathcal{H}(\varepsilon))$ and $\bar{\zeta}_k = \zeta_k \otimes \zeta_k$. With the above parametrization, we have the following linear equation unknown in $\bar{\mathcal{H}}$,

$$\bar{\mathcal{H}}^{\mathrm{T}}(\varepsilon)\bar{\zeta}_k = \varepsilon y_k^{\mathrm{T}} y_k + u_k^{\mathrm{T}} u_k + \bar{\mathcal{H}}^{\mathrm{T}}(\varepsilon)\bar{\zeta}_{k+1}. \tag{4.15}$$

Based on (4.15), we can utilize the Q-learning technique to learn the output feedback Q-function matrix $\mathcal{H}$. We now present an iterative Q-learning algorithm to find a output feedback control law that globally stabilizes the system and prevents saturation.

Algorithm 2: Iterative Q-learning Under Constrained Control Using Output Feedback

1. Initialization. Start with any arbitrary policy $u_k^0 = \nu_k$ with ν_k being the exploration signal and u_k satisfying the control constraint $\|u_k\|_\infty \leq b$. Set $\varepsilon = 1$ and $\mathcal{H}^0(\varepsilon) = 0$.

2. Collect Online Data. Apply the initial policy to collect L datasets of $(\bar{u}_{k-1,k-N}, \bar{y}_{k-1,k-N}, u_k)$ for $k \in [0, L-1]$ along with their quadratic terms, where

$$L \geq (mN + pN + m)(mN + pN + m + 1)/2.$$

Then for each iteration $i = 1, 2, \cdots$, perform:

3. Value Update. Evaluate the cost of the output feedback policy by solving for $\mathcal{H}^i(\varepsilon)$ using the Bellman equation,

$$(\bar{\mathcal{H}}^i(\varepsilon))^{\mathrm{T}}\bar{\zeta}_k = \varepsilon y_k^{\mathrm{T}} y_k + u_k^{\mathrm{T}} u_k + (\bar{\mathcal{H}}^{i-1}(\varepsilon))^{\mathrm{T}}\bar{\zeta}_{k+1}. \tag{4.16}$$

4. Update Policy. Determine an updated policy $\mathcal{K}^{i+1}(\varepsilon)$ using,

$$\mathcal{K}^{i+1}(\varepsilon) = -(\mathcal{H}_{uu}^i)^{-1}\begin{bmatrix}\mathcal{H}_{u\bar{u}}^i & \mathcal{H}_{u\bar{y}}^i\end{bmatrix}. \tag{4.17}$$

5. Repeat. Iterate on Steps 3 and 4 until the following convergence criterion is met,

$$\left\|\mathcal{K}^{i+1}(\varepsilon) - \mathcal{K}^i(\varepsilon)\right\|_2 < \delta,$$

for some small constant $\delta > 0$. Once the criterion is met, the resulting $\hat{\mathcal{K}}(\varepsilon)$ is an approximation of

$\mathcal{K}^*(\varepsilon)$ for a given ε.

6. Control Saturation Check. For each $k = L, L+1, \cdots$, check the following saturation condition,

$$\|\hat{\mathcal{K}}(\varepsilon) \begin{bmatrix} \bar{u}_{k-1,k-N}^{\mathrm{T}} & \bar{y}_{k-1,k-N}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \|_\infty \leq b.$$

If for any $k = L, L+1, \cdots$, the saturation condition is violated, reduce ε, reset $i = 1$, and repeat Steps 3 to 5 with a new ε.

7. Terminate. Stop when the control policy is no longer saturating.

Compared to Algorithm 1, Algorithm 2 solves the output feedback Bellman equation (4.16). The output feedback Q-function matrix $\mathcal{H}$ has $(mN + pN + m)(mN + pN + m + 1)/2$ unknowns. In the output feedback case the data matrices $\Phi \in \mathbb{R}^{(mN+pN+m)(mN+pN+m+1)/2 \times L}$ and $\Upsilon^{i-1} \in \mathbb{R}^{L \times 1}$ are defined by

$$\begin{aligned} \Phi &= \left[\bar{\zeta}_k^1, \bar{\zeta}_k^2, \cdots, \bar{\zeta}_k^L \right], \\ \Upsilon^{i-1} &= \left[r_k^1 + (\bar{\mathcal{H}}^{i-1}(\varepsilon))^{\mathrm{T}} \bar{\zeta}_{k+1}^1, \cdots, r_k^L + (\bar{\mathcal{H}}^{i-1}(\varepsilon))^{\mathrm{T}} \bar{\zeta}_{k+1}^L \right]^{\mathrm{T}}. \end{aligned}$$

Then the least-squares solution of (4.16) is given by

$$\bar{\mathcal{H}}^i(\varepsilon) = (\Phi \Phi^{\mathrm{T}})^{-1} \Phi \Upsilon^{i-1}. \tag{4.18}$$

Similar to Algorithm 1, an excitation signal v_k is added in the control during the learning phase to satisfy the following rank condition,

$$\text{rank}(\Phi) = (mN + pN + m)(mN + pN + m + 1)/2. \tag{4.19}$$

Theorem 4.3 shows the convergence of Algorithm 2.

Theorem 4.3. *Under the Assumption 4.1, together with the observability of (A, C) and the rank condition (4.19), the proposed output feedback based iterative Q-learning scheme globally stabilizes the system.*

Proof: According to Theorem 2.2 in Chapter 2, the state x_k can be reconstructed by the past input and output sequence as given by (2.17) if the pair (A, C) is observable. Thus, the output feedback and

state feedback Q-learning equations are equivalent. Under the rank condition (4.19), the output feedback Q-learning equation has a unique solution. Then, following the arguments in Theorem 4.2, the output feedback Algorithm 2 also achieves global stabilization of the system by finding a suitable low gain parameter ε and the corresponding control matrix $\mathcal{K}(\varepsilon)$ without saturating the actuators. This completes the proof. □

4.3. Integral Reinforcement Learning Based State Feedback and Output Feedback Stabilization Under Actuator Constraints

In this section, we will extend the low gain feedback framework to solve the continuous-time counterpart of the actuator constraint stabilization problem. We present integral reinforcement learning equations together with a low gain scheduling mechanism to achieve global stabilization of a class of continuous-time linear systems that satisfy the ANCBC assumption.

Consider a continuous-time system subject to actuator constraints,

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B\sigma(u(t)), \\ y(t) &= Cx(t), \end{aligned} \tag{4.20}$$

where $x = [x_1, x_2, \cdots, x_n]^{\mathrm{T}} \in \mathbb{R}^n$ is the state vector, $u = [u_1, u_2, \cdots, u_m]^{\mathrm{T}} \in \mathbb{R}^m$ is the control vector and $y = [y_1, y_2, \cdots, y_p]^{\mathrm{T}} \in \mathbb{R}^p$ is the output vector. Without loss of generality, we assume that $\sigma : \mathbb{R}^m \to \mathbb{R}^m$ is a standard saturation function, that is, for $i = 1, 2, \cdots, m$,

$$\sigma(u_i) = \begin{cases} -b & \text{if } u_i < -b, \\ u_i & \text{if } -b \le u_i \le b, \\ b & \text{if } u_i > b, \end{cases} \tag{4.21}$$

where b denotes the actuator limit.

Assumption 4.2. *The pair (A, B) is asymptotically null controllable with bounded controls (ANCBC), i.e.,*

1) *(A, B) is stabilizable.*
2) *All eigenvalues of the system matrix A are in the closed left-half s-plane.*

Assumption 4.3. *The pair (A, C) is observable.*

Condition 1) is a standard requirement for stabilization. Condition 2) is a necessary condition for global stabilization in the presence of actuator saturation [56]. Note that this condition allows systems to be polynomially unstable, that is, systems that have repeated poles on the imaginary axis. Furthermore, as is known in the literature on control systems with actuator saturation, when there is an eigenvalue of A in the open right-half plane, global or even semi-global stabilization is not achievable. Assumption 4.3 is needed for output feedback control.

To solve this problem, we proposed a parameterized ARE method to find the low gain control matrix in our previous work [42] based on the solution of the following parameterized ARE,

$$A^{\mathrm{T}}P(\gamma) + P(\gamma)A + \gamma I - P(\gamma)BB^{\mathrm{T}}P(\gamma) = 0, \gamma \in (0, 1]. \tag{4.22}$$

The resulting family of parameterized low gain feedback control laws is given by,

$$u = K^*(\gamma)x(t), \tag{4.23}$$

where

$$K^*(\gamma) = -B^{\mathrm{T}}P^*(\gamma)$$

and $P^*(\gamma) > 0$ is the unique positive definite solution of the ARE (4.22), parameterized in the low gain parameter $\gamma \in (0, 1]$.

To this end, we recall some previous results from [42].

Lemma 4.2. *Under Assumption 4.2, for each $\gamma \in (0, 1]$, there exists a unique positive definite solution $P^*(\gamma)$ to the ARE (4.22) that satisfies $\lim_{\gamma \to 0} P^*(\gamma) = 0$.*

Theorem 4.4. *Let Assumption 4.2 hold. Then, for any* a priori *given (arbitrarily large) bounded set $\mathcal{W}$, there exists a γ^* such that for any $\gamma \in (0, \gamma^*]$, the low gain feedback control law (4.23) renders the closed-loop system exponentially stable with $\mathcal{W}$ contained in the domain of attraction.*

The above discussion focused on the model based solution of the continuous-time actuator constraint problem. In the next sections, we will present the low gain parameterized version of the IRL equations that we presented in Chapter 2.

Consider the Lyapunov function candidate parameterized in a low gain parameter $\gamma \in (0, 1]$,

$$V = x^{\mathrm{T}} P(\gamma) x. \tag{4.24}$$

Evaluating the derivative of (4.24) along the system trajectories,

$$\begin{aligned}\frac{d}{dt}(x^{\mathrm{T}} P(\gamma) x) &= (Ax + B\sigma(u))^{\mathrm{T}} P(\gamma) x + x^{\mathrm{T}} P(\gamma)(Ax + B\sigma(u)) \\ &= x^{\mathrm{T}} H(\gamma) x - 2\sigma^{\mathrm{T}}(u) K(\gamma) x,\end{aligned}$$

where $H(\gamma) = A^{\mathrm{T}} P(\gamma) + P^{\mathrm{T}}(\gamma) A$ and $K(\gamma) = -B^{\mathrm{T}} P(\gamma)$. We are interested in finding the unknown matrices $H(\gamma)$ and $K(\gamma)$ without requiring the knowledge of (A, B).

Performing finite window integrals of length $T > 0$ on both sides results in the following learning equation,

$$\begin{aligned}&x^{\mathrm{T}}(t) P(\gamma) x(t) - x^{\mathrm{T}}(t - T) P(\gamma) x(t - T) \\ &= \int_{t-T}^{t} x^{\mathrm{T}}(\tau) H(\gamma) x(\tau) d\tau - 2 \int_{t-T}^{t} \sigma^{\mathrm{T}}(u(\tau)) K(\gamma) x(\tau) d\tau,\end{aligned} \tag{4.25}$$

or equivalently,

$$\begin{aligned}&x^{\mathrm{T}}(t) \otimes x^{\mathrm{T}}(t)|_{t-T}^{t} \mathrm{vec}(P(\gamma)) \\ &= \int_{t-T}^{t} \bar{x} d\tau \ \mathrm{vecs}(H(\gamma)) - 2 \int_{t-T}^{t} x^{\mathrm{T}}(\tau) \otimes \sigma^{\mathrm{T}}(u(\tau)) d\tau \ \mathrm{vec}(K(\gamma)),\end{aligned}$$

where $\bar{x} = \{x_1^2, 2x_1x_2, \cdots, x_2^2, 2x_2x_3, \cdots, x_n^2\}$ and $\mathrm{vecs}(H(\gamma)) = \{H_{11}, H_{12}, \cdots, H_{1n}, H_{22}, H_{23}, \cdots, H_{nn}\}$.

Equation (4.25) is the low gain parameterized version of the IRL equation presented in Chapter 2. It is a scalar equation which is linear in terms of the unknowns $H(\gamma)$ and $K(\gamma)$. Clearly, there are more unknowns than the number of equations. Therefore, we develop a system of l number of such equations by performing l finite window integrals each of length T. To solve this linear system of equations, we define the following data matrices as done in [14],

$$\begin{aligned}\delta_{xx} &= \left[x \otimes x|_{t_0}^{t_1}, x \otimes x|_{t_1}^{t_2}, \cdots, x \otimes x|_{t_{l-1}}^{t_l}\right]^{\mathrm{T}}, \\ I_{xu} &= \left[\int_{t_0}^{t_1} x \otimes \sigma(u) d\tau, \int_{t_1}^{t_2} x \otimes \sigma(u) d\tau, \cdots, \int_{t_{l-1}}^{t_l} x \otimes \sigma(u) d\tau\right]^{\mathrm{T}}, \\ I_{xx} &= \left[\int_{t_0}^{t_1} \bar{x}^{\mathrm{T}} d\tau, \int_{t_1}^{t_2} \bar{x}^{\mathrm{T}} d\tau, \cdots, \int_{t_{l-1}}^{t_l} \bar{x}^{\mathrm{T}} d\tau\right].\end{aligned}$$

We rewrite Equation (4.25) as the following matrix equation,

$$\Big[I_{xx}, 2I_{xu}\Big] \begin{bmatrix} \text{vecs}(H(\gamma)) \\ \text{vec}(K(\gamma)) \end{bmatrix} = \delta_{xx}\text{vec}(P(\gamma)),$$

whose least-squares solution is given by,

$$\begin{bmatrix} \text{vecs}(H(\gamma)) \\ \text{vec}(K(\gamma)) \end{bmatrix} = \Bigg(\Big[I_{xx},\ 2I_{xu}\Big]^{\text{T}} \Big[I_{xx},\ 2I_{xu}\Big] \Bigg)^{-1} \Big[I_{xx},\ 2I_{xu}\Big]^{\text{T}} \delta_{xx}\text{vec}(P(\gamma)) \tag{4.26}$$

Remark 4.2. *Equation* (4.26) *is a standard least-squares problem in which the two unknown matrices are $H(\gamma)$ and $K(\gamma)$, which have $n(n+1)/2$ and mn unknown components, respectively. Therefore, we need $l \geq n(n+1)/2 + mn$ number of datasets (learning intervals) to solve (4.26). Notice that the control $u = Kx$ linearly depends on the state x, which means that $\Big[I_{xx},\ 2I_{xu}\Big]^{T}\Big[I_{xx},\ 2I_{xu}\Big]$ will not be invertible. A common practice to address this difficulty is to add an exploration noise signal ν in the control during the data collection step. These two conditions boil down to satisfying the following rank condition in order to have a unique solution to* (4.26)*,*

$$\textit{rank}\Big(\Big[I_{xx},\ 2I_{xu}\Big] \Big) = n(n+1)/2 + mn. \tag{4.27}$$

Some common exploration signals include sinusoidal and Gaussian signals. It should also be noted that these exploration signals do not incur bias in the estimates, as has been shown in Chapter 2. Morever, once the learning is accomplished, these signals can be safely removed.

We are now ready to propose our low gain scheduled IRL algorithm that achieves model-free global stabilization of the system in the presence of actuator saturation. The stabilization is achieved by preventing saturation under the low gain feedback.

Algorithm 3: Iterative State Feedback IRL Scheme Under Constrained Control

1. Initialization. Start with an arbitrary open-loop policy $u^0 = \nu$ with ν being the exploration signal and u^0 satisfying the control constraint $\|u\|_\infty \leq b$, where $\|\cdot\|_\infty$ denotes the ∞ norm of a vector. Set $\gamma \leftarrow 1$, $k \leftarrow 0$, $q \leftarrow 0$, and $P_0(\gamma) > 0$.

2. Collect Online Data. Apply the policy u^0 for the duration $t \in [t_0, t_l]$, composed of l learning intervals of length $t_j - t_{j-1} = T$, where $t_l = t_0 + lT$ and $l \geq n(n+1)/2 + mn$. Collect the data for x and its quadratic integrals for each learning interval. Then, for $k = 0, 1, \cdots$, perform,

3. Loop:

Determine the least-squares solution, $H_k(\gamma)$ and $K_k(\gamma)$, of the following learning equation,

$$\begin{aligned} & x^{\mathrm{T}}(t)P_k(\gamma)x(t) - x^{\mathrm{T}}(t-T)P_k(\gamma)x(t-T) \\ = & \int_{t-T}^{t} x^{\mathrm{T}}(\tau)H_k(\gamma)x(\tau)d\tau - 2\int_{t-T}^{t} \sigma^{\mathrm{T}}(u(\tau))K_k(\gamma)x(\tau)d\tau. \end{aligned}$$

Update $\tilde{P}(\gamma)$ as,

$$\tilde{P}_{k+1}(\gamma) \leftarrow P_k(\gamma) + \epsilon_k(H_k(\gamma) + \gamma I - K_k^{\mathrm{T}}(\gamma)K_k(\gamma)).$$

if $\tilde{P}_{k+1}(\gamma) \notin B_q$ **then**

$$P_{k+1}(\gamma) \leftarrow P_0(\gamma) \text{ and } q \leftarrow q+1.$$

else if $\left\| \tilde{P}_{k+1}(\gamma) - P_k(\gamma) \right\| / \epsilon_k < \varepsilon$ **then**

return $P_k(\gamma)$ as an estimate of $P^*(\gamma)$.

else

$$P_{k+1}(\gamma) \leftarrow \tilde{P}_{k+1}(\gamma).$$

end if

$$k \leftarrow k+1.$$

end loop.

The resulting $\hat{K}(\gamma)$ is an approximation of $K^*(\gamma)$ for a given γ.

4. Control Saturation Check. For all $t \geq t_f$, check the control constraints,

$$\|\hat{K}(\gamma)x\|_\infty \leq b.$$

If for any $t \geq t_f$ the control constraint does not hold, reduce γ, reset $i = 1$, and repeat Step 3 with a reduced γ.

5. Terminate.

The details of Algorithm 3 are as follows. The algorithm is initialized with some low gain parameter $\gamma \in (0, 1]$ and an arbitrary control policy comprising of exploration signals ν is used to generate system data. Note that this initial policy is open-loop and not necessarily stable. Furthermore, it is selected so that the control constraints are satisfied. The trajectory data is used to solve the learning equation in the Step 3 where an stabilizing low gain control matrix is obtained corresponding to our choice of low gain parameter. However, this control policy needs to be checked for control constraints to ensure that the low gain parameter is the appropriate one $\gamma \in (0, 1]$, which is done in Step 4. The low gain parameter γ can be updated under a proportional rule $\gamma_{j+1} = a\gamma_j$ for $0 < a < 1$ in the future iterations.

Theorem 4.5 states the convergence of the proposed Algorithm 3.

Theorem 4.5. *Under Assumption 4.2, the proposed iterative Algorithm 3 globally stabilizes the system if the rank condition* (4.27) *is satisfied.*

Proof: The proof is carried out in two steps, which correspond to the iterations on the low gain parameter γ and the value iterations on the matrix $K(\gamma)$ for a given low gain parameter γ. We consider the iterations on the low gain parameter γ to show that there exists $\gamma \in (0, 1]$ such that the low gain feedback law $u = -B^{\mathrm{T}}P^*(\gamma)x$ renders the closed-loop system exponentially stable. To this end, we consider system (4.20) in the following closed-loop form,

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B\sigma(u(t)) \\ &= (A - BB^{\mathrm{T}}P^*(\gamma))x(t) + B\left(\sigma(u(t)) - u(t)\right). \end{aligned} \tag{4.28}$$

We choose a candidate Lyapunov function

$$V(x) = x^{\mathrm{T}} P^*(\gamma) x. \tag{4.29}$$

For the given initial condition $x(0)$, let $\mathcal{W}$ be a bounded set that contains $x(0)$ and let $c > 0$ be a constant such that

$$c \geq \sup_{x \in \mathcal{W}, \gamma \in (0,1]} x^{\mathrm{T}} P^*(\gamma) x. \tag{4.30}$$

Such a c exists because $\mathcal{W}$ is bounded and $\lim_{\gamma \to 0} P^*(\gamma) = 0$ by Lemma 4.2. Let us define a level set $L_V(c) = \{x \in \mathbb{R}^n \mid V(x) \leq c\}$ and let γ^* be such that for all $\gamma \in (0, \gamma^*]$, $x \in L_V(c)$ implies $\|B^{\mathrm{T}} P^*(\gamma) x\| \leq b$. To see that such a γ^* exists, we note that

$$\begin{aligned} \|B^{\mathrm{T}} P^*(\gamma) x\| &= \|B^{\mathrm{T}} (P^*(\gamma))^{1/2} (P^*(\gamma))^{1/2} x\| \\ &\leq \|B^{\mathrm{T}} (P^*(\gamma))^{1/2}\| \sqrt{c}, \end{aligned}$$

where we have used the inequality (4.30). Since $\lim_{\gamma \to 0} P^*(\gamma) = 0$ by Lemma 4.2, the term involving $P^*(\gamma)$ can be made arbitrary small, and therefore, there exists a $\gamma^* \in (0, 1]$ such that $\|(B^{\mathrm{T}} P^*(\gamma) x\| \leq b$ for all $\gamma \in (0, \gamma^*]$. In other words, we can always find a γ that makes the above norm small enough so that the control operates in the linear region of the saturation curve. The evaluation of the derivative of V along the trajectories of the closed-loop system (4.28) shows that, for all $x \in L_V(c)$,

$$\begin{aligned} \dot{V} &= x^{\mathrm{T}} P^*(\gamma) \big((A - BB^{\mathrm{T}} P^*(\gamma)) x + B(\sigma(u) - u) \big) \\ &\quad + \big((A - BB^{\mathrm{T}} P^*(\gamma)) x + B(\sigma(u) - u) \big)^{\mathrm{T}} P^*(\gamma) x \\ &= x^{\mathrm{T}} \big(A^{\mathrm{T}} P^*(\gamma) + P^*(\gamma) A \big) x - 2 x^{\mathrm{T}} P^*(\gamma) B B^{\mathrm{T}} P^*(\gamma) x \\ &\quad + 2 x^{\mathrm{T}} P^*(\gamma) B \big(\sigma(u) - u \big) \end{aligned}$$

From the ARE (4.22), we have,

$$A^{\mathrm{T}} P^*(\gamma) + P^*(\gamma) A = -\gamma I + P^*(\gamma) B B^{\mathrm{T}} P^*(\gamma).$$

It then follows that,

$$\dot{V} = x^{\mathrm{T}} \big(-\gamma I + P^*(\gamma) B B^{\mathrm{T}} P^*(\gamma) \big) x - 2 x^{\mathrm{T}} P^*(\gamma) B B^{\mathrm{T}} P^*(\gamma) x$$

$$
\begin{aligned}
&+2x^{\mathrm{T}}P^*(\gamma)B\big(\sigma(u)-u\big)\\
&=-\gamma x^{\mathrm{T}}x-x^{\mathrm{T}}P^*(\gamma)BB^{\mathrm{T}}P^*(\gamma)x\\
&+2x^{\mathrm{T}}P^*(\gamma)B(\sigma(u)-u)
\end{aligned}
$$

Since $\|B^{\mathrm{T}}P^*(\gamma)x\| \leq b$, it follows from the definition of the saturation function (4.2) that $\sigma(u) = u$, which results in,

$$
\begin{aligned}
\dot{V} &= -\gamma x^{\mathrm{T}}x - x^{\mathrm{T}}P^*(\gamma)BB^{\mathrm{T}}P^*(\gamma)x\\
&\leq -\gamma x^{\mathrm{T}}x,
\end{aligned}
$$

which in turn shows that, for any $\gamma \in (0, \gamma^*]$, the equilibrium $x = 0$ of the closed-loop system is asymptotically stable with $x(0) \in \mathcal{W} \subset L_V(c)$ in the domain of attraction. Since $x(0)$ is arbitrary, we have global stabilization. Furthermore, since $\gamma_j < \gamma_{j-1}$, the algorithm is able to find a suitable $\gamma \in (0, \gamma^*]$. For such a γ, the convergence of $P_k(\gamma)$ to $P^*(\gamma)$ follows from the convergence of value iteration algorithm [14] and the unique solution of the learning equation (2.45) if the rank condition (4.27) holds. This completes the proof. □

We now work towards developing the output feedback version of Algorithm 2. For this purpose, we employ the continuous-time state parametrization result presented in Chapter 2.

Theorem 4.6. *There exists a state parametrization*

$$
\bar{x}(t) = M_u\zeta_u(t) + M_y\zeta_y(t) \tag{4.31}
$$

that converges exponentially to the state x if the system is observable, where M_u and M_y are the system dependent matrices containing the transfer function coefficients, and ζ_u and ζ_y are the user-defined dynamics given by

$$
\dot{\zeta}_u^i(t) = \mathcal{A}\zeta_u^i(t) + \mathcal{B}\sigma(u_i(t)),\ \zeta_u^i(0) = 0, i = 1, 2, \cdots, m \tag{4.32}
$$

$$
\dot{\zeta}_y^i(t) = \mathcal{A}\zeta_y^i(t) + \mathcal{B}y_i(t),\ \zeta_y^i(0) = 0, i = 1, 2, \cdots, p, \tag{4.33}
$$

for some Hurwitz matrix $\mathcal{A}$ and input vector $\mathcal{B}$ of the form,

$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -\alpha_0 & -\alpha_1 & \cdots & \cdots & -\alpha_{n-1} \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

Proof: The proof remains the same as that of Theorem 2.5 in Chapter 2 by treating the constrained input $\sigma(u)$ as the input vector. This completes the proof. □

We will now employ the result in Theorem 4.6 to develop the output feedback counterpart of the learning equation (4.25).

To this end, we introduce the following definitions. Let

$$z(t) = \begin{bmatrix} \zeta_u^{\mathrm{T}}(t) & \zeta_y^{\mathrm{T}}(t) \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{(mn+pn)}, \tag{4.34}$$

$$M = \begin{bmatrix} M_u & M_y \end{bmatrix} \in \mathbb{R}^{n\times(mn+pn)}, \tag{4.35}$$

$$\bar{P}(\gamma) = M^{\mathrm{T}} P(\gamma) M \in \mathbb{R}^{(mn+pn)\times(mn+pn)}, \tag{4.36}$$

$$\bar{K}(\gamma) = K(\gamma) M \in \mathbb{R}^{m\times(mn+pn)}. \tag{4.37}$$

With these definitions, we revisit Equation (4.25). Adding $\gamma \int_{t-T}^{t} y^{\mathrm{T}}(\tau) y(\tau) d\tau$ on both sides of this equation, and substituting $y(t) = Cx(t)$ and $H_k = A^{\mathrm{T}}P + PA$ on the right-hand side, we have,

$$\begin{aligned} & x^{\mathrm{T}}(t)P(\gamma)x(t) - x^{\mathrm{T}}(t-T)P(\gamma)x(t-T) + \gamma \int_{t-T}^{t} y^{\mathrm{T}}(\tau)y(\tau)d\tau \\ =& \int_{t-T}^{t} x^{\mathrm{T}}(\tau)(A^{\mathrm{T}}P(\gamma) + P(\gamma)A + \gamma C^{\mathrm{T}}C)x(\tau)d\tau \\ & -2\int_{t-T}^{t} (\sigma^{\mathrm{T}}(u(\tau)))K(\gamma)x(\tau)d\tau. \end{aligned}$$

Next, we use our state parametrization (4.31) and the definitions (4.34)-(4.37) in the above equation, which results in,

$$\begin{aligned} & z^{\mathrm{T}}(t)\bar{P}(\gamma)z(t) - z^{\mathrm{T}}(t-T)\bar{P}(\gamma)z(t-T) + \gamma \int_{t-T}^{t} y^{\mathrm{T}}(\tau)y(\tau)d\tau \\ =& \int_{t-T}^{t} z^{\mathrm{T}}(\tau)\bar{H}(\gamma)z(\tau)d\tau - 2\int_{t-T}^{t} \sigma^{\mathrm{T}}(u(\tau))\bar{K}(\gamma)z(\tau)d\tau, \end{aligned} \tag{4.38}$$

or equivalently,

$$z^{\mathrm{T}}(t)\otimes z^{\mathrm{T}}(t)|_{t-T}^{t}\ \mathrm{vec}(\bar{P}(\gamma))+\gamma\int_{t-T}^{t}y^{\mathrm{T}}\otimes y^{\mathrm{T}}d\tau\ \mathrm{vec}(I)$$
$$=\int_{t-T}^{t}\bar{z}d\tau\ \mathrm{vecs}(\bar{H}(\gamma))-2\int_{t-T}^{t}z^{\mathrm{T}}(\tau)\otimes\sigma^{\mathrm{T}}(u(\tau))d\tau\ \mathrm{vec}(\bar{K}(\gamma)),$$

where $\bar{H}(\gamma)=M^{\mathrm{T}}(A^{\mathrm{T}}P(\gamma)+P(\gamma)A+\gamma C^{\mathrm{T}}C)M$ and

$\bar{z}=\{z_1^2,2z_1z_2,2z_1z_3,\cdots,z_2^2,2z_2z_3,\cdots,z_{nm+np}^2\}$. Equation (4.38) is the low gain parameterized learning equation that uses only output feedback. Similar to our state feedback equation (4.25), it is a scalar equation which is linear in terms of the unknowns $\bar{H}(\gamma)$ and $\bar{K}(\gamma)$. These matrices are the output feedback counterparts of the matrices $H(\gamma)$ and $K(\gamma)$. As there are more unknowns than the number of equations, we develop a system of l number of such equations by performing l finite window integrals each of length T. To solve this linear system of equations, we define the following data matrices,

$$\delta_{zz}=\Big[z\otimes z|_{t_0}^{t_1},z\otimes z|_{t_1}^{t_2},\cdots,z\otimes z|_{t_{l-1}}^{t_l}\Big]^{\mathrm{T}},$$
$$I_{zu}=\Big[\textstyle\int_{t_0}^{t_1}z\otimes\sigma(u)d\tau,\int_{t_1}^{t_2}z\otimes\sigma(u)d\tau,\cdots,\int_{t_{l-1}}^{t_l}z\otimes\sigma(u)d\tau\Big]^{\mathrm{T}},$$
$$I_{zz}=\Big[\textstyle\int_{t_0}^{t_1}\bar{z}^{\mathrm{T}}d\tau,\int_{t_1}^{t_2}\bar{z}^{\mathrm{T}}d\tau,\cdots,\int_{t_{l-1}}^{t_l}\bar{z}^{\mathrm{T}}d\tau\Big]^{\mathrm{T}},$$
$$I_{yy}=\Big[\textstyle\int_{t_0}^{t_1}y\otimes yd\tau,\int_{t_1}^{t_2}y\otimes yd\tau,\cdots,\int_{t_{l-1}}^{t_l}y\otimes yd\tau\Big]^{\mathrm{T}}.$$

We rewrite (4.38) as the following matrix equation,

$$\Big[I_{zz},2I_{zu}\Big]\begin{bmatrix}\mathrm{vecs}(\bar{H}(\gamma))\\ \mathrm{vec}(\bar{K}(\gamma))\end{bmatrix}=\delta_{zz}\mathrm{vec}(\bar{P}(\gamma))+\gamma I_{yy}\mathrm{vec}(I),$$

whose least-squares solution is given by,

$$\begin{bmatrix}\mathrm{vecs}(\bar{H}(\gamma))\\ \mathrm{vec}(\bar{K}(\gamma))\end{bmatrix}=\Big(\Big[I_{zz},\ 2I_{zu}\Big]^{\mathrm{T}}\Big[I_{zz},\ 2I_{zu}\Big]\Big)^{-1}\Big[I_{zz},\ 2I_{zu}\Big]^{\mathrm{T}}$$
$$\times(\delta_{xx}\mathrm{vec}(P(\gamma))+\gamma I_{yy}\mathrm{vec}(I)) \tag{4.39}$$

Remark 4.3. *Compared to its state feedback counterpart, Equation* (4.39) *solves the least-squares problem using input-output data instead of input-state data. The number of unknown parameters corresponding to* $\bar{H}(\gamma)$ *and* $\bar{K}(\gamma)$ *is* $(mn+pn)(mn+pn+1)/2$ *and* $m(mn+pn)$*, respectively, which are larger than that for the state feedback problem* (4.26)*. As a result, we require* $l\geq(mn+pn)(mn+pn+1)/2+m(mn+pn)$

learning intervals and datasets. These extra control parameters relax the requirement of full-state measurement. Similar to the state feedback problem, we need to inject an exploration signal so that the following rank condition is satisfied,

$$\textit{rank}\left(\begin{bmatrix} I_{zz}, & 2I_{zu} \end{bmatrix}\right) = (mn+pn)(mn+pn+1)/2 + m(mn+pn). \tag{4.40}$$

We are now ready to propose an output feedback low gain scheduled learning algorithm that achieves model-free global stabilization of the system in the presence of actuator saturation.

Algorithm 4: Iterative Output Feedback IRL Scheme Under Constrained Control

1. Initialization. Start with an arbitrary open-loop policy $u^0 = \nu$ with ν being the exploration signal and u^0 satisfying the control constraint $\|u\|_\infty \leq b$, where $\|\cdot\|_\infty$ denotes the ∞ norm of a vector. Set $\gamma \leftarrow 1$, $k \leftarrow 0$, $q \leftarrow 0$, and $\bar{P}_0(\gamma) \geq 0$.

2. Collect Online Data. Apply the policy u^0 for the duration $t \in [t_0, t_l]$, composed of l learning intervals of length $t_j - t_{j-1} = T$, where $t_l = t_0 + lT$ and $l \geq (mn+pn)(mn+pn+1)/2 + m(mn+pn)$. Collect the data for $y, z, \sigma(u)$ and their quadratic integrals for each learning interval. Then, for $k = 0, 1, \cdots$, perform,

3. loop:

Determine the least-squares solution, $\bar{H}_k(\gamma)$ and $\bar{K}_k(\gamma)$, of the learning equation,

$$\begin{aligned} &z^{\mathrm{T}}(t)\bar{P}_k(\gamma)z(t) - z^{\mathrm{T}}(t-T)\bar{P}_k(\gamma)z(t-T) + \gamma \int_{t-T}^{t} y^{\mathrm{T}}(\tau)y(\tau)d\tau \\ =&\int_{t-T}^{t} z^{\mathrm{T}}(\tau)\bar{H}_k(\gamma)z(\tau)d\tau - 2\int_{t-T}^{t} \sigma^{\mathrm{T}}(u(\tau))\bar{K}_k(\gamma)z(\tau)d\tau, \end{aligned}$$

Update $\tilde{\bar{P}}(\gamma)$ as,

$$\tilde{\bar{P}}_{k+1}(\gamma) \leftarrow \bar{P}_k(\gamma) + \epsilon_k(\bar{H}_k(\gamma) - \bar{K}_k^{\mathrm{T}}(\gamma)\bar{K}_k(\gamma)).$$

if $\tilde{\bar{P}}_{k+1}(\gamma) \notin B_q$ **then**

$$\bar{P}_{k+1}(\gamma) \leftarrow \bar{P}_0(\gamma) \text{ and } q \leftarrow q+1.$$

else if $\left\|\tilde{\bar{P}}_{k+1}(\gamma)-\bar{P}_k(\gamma)\right\|/\epsilon_k<\varepsilon$ **then**

return $\bar{P}_k(\gamma)$ as an estimate of $\bar{P}^*(\gamma)$.

else

$$\bar{P}_{k+1}(\gamma) \leftarrow \tilde{\bar{P}}_{k+1}(\gamma).$$

end if

$$k \leftarrow k+1.$$

end loop.

The resulting $\hat{\bar{K}}(\gamma)$ is an approximation of $\bar{K}^*(\gamma)$ for a given γ.

4. Control Saturation Check. For all $t \geq t_f$, check the following saturation condition,

$$\|\hat{\bar{K}}(\gamma)z\|_\infty \leq b.$$

If for any $t \geq t_f$ the saturation condition is violated, reduce γ, reset $i = 1$, and repeat Step 3 with the reduced γ.

5. Terminate.

The convergence of the Algorithm 3 is shown in Theorem 4.7.

Theorem 4.7. *Under Assumption 4.2 and 4.3 and the rank condition* (4.40)*, the proposed output feedback based Algorithm 4 globally stabilizes the system without saturating the actuators.*

Proof: According to Theorem 4.6, the state $x(t)$ can be reconstructed in terms of the filtered inputs and outputs as given by (4.31) if the pair (A, C) is observable. Thus, the output feedback equation (4.38) becomes exponentially equivalent to the state feedback IRL equation (4.25). The output feedback IRL equation (4.38) is uniquely solvable if the rank condition (4.40) is satisfied. Then, following the

arguments in Theorem 4.5, the output feedback Algorithm 4 also achieves global stabilization of the system by finding a suitable low gain parameter γ and the corresponding output feedback control matrix $\bar{K}(\gamma)$ without saturating the actuators. This completes the proof. □

4.4. Results

In this section we test the proposed scheme using numerical simulation.

Example 1: A Discrete-time Polynomially Unstable System:

Consider the discrete-time system (4.20) with

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 2\sqrt{2} & -4 & 2\sqrt{2} \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}.$$

Matrix A has a pair of repeated eigenvalues at $\frac{\sqrt{2}}{2} \pm j\frac{\sqrt{2}}{2}$, which lie on the unit circle, and therefore, the system is open-loop unstable. The actuator saturation limit is $b = 1$. The initial state of the system is $x_0 = [1\ 1\ 1\ 1]^{\mathrm{T}}$. The algorithm is initialized with $\varepsilon = 1$ and $K^0(\varepsilon) = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$. Fig. 4.1 shows the state response and the control effort. The low gain parameter ε is reduced by a factor of one-half and the convergence criterion of $\delta = 0.01$ was selected. It can be seen that the algorithm is able to find a suitable low gain parameter ε and the corresponding low gain state feedback control matrix $K(\varepsilon)$ that guarantees convergence of the state without saturating the actuators. The final value of $\varepsilon = 2.4 \times 10^{-4}$. The corresponding nominal low gain feedback matrix is

$$K^*(\varepsilon) = \begin{bmatrix} -0.3002 & 0.6717 & -0.6801 & 0.2537 \end{bmatrix}.$$

and its final estimate is found to be

$$\hat{K}(\varepsilon) = \begin{bmatrix} -0.3025 & 0.6771 & -0.6856 & 0.2558 \end{bmatrix},$$

which shows convergence to the solution of the ARE. We simulate again Algorithm 1 with a different initial condition, i.e., $x_0 = [5\ 5\ 5\ 5]^{\mathrm{T}}$. All other simulation parameters are kept the same. Fig. 4.2 shows the closed-loop response. The final value of ε is found to be 6.1×10^{-5} and the corresponding nominal

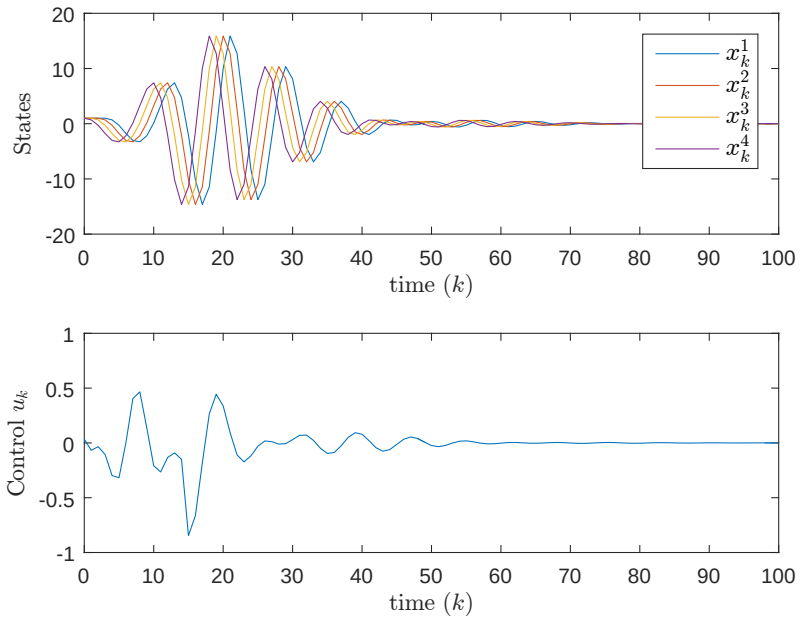

Fig. 4.1: Example 1: Closed-loop response using state feedback Algorithm 1 with $x_0 = [1\ 1\ 1\ 1]^{\mathrm{T}}$

low gain matrix is

$$K^*(\varepsilon) = \begin{bmatrix} -0.2222 & 0.4899 & -0.4860 & 0.1781 \end{bmatrix}$$

and its corresponding estimate is found to be

$$\hat{K}(\varepsilon) = \begin{bmatrix} -0.2254 & 0.4970 & -0.4930 & 0.1807 \end{bmatrix}.$$

Following the state feedback case, we now validate Algorithm 2 which uses output feedback. We choose $N = 4$ as the bound on the observability index. The initial state of the system is $x_0 = [1\ 1\ 1\ 1]^{\mathrm{T}}$. The algorithm is initialized with $\varepsilon = 1$, $\mathcal{H}^0(\varepsilon) = I$, $\mathcal{K}^0(\varepsilon) = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$. Fig. 4.3 shows the state response and the control effort. The low gain parameter ε is reduced by a factor of one-half and the convergence criterion of $\delta = 0.01$ was selected. In this simulation, we collected $L = 50$ data samples to satisfy the rank condition (4.9). It can be seen that the algorithm is able to find a suitable low gain parameter ε and the corresponding low gain state feedback control matrix $\mathcal{K}(\varepsilon)$ that guarantees convergence of the state without saturating the actuators. The final value of $\varepsilon = 1.2 \times 10^{-4}$. The nominal

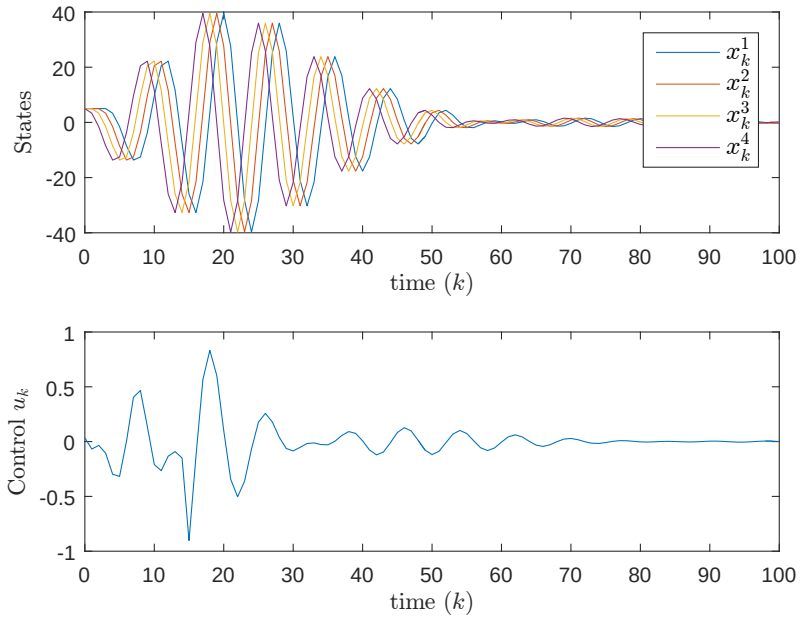

Fig. 4.2: Example 1: Closed-loop response using state feedback Algorithm 1 with $x_0 = [5\ 5\ 5\ 5]^{\mathrm{T}}$

output feedback low gain feedback matrix is

$$\mathcal{K}^*(\varepsilon) = \begin{bmatrix} 0.1495 & 0.0123 & -0.1460 & -0.2298 & -0.1804 & 0.4937 & -0.5038 & 0.2298 \end{bmatrix}.$$

and its final estimate is found to be

$$\hat{\mathcal{K}}(\varepsilon) = \begin{bmatrix} 0.1515 & 0.0126 & -0.1477 & -0.2325 & -0.1826 & 0.4996 & -0.5099 & 0.2325 \end{bmatrix},$$

which shows convergence to the solution of the ARE.

We next simulate again Algorithm 2 with a different initial condition, i.e., $x_0 = [5\ 5\ 5\ 5]^{\mathrm{T}}$. All other simulation parameters are kept the same. Fig. 4.4 shows the closed-loop response. The final value of ε is found to be 1.5×10^{-5} and the corresponding nominal output matrix is

$$\mathcal{K}^*(\varepsilon) = \begin{bmatrix} 0.0886 & 0.0042 & -0.0878 & -0.1323 & -0.0996 & 0.2765 & -0.2863 & 0.1323 \end{bmatrix}.$$

and its final estimate is

$$\hat{\mathcal{K}}(\varepsilon) = \begin{bmatrix} 0.0904 & 0.0042 & -0.0897 & -0.1351 & -0.1016 & 0.2822 & -0.2923 & 0.1351 \end{bmatrix}.$$

It can be seen that even with this larger initial condition, the algorithm was able to stabilize the system

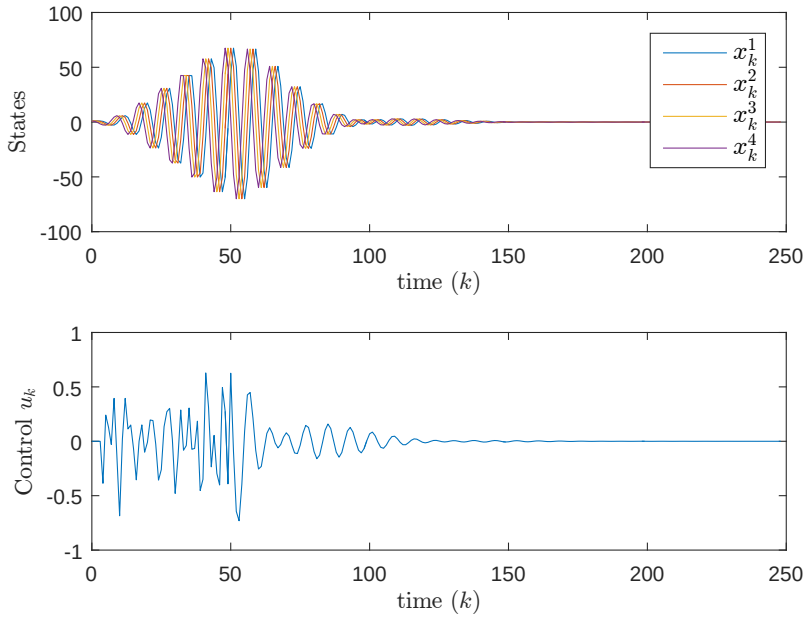

Fig. 4.3: Example 1: Closed-loop response using output feedback Algorithm 2 with $x_0 = [1\ 1\ 1\ 1]^{\mathrm{T}}$

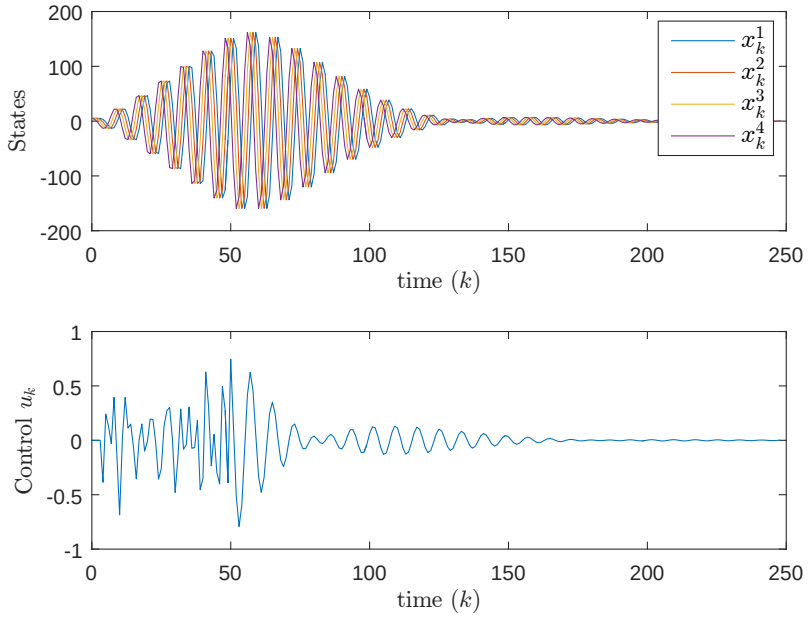

Fig. 4.4: Example 1: Closed-loop response using output feedback Algorithm 2 with $x_0 = [5\ 5\ 5\ 5]^{\mathrm{T}}$

with a lower value of the low gain parameter.

Example 2: A Continuous-time Polynomially Unstable System Under Actuator Constraints:

Consider the following continuous-time system subject to actuator constraints,

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & -2 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

Matrix A has a pair of repeated eigenvalues at $\pm j$, and therefore, the system is open-loop unstable. The actuator saturation limit is $b=1$. The initial state of the system is $x(0) = \begin{bmatrix} 0.25 & -0.5 & -0.5 & 0.25 \end{bmatrix}^{\mathrm{T}}$ and the algorithm is initialized with $\gamma = 1$ and $P_0(\gamma) = 0.01 I_4$. Step size $\epsilon_k = (k^{0.5} + 5)^{-1}, k = 0, 1, 2, \ldots$ and the set $B_q = \left\{ P \in \mathcal{P}_+^4 : |P| \leq 200(q+1) \right\}, q = 0, 1, 2, \ldots$ Fig. 4.5 shows the state response and the control effort. In every main iteration of the algorithm, the low gain parameter γ is reduced by a factor of one half. Since $n = 4$ and $m = 1$, we need at least $n(n+1)/2 + mn = 14$ learning intervals to collect data for solving Equation (4.25). In these simulations, we use sinusoids of varying frequencies and magnitudes while ensuring that the control constraint is satisfied. This trajectory data can be used in subsequent iterations of the low gain parameter γ. In this simulation, we chose $l = 30$ learning intervals of period $T = 0.2$ sec to satisfy the above mentioned condition. It can be seen in Fig. 4.5 that the algorithm finds a suitable low gain parameter γ and learns the corresponding low gain feedback matrix $K(\gamma)$ using the IRL equation to guarantee convergence of the state. The final value of γ as obtained by the scheduling mechanism is 0.1250. The corresponding nominal value of the low gain control matrix is

$$K^*(\gamma) = \begin{bmatrix} 0.0607 & 1.4046 & 0.7567 & 1.2800 \end{bmatrix}.$$

and the corresponding estimate of the low gain matrix is

$$\hat{K}(\gamma) = \begin{bmatrix} 0.0600 & 1.4006 & 0.7550 & 1.2773 \end{bmatrix},$$

which shows convergence to the solution of the ARE. It should be noted that the exploration signal is only needed in the first $l = 30$ intervals and can be removed afterwards. We next verify the global stabilization property of Algorithm 3. To this end, we simulate the system with a different initial condition $x(0) = \begin{bmatrix} 0.5 & -1 & -1 & 0.5 \end{bmatrix}^{\mathrm{T}}$. All other simulation parameters are the same. In this case, the value of the low gain parameter can be expected to be lower since our initial condition belongs to a larger basin of attraction. Indeed, this is the case and the final value of low gain parameter as found from the scheduling

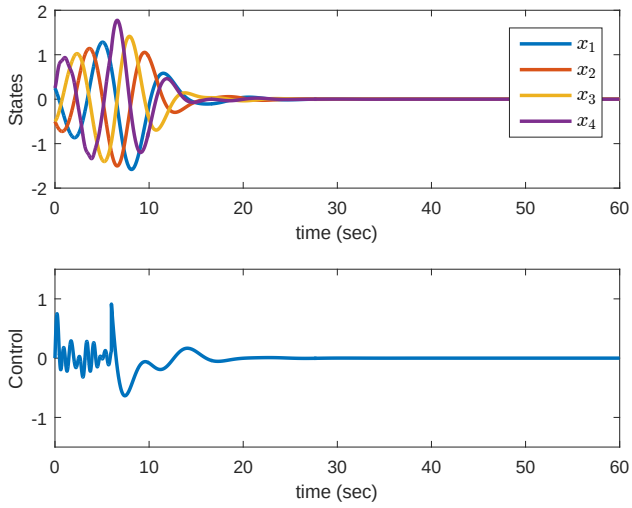

Fig. 4.5: Example 2: Closed-loop response using state feedback Algorithm 3 with $x(0) = [0.25 \ -0.5 \ -0.5 \ \ 0.25]^{\mathrm{T}}$.

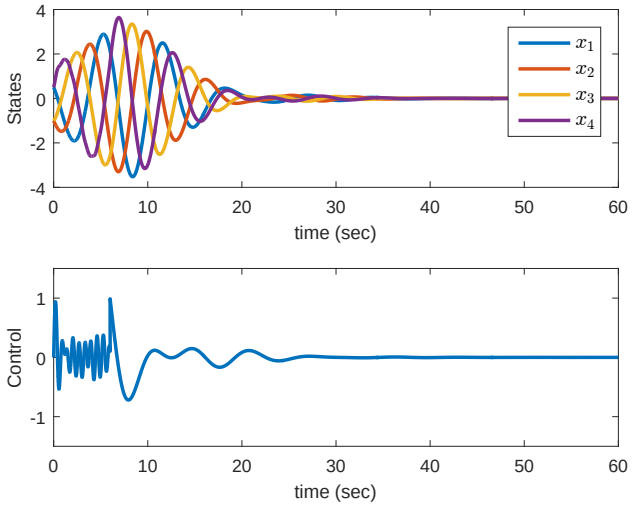

Fig. 4.6: Example 2: Closed-loop response using state feedback Algorithm 3 with $x(0) = [0.5 \ -1 \ -1 \ \ 0.5]^{\mathrm{T}}$.

mechanism is 0.0156. The nominal low gain matrix is

$$K^*(\gamma) = \begin{bmatrix} 0.0078 & 0.7510 & 0.2567 & 0.7273 \end{bmatrix}.$$

and the corresponding estimate of the low gain matrix is

$$\hat{K}(\gamma) = \begin{bmatrix} 0.0095 & 0.7463 & 0.2570 & 0.7224 \end{bmatrix}.$$

Fig. 4.6 shows the closed-loop response. It can be seen that, with this larger initial condition, the algorithm stabilizes the system with a lower value of the low gain parameter. This, in turn, verifies that the basin of attraction can made arbitrary large by scheduling the low gain parameter. In both the cases, we find that the control does not violate the saturation condition during both the learning and stabilization phases, which is an advantage of the proposed scheme.

The results discussed so far achieve global stabilization under actuator constraints based on full-state feedback. We now present the results of the proposed output feedback Algorithm 4 that uplifts the requirement of the full-state feedback. For the state parametrization, we construct the user-defined system matrix $\mathcal{A}$ with the our choice of desired eigenvalues at -1. This corresponds to $\alpha_0 = 1$, $\alpha_1 = 4$, $\alpha_2 = 6$, $\alpha_3 = 4$, which are the entries of $\mathcal{A}$. These constants are obtained from the characteristic polynomial $\Lambda(s) = (s+1)^4$ corresponding to our choice of eigenvalues -1 of the matrix $\mathcal{A}$. The initial state of the system is $x(0) = \begin{bmatrix} 0.25 & -0.5 & -0.5 & 0.25 \end{bmatrix}^{\mathrm{T}}$ and the algorithm is initialized with $\gamma = 1$ and $\bar{P}_0(\gamma) = 0.01 I_8$. Step size $\epsilon_k = (k^{0.5}+5)^{-1}, k = 0, 1, 2, \ldots$ and the set $B_q = \left\{ \bar{P} \in \mathcal{P}_+^8 : |\bar{P}| \leq 800(q+1) \right\}, q = 0, 1, 2, \ldots$ Fig. 4.7 shows the state response and the control effort. In every main iteration of the algorithm, the low gain parameter γ is reduced by a factor of one half. Since $n = 4$, $m = 1$ and $p = 1$, we need at least $(mn + pn)(mn + pn + 1)/2 + m(mn + pn) = 44$ learning intervals to solve (4.38). In these simulations, we use sinusoids of varying frequencies and magnitudes while ensuring that the control constraint is satisfied. This trajectory data can be used in subsequent iterations of the low gain parameter γ. In this simulation, we chose $l = 60$ learning intervals of period $T = 0.2$ sec to satisfy the above mentioned condition. It can be seen in Fig. 4.7. The nominal output feedback low gain control matrix corresponding to $\gamma = 0.1250$ is computed using (4.37) as

$$\mathcal{K}^*(\gamma) = [\, 2.2071 \quad 6.0464 \quad 3.8707 \quad 0.8715 \quad -2.1460 \quad -4.7727 \quad -3.4243 \quad -4.1577 \,],$$

and its estimate is

$$\hat{\mathcal{K}}(\gamma) = [\, 2.2034 \quad 6.0429 \quad 3.8691 \quad 0.8711 \quad -2.1439 \quad -4.7732 \quad -3.4223 \quad -4.1588 \,],$$

which shows convergence to the solution of the ARE. It should be noted that the exploration signal is only needed in the first $l = 60$ intervals and can be removed afterwards. We next verify the global stabilization property of Algorithm 4. To this end, we simulate the system with a different initial condition $x(0) = \begin{bmatrix} 0.5 & -1 & -1 & 0.5 \end{bmatrix}^{\mathrm{T}}$. All other simulation parameters are the same. In this case, the value of

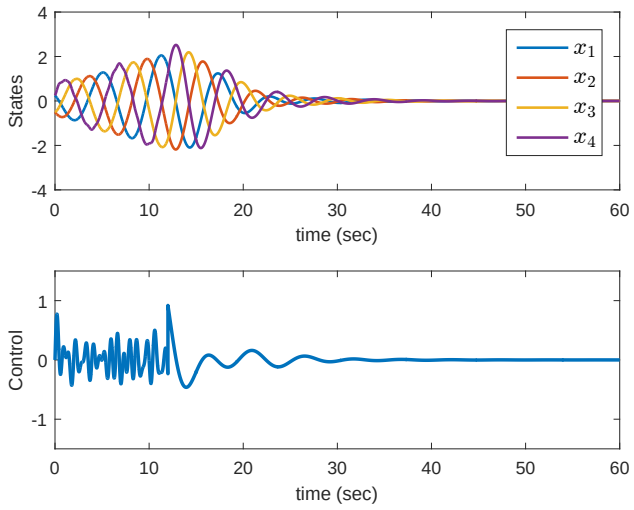

Fig. 4.7: Example 2: Closed-loop response using output feedback Algorithm 4 with $x(0) = [0.25 \ -0.5 \ -0.5 \ \ 0.25]^{\mathrm{T}}$.

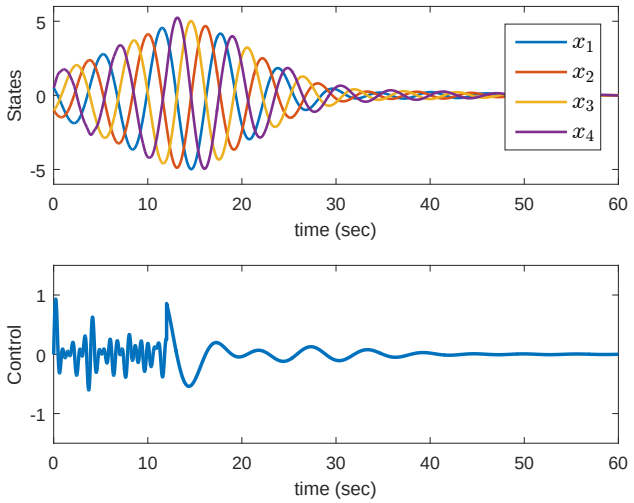

Fig. 4.8: Example 2: Closed-loop response using output feedback Algorithm 4 with $x(0) = [0.5 \ -1 \ -1 \ \ 0.5]^{\mathrm{T}}$.

the low gain parameter can be expected to be lower since our initial condition belongs to larger basin of attraction. Indeed, this is the case and the final value of the low gain parameter is found to be $\gamma = 0.0156$. The corresponding nominal output feedback low gain control matrix is

$$\mathcal{K}^*(\gamma) = [\, 0.6579 \quad 3.0932 \quad 2.1612 \quad 0.5074 \quad -0.6500 \quad -2.5196 \quad -1.1348 \quad -2.3912 \,],$$

and its estimate is

$$\hat{\mathcal{K}}(\gamma) = [\, 0.6540 \quad 3.1001 \quad 2.1678 \quad 0.5091 \quad -0.6468 \quad -2.5277 \quad -1.1326 \quad -2.4016 \,].$$

Fig. 4.8 shows the closed-loop response. It can be seen that, with this larger initial condition, the algorithm stabilizes the system with a lower value of the low gain parameter. This, in turn, verifies that the basin of attraction can made arbitrary large by scheduling the low gain parameter.

Upon comparing the results of Algorithms 3 and 4, we find that both algorithms are able to achieve global stabilization without saturating the actuators and without requiring the knowledge of system dynamics. Furthermore, Algorithm 4 relaxes the requirement of full-state feedback at the cost of estimating more parameters, and as a result, its learning phase lasts longer than that of Algorithm 3.

4.5. Summary

In this chapter, we have presented model-free algorithms for global stabilization of linear systems subject to actuator saturation. The idea of gain-scheduled low gain feedback is applied to develop control laws that avoid saturation and achieve global stabilization. To design these control laws, we employ the framework of parameterized algebraic Riccati equations (AREs). We present an iterative Q-learning algorithm that searches for a low gain parameter and iteratively solves the parameterized ARE using the Bellman equation. Both state feedback and output feedback algorithms are developed. It is shown that the proposed algorithms achieves model-free global stabilization under bounded controls and converges to the solution of the ARE. The low gain feedback design approach is also extended to solve the continuous-time constrained control problems. Both state feedback and output feedback IRL algorithms are developed. Compared to the previous works, the proposed scheme has the advantage that the resulting control laws have a linear structure and global asymptotic stability is ensured without causing actuator saturation. Simulation results have been presented that confirm the effectiveness of the proposed method.

5. MODEL-FREE OPTIMAL STABILIZATION OF TIME DELAY SYSTEMS

5.1. Introduction

A time delay system is a dynamical system whose evolution depends on its past inputs and states. Many real world problems in sciences and engineering can be described as time delay problems. Very often delays arise in control implementation such as those associated with the sensors, actuators, communication and computation. From the controls perspective, the presence of time delays makes the control problem quite challenging because of the performance degradation and instabilities that may result from these delays. As a result, optimal stabilization of time delay systems remains a practical control issue. To this date, significant progress has been demonstrated in the control literature on the model-based control approaches to addressing time delay control problems [22]. However, the requirement of perfectly known models is not always realistic in many applications owing to the complexity of the systems themselves and the ever presence of model uncertainties. As a result, model-free optimal control techniques are more desirable in these scenarios.

In this chapter, we will present model-free control schemes for time delay systems based on reinforcement learning. The discussion throughout this chapter is restricted to discrete-time delay systems as we exploit the finite dimensional property of these systems to bring them into a delay-free form. We present an extended state augmentation approach that treats the delayed state and input variables as additional states in order to obtain a delay-free form. The method of Q-learning will be applied to design both state feedback and output feedback controllers.

5.2. Extended State Augmentation

Consider a discrete-time linear system with both state and input delays given by the following state space representation,

$$\begin{aligned} x_{k+1} &= \textstyle\sum_{i=0}^{S} A_i x_{k-i} + \sum_{i=0}^{T} B_i u_{k-i}, \\ y_k &= C x_k, \end{aligned} \tag{5.1}$$

where $x_k \in \mathbb{R}^n$ represents the internal state, $u_k \in \mathbb{R}^m$ is the control input vector and $y_k \in \mathbb{R}^p$ is the output vector. The system is subject to multiple delays in both the state and inputs with S and T being the maximum amount of state and input delays, respectively. The model of the system as well as the length and the number of delays are assumed unknown in our problem.

We are interested in solving the linear quadratic regulation (LQR) problem for the time delay system (5.1) with the following cost function,

$$J = \sum_{i=0}^{\infty} r(x_i, u_i), \tag{5.2}$$

where $r(x_k, u_k)$ takes the following quadratic form,

$$r_k = x_k^{\mathrm{T}} Q x_k + u_k^{\mathrm{T}} R u_k, \tag{5.3}$$

and where $Q = Q^{\mathrm{T}} \geq 0$ and $R = R^{\mathrm{T}} > 0$ correspond to the desired parameters for penalizing the states and control, respectively.

Due to the presence of time delays, the optimal control techniques that we developed in the earlier chapters are not directly applicable. In the following, we will present a state augmentation procedure that brings the system into a delay-free form. The augmentation is carried out by introducing the delayed states and control inputs as additional states. To this end, let us define the augmented state vector as,

$$X_k = \begin{bmatrix} x_k^{\mathrm{T}} & x_{k-1}^{\mathrm{T}} & \cdots & x_{k-S}^{\mathrm{T}} & u_{k-T}^{\mathrm{T}} & u_{k-T+1}^{\mathrm{T}} & \cdots & u_{k-1}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}. \tag{5.4}$$

The dynamic equation of the augmented system can be obtained from the original dynamics (5.1) as follows,

$$\begin{aligned} X_{k+1} &= \left[\begin{array}{ccccc|ccccc} A_0 & A_1 & A_2 & \cdots & A_S & B_T & B_{T-1} & B_{T-2} & \cdots & B_1 \\ I_n & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & I_n & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & I_n & 0 & 0 & \cdots & 0 & 0 & 0 \\ \hline 0 & \cdots & 0 & 0 & 0 & 0 & I_m & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & I_m & \vdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & I_m \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \end{array}\right] X_k + \begin{bmatrix} B_0 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ I_m \end{bmatrix} u_k \\ &\triangleq AX_k + Bu_k. \end{aligned} \tag{5.5}$$

Since the maximum state delay S and input delay T are not known, we will extend the augmentation

further up to their upper bounds $\bar{S}$ and $\bar{T}$, respectively. For this purpose, we introduce the extended augmented state vector as,

$$\begin{aligned}\bar{X}_k &= [x_k^{\mathrm{T}} \quad x_{k-1}^{\mathrm{T}} \quad \cdots \quad x_{k-S}^{\mathrm{T}} \quad x_{k-S-1}^{\mathrm{T}} \quad \cdots \quad x_{k-\bar{S}}^{\mathrm{T}} \\ &\qquad u_{k-\bar{T}}^{\mathrm{T}} \quad u_{k-\bar{T}+1}^{\mathrm{T}} \quad \cdots \quad u_{k-T}^{\mathrm{T}} \quad \cdots \quad u_{k-1}^{\mathrm{T}}]^{\mathrm{T}}, \end{aligned} \tag{5.6}$$

and the corresponding extended augmented dynamics is given by,

$$\begin{aligned}\bar{X}_{k+1} = &\left[\begin{array}{ccccc|ccccc} \overbrace{A_0 & \cdots & A_S & 0 & \cdots}^{\bar{S}+1 \text{ blocks}} & \overbrace{0 & \cdots & B_T & \cdots & B_1}^{\bar{T} \text{ blocks}} \\ I_n & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & I_n & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & \cdots & 0 & I_n & 0 & 0 & \cdots & 0 & 0 & 0 \\ \hline 0 & \cdots & 0 & 0 & 0 & 0 & I_m & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & I_m & \vdots & 0 \\ \vdots & \cdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & I_m \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \end{array}\right] \bar{X}_k + \left[\begin{array}{c} B_0 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ I_m \end{array}\right] u_k \\ &\triangleq \bar{A}\bar{X}_k + \bar{B}u_k. \end{aligned} \tag{5.7}$$

We will now study the controllability property of the augmented systems (5.5) and (5.7).

Theorem 5.1. *The delay-free augmented systems* (5.5) *and* (5.7) *are controllable if and only if*

$$\rho\left[\sum_{i=0}^{S} A_i\lambda^{S-i} - \lambda^{S+1}I \quad \sum_{i=0}^{T} B_i\lambda^{T-i}\right] = n, \tag{5.8}$$

for any $\lambda \in \left\{\lambda \in \mathbb{C} : det\left(\sum_{i=0}^{S} A_i\lambda^{S-i} - \lambda^{S+1}I\right) = 0\right\}$.

Proof: Consider the augmented system (5.5). From linear systems theory, the system is controllable if and only if $\left[A - \lambda I \quad B\right]$ has a full row rank for all $\lambda \in \mathbb{C}$. We evaluate $\rho\left[A - \lambda I \quad B\right]$ as,

$$\rho\begin{bmatrix} A-\lambda I & B \end{bmatrix} = \rho \left[\begin{array}{ccccc|ccccc} A_0-\lambda I & A_1 & A_2 & \cdots & A_S & B_T & B_{T-1} & B_{T-2} & \cdots & B_1 & B_0 \\ I_n & -\lambda I & 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 \\ 0 & I_n & -\lambda I & \vdots & \vdots & 0 & \cdots & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & I_n & -\lambda I & 0 & \cdots & 0 & \cdots & 0 & 0 \\ \hline 0 & \cdots & 0 & 0 & 0 & -\lambda I & I_m & 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & -\lambda I & I_m & \vdots & 0 & 0 \\ \vdots & \cdots & \vdots & \vdots & \vdots & \vdots & \cdots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & -\lambda I & I_m & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & -\lambda I & I_m \end{array}\right].$$

Performing some rows and column operations to cancel out the λI entries in the columns of $B's$, we have

$$\rho\begin{bmatrix} A-\lambda I & B \end{bmatrix} = \rho \left[\begin{array}{c|ccccc} * & B_T+B_{T-1}\lambda+\cdots+B_0\lambda^T & 0 & 0 & \cdots & 0 \\ * & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ * & 0 & 0 & 0 & \cdots & 0 \\ \hline 0 & 0 & I_m & 0 & \cdots & 0 \\ 0 & 0 & 0 & I_m & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & I_m \end{array}\right].$$

Applying similar row and column operations to cancel out the λI entries corresponding to the columns

of A_i's results in,

$$\rho\begin{bmatrix} A-\lambda I & B \end{bmatrix} = \rho\left[\begin{array}{cccc|c|ccc} 0 & 0 & \cdots & \sum_{i=0}^{S} A_i\lambda^{S-i}-\lambda^{S+1}I & \sum_{i=0}^{T} B_i\lambda^{T-i} & 0 & 0 & \cdots & 0 \\ I_n & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & I_n & 0 & 0 & 0 & 0 & \cdots & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & I_m & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I_m & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & I_m \end{array}\right]$$

$$= \rho\begin{bmatrix} \sum_{i=0}^{S} A_i\lambda^{S-i}-\lambda^{S+1}I & \sum_{i=0}^{T} B_i\lambda^{T-i} \end{bmatrix} + nS + mT.$$

At full row rank, $\rho\begin{bmatrix} A-\lambda I & B \end{bmatrix} = n + nS + mT$, which requires

$$\rho\begin{bmatrix} \sum_{i=0}^{S} A_i\lambda^{S-i}-\lambda^{S+1}I & \sum_{i=0}^{T} B_i\lambda^{T-i} \end{bmatrix} = n.$$

Let $P(\lambda) = A_S + A_{S-1}\lambda + \cdots + A_0\lambda^S - \lambda^{S+1}I$ be a matrix polynomial of λ. Then, $P(\lambda)$ loses its rank for any $\lambda \in \left\{\lambda \in \mathbb{C} : \det\left(\sum_{i=0}^{S} A_i\lambda^{S-i}-\lambda^{S+1}I\right) = 0\right\}$. Thus, only for such λ's does it need to be ensured that

$$\rho\begin{bmatrix} \sum_{i=0}^{S} A_i\lambda^{S-i}-\lambda^{S+1}I & \sum_{i=0}^{T} B_i\lambda^{T-i} \end{bmatrix} = n,$$

which is condition (5.8).

For the extended augmented system (5.7), we evaluate $\rho\begin{bmatrix} \bar{A}-\lambda I & \bar{B} \end{bmatrix}$ following similar row and

column operations to result in,

$$\rho\begin{bmatrix}\bar{A}-\lambda I & \bar{B}\end{bmatrix} = \rho\left[\begin{array}{ccccc|ccccc} A_0-\lambda I & \cdots & A_S & 0 & \cdots & 0 & \cdots & B_T & \cdots & B_1 & B_0 \\ I_n & -\lambda I & 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 \\ 0 & I_n & -\lambda I & \vdots & \vdots & 0 & \cdots & 0 & \cdots & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots & \vdots & \cdots & \vdots & \cdots & \vdots & \vdots \\ 0 & \cdots & 0 & I_n & -\lambda I & 0 & \cdots & 0 & \cdots & 0 & 0 \\ \hline 0 & \cdots & 0 & 0 & 0 & -\lambda I & I_m & 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & -\lambda I & I_m & \vdots & 0 & 0 \\ \vdots & \cdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & 0 & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & -\lambda I & I_m & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & -\lambda I & I_m \end{array}\right]$$

$$= \rho\begin{bmatrix}\sum_{i=0}^{S} A_i\lambda^{S-i} - \lambda^{S+1} I & \sum_{i=0}^{T} B_i\lambda^{T-i}\end{bmatrix} + n\bar{S} + m\bar{T},$$

where we have used the fact that the padded zero columns do not affect the row rank, whereas the $\bar{S}$ number of I_n and $\bar{T}$ number of I_m matrices contribute $n\bar{S}+m\bar{T}$ to the row rank. For the controllability of the extended augmented system (5.7), we need $\rho\begin{bmatrix}\bar{A}-\lambda I & \bar{B}\end{bmatrix} = n + n\bar{S} + m\bar{T}$, which requires

$$\rho\begin{bmatrix}\sum_{i=0}^{S} A_i\lambda^{S-i} - \lambda^{S+1} I & \sum_{i=0}^{T} B_i\lambda^{T-i}\end{bmatrix} = n,$$

which is again condition (5.8). This shows that the augmented and extended augmented systems are controllable if and only if (5.8) holds. This completes the proof. □

We next work towards deriving the conditions for the observability of the extended augmented system. To this end, we introduce an augmented output vector as defined by

$$Y_k = \begin{bmatrix} y_k^{\mathrm{T}} & u_{k-T}^{\mathrm{T}} & \overbrace{0 \quad \cdots \quad 0}^{T-1}\end{bmatrix}^{\mathrm{T}} = \mathcal{C}X_k, \quad \mathcal{C} = \left[\begin{array}{ccccc|ccccc} \overbrace{C & 0 & 0 & \cdots & 0}^{S+1} & \overbrace{0 & 0 & 0 & \cdots & 0}^{T} \\ \hline 0 & 0 & 0 & \cdots & 0 & I_m & 0 & 0 & \cdots & 0 \end{array}\right].$$

In the same spirit, we can obtain an extended augmented output vector by incorporating the fictitious states corresponding to the delayed states from x_{k-S-1} to $x_{k-\bar{S}}$ and to the delayed inputs from x_{k-T-1} to $x_{k-\bar{T}}$ as follows,

$$\bar{Y}_k = \begin{bmatrix} y_k^{\mathrm{T}} & u_{k-\bar{T}}^{\mathrm{T}} & \overbrace{0 \quad \cdots \quad 0}^{\bar{T}-1}\end{bmatrix}^{\mathrm{T}} = \bar{\mathcal{C}}\bar{X}_k, \quad \bar{\mathcal{C}} = \left[\begin{array}{ccccc|ccccc} \overbrace{C & 0 & 0 & \cdots & 0}^{\bar{S}+1} & \overbrace{0 & 0 & 0 & \cdots & 0}^{\bar{T}} \\ \hline 0 & 0 & 0 & \cdots & 0 & I_m & 0 & 0 & \cdots & 0 \end{array}\right].$$

We will now study the observability property of the augmented systems (5.5) and (5.7).

Theorem 5.2. *The delay-free augmented systems* (5.5) *is observable if and only if*

$$\rho\left[\begin{array}{cc}\sum_{i=0}^{S} A_i^T\lambda^{S-i} - \lambda^{S+1}I & C^T\end{array}\right] = n, \tag{5.9}$$

for any $\lambda \in \left\{\lambda \in \mathbb{C} : det\left(\sum_{i=0}^{S} A_i\lambda^{S-i} - \lambda^{S+1}I\right) = 0 \text{ and } \lambda \neq 0\right\}$, *and*

$$\rho(A_S) = n. \tag{5.10}$$

The extended augmented system (5.7) *is detectable if and only if* (5.9) *holds for any* $\lambda \in \{\lambda \in \mathbb{C} : |\lambda| \geq 1\}$.

Proof: By duality, the observability of $(A, \mathcal{C})$ implies the controllability of $(A^{\mathrm{T}}, \mathcal{C}^{\mathrm{T}})$. Thus, $(A, \mathcal{C})$ is observable if and only if $\left[\begin{array}{cc}A^{\mathrm{T}} - \lambda I & \mathcal{C}^{\mathrm{T}}\end{array}\right]$ has a full row rank of $(S+1)n + mT$. We evaluate $\rho\left[\begin{array}{cc}A^{\mathrm{T}} - \lambda I & \mathcal{C}^{\mathrm{T}}\end{array}\right]$. Moving the last column block to beginning of the right side partition and then adding λ times this column to the next column on its right and then repeating this for the remaining columns, results in,

$$\rho\left[\begin{array}{cc}A^{\mathrm{T}} - \lambda I & \mathcal{C}^{\mathrm{T}}\end{array}\right] = \rho\left[\begin{array}{ccccc|ccccccc} A_0^{\mathrm{T}}-\lambda I & I_n & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & C^{\mathrm{T}} \\ A_1^{\mathrm{T}} & -\lambda I & I_n & \cdots & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & 0 \\ A_2^{\mathrm{T}} & 0 & -\lambda I & \ddots & \vdots & 0 & \cdots & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & I_n & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ A_S^{\mathrm{T}} & \cdots & 0 & 0 & -\lambda I & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \hline B_T^{\mathrm{T}} & 0 & \cdots & 0 & 0 & I_m & 0 & 0 & \cdots & 0 & 0 & 0 \\ B_{T-1}^{\mathrm{T}} & 0 & \cdots & 0 & 0 & 0 & I_m & 0 & 0 & \vdots & 0 & 0 \\ B_{T-2}^{\mathrm{T}} & 0 & \cdots & 0 & 0 & 0 & 0 & I_m & 0 & 0 & 0 & 0 \\ \vdots & 0 & \cdots & 0 & 0 & 0 & 0 & \ddots & \ddots & 0 & 0 & 0 \\ B_1^{\mathrm{T}} & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & I_m & 0 & 0 \end{array}\right].$$

Similarly, we can cancel all the B_i entries in the left partition using the identity columns from the right

partition, which gives

$$\rho\begin{bmatrix}A^{\mathrm{T}}-\lambda I & \mathcal{C}^{\mathrm{T}}\end{bmatrix}=\rho\left[\begin{array}{ccccc|ccccccc}A_0^{\mathrm{T}}-\lambda I & I_n & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & C^{\mathrm{T}}\\ A_1^{\mathrm{T}} & -\lambda I & I_n & \cdots & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & 0\\ A_2^{\mathrm{T}} & 0 & -\lambda I & \ddots & \vdots & 0 & \cdots & \cdots & \cdots & 0 & 0 & 0\\ \vdots & \vdots & \ddots & \ddots & I_n & \vdots & \cdots & \cdots & \vdots & \vdots & \vdots & \vdots\\ A_S^{\mathrm{T}} & 0 & 0 & \cdots & -\lambda I & 0 & \cdots & 0 & \cdots & 0 & 0 & 0\\ \hline 0 & \cdots & 0 & 0 & 0 & I_m & 0 & 0 & \cdots & 0 & 0 & 0\\ 0 & \cdots & 0 & 0 & 0 & 0 & I_m & 0 & 0 & \vdots & 0 & 0\\ \vdots & \vdots & \vdots & \vdots & \vdots & 0 & 0 & I_m & 0 & 0 & \cdots & 0\\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & \ddots & 0 & 0 & 0\\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & I_m & 0 & 0\end{array}\right].$$

The bottom half rows involve T number of I_m matrices which contribute mT to the rank. We evaluate the remaining non-zero partition in the top half as,

$$\rho\left[\begin{array}{ccccc|c}A_0^{\mathrm{T}}-\lambda I & I_n & 0 & \cdots & 0 & C^{\mathrm{T}}\\ A_1^{\mathrm{T}} & -\lambda I & I_n & \cdots & 0 & 0\\ A_2^{\mathrm{T}} & 0 & -\lambda I & \ddots & \vdots & 0\\ \vdots & \vdots & 0 & \ddots & I_n & \vdots\\ A_S^{\mathrm{T}} & 0 & 0 & \cdots & -\lambda I & 0\end{array}\right].$$

For the case when $\lambda \neq 0$, we can perform some elementary operations on the above matrix that gives

$$\rho\left[\begin{array}{ccccc|c}A_0^{\mathrm{T}}-\lambda I+\frac{1}{\lambda}A_1^{\mathrm{T}}+\cdots+\frac{1}{\lambda^H}A_H^{\mathrm{T}} & 0 & 0 & \cdots & 0 & C^{\mathrm{T}}\\ 0 & I_n & 0 & \cdots & 0 & 0\\ 0 & 0 & I_n & \ddots & \vdots & 0\\ \vdots & \vdots & \ddots & \ddots & 0 & \vdots\\ 0 & 0 & 0 & \cdots & I_n & 0\end{array}\right].$$

The S number of I_n matrices contribute Sn to the rank. Thus, $(A, \mathcal{C})$ is observable if and only if

$$\rho\begin{bmatrix}A_0^{\mathrm{T}}-\lambda I+\frac{1}{\lambda}A_1^{\mathrm{T}}+\cdots+\frac{1}{\lambda^H}A_H^{\mathrm{T}} & C^{\mathrm{T}}\end{bmatrix}=n,$$

or equivalently,

$$\rho\begin{bmatrix}\sum_{i=0}^{S}A_i^{\mathrm{T}}\lambda^{S-i}-\lambda^{S+1}I & C^{\mathrm{T}}\end{bmatrix}=n,$$

for any $\lambda \in \left\{\lambda \in \mathbb{C} : \det\left(\sum_{i=0}^{S} A_i \lambda^{S-i} - \lambda^{S+1} I\right) = 0 \text{ and } \lambda \neq 0\right\}$, which is condition (5.9).

If $\lambda = 0$ then

$$\rho \left[\begin{array}{ccccc|c} A_0^{\mathrm{T}} & I_n & 0 & \cdots & 0 & C^{\mathrm{T}} \\ A_1^{\mathrm{T}} & 0 & I_n & \cdots & 0 & 0 \\ A_2^{\mathrm{T}} & 0 & 0 & \ddots & \vdots & 0 \\ \vdots & \vdots & \ddots & \ddots & I_n & \vdots \\ A_S^{\mathrm{T}} & \cdots & 0 & 0 & 0 & 0 \end{array}\right] = (S+1)n$$

if and only if

$$\rho(A_S) = n,$$

which is condition (5.10).

For the case of the extended augmented system, it can be readily verified that for $\lambda \neq 0$, the first condition (5.9) remains the same due to the additional identity matrices, as in the proof of the controllability results in Theorem 5.1. The second condition in this case becomes $\rho(A_{\bar{S}}) = n$. However, this condition cannot be satisfied because $A_{\bar{S}} = 0$. As a result, the observability condition (5.10) for the extended augmented system loses rank for $\lambda = 0$. Since $\lambda = 0$ is a stable eigenvalue, we still have the detectability condition satisfied. In the special case, when $\bar{S} = S$, the extended augmented system is observable if and only if (5.9) and (5.10) hold. This completes the proof. □

Remark 5.1. *The observability condition for the augmented system* (5.5) *can also be relaxed to detectability, in which case we require only the first condition* (5.9) *to hold for* $|\lambda| \geq 1$.

5.3. Q-learning Based State Feedback Stabilization of Time Delay Systems

In this section, we will present a Q-learning scheme for learning the optimal control parameters that uplifts the requirement of the knowledge of the system dynamics and the delays.

To this end, we rewrite the original utility function for (5.1) in terms of the extended augmented state vector (5.6) as,

$$r_k = \bar{X}_k^{\mathrm{T}} \bar{Q} \bar{X}_k + u_k^{\mathrm{T}} R u_k, \tag{5.11}$$

where

$$\bar{Q} = \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix}.$$

Since the system is now in a delay-free form, we can readily compute the optimal controller. From optimal control theory [37], we know that there exists a unique optimal control sequence

$$u_k^* = \bar{K}^* \bar{X}_k = -(R + \bar{B}^{\mathrm{T}} \bar{P}^* \bar{B})^{-1} \bar{B}^{\mathrm{T}} \bar{P}^* \bar{A} \bar{X}_k \tag{5.12}$$

that minimizes (5.2) under the conditions of the stabilizability of $\left(\bar{A}, \bar{B}\right)$ and detectability of $\left(\bar{A}, \sqrt{\bar{Q}}\right)$, where $\bar{P}^* = (\bar{P}^*)^{\mathrm{T}}$ is the unique positive semi-definite solution to the following ARE,

$$\bar{A}^{\mathrm{T}} \bar{P} \bar{A} - \bar{P} + \bar{Q} - \bar{A}^{\mathrm{T}} \bar{P} \bar{B} (R + \bar{B}^{\mathrm{T}} \bar{P} \bar{B})^{-1} \bar{B}^{\mathrm{T}} \bar{P} \bar{A} = 0. \tag{5.13}$$

We will now obtain a Q-function for the extended augmented system. Consider a stabilizing control policy $u_k = \bar{K} \bar{X}_k$, which is not necessarily optimal with respect to our utility function. Corresponding to this controller, there is a cost given by the following value function that represents the infinite horizon cost of executing the control starting from state $\bar{X}_k$ from time k to time ∞ as follows,

$$V_{\bar{K}}(\bar{X}_k) = \bar{X}_k^{\mathrm{T}} \bar{P} \bar{X}_k. \tag{5.14}$$

The above infinite horizon value function can be recursively written as,

$$V_{\bar{K}}(\bar{X}_k) = \bar{X}_k^{\mathrm{T}} \bar{Q} \bar{X}_k + \bar{X}_k^{\mathrm{T}} \bar{K}^{\mathrm{T}} R \bar{K} \bar{X}_k + V_{\bar{K}}(\bar{X}_{k+1}).$$

Similar to the value function above, we can define a Quality function (Q-function) which gives the value of executing an arbitrary control u_k instead of $u_k = \bar{K} \bar{X}_k$ at time k and then following policy $\bar{K}$ from time $k+1$,

$$Q_{\bar{K}}(\bar{X}_k, u_k) = \bar{X}_k^{\mathrm{T}} \bar{Q} \bar{X}_k + u_k^{\mathrm{T}} R u_k + V_{\bar{K}}(\bar{X}_{k+1}). \tag{5.15}$$

Substituting the dynamics (5.7) in the above expression results in,

$$\begin{aligned} Q_{\bar{K}}(\bar{X}_k, u_k) &= \bar{X}_k^{\mathrm{T}} \bar{Q} \bar{X}_k + u_k^{\mathrm{T}} R u_k + \bar{X}_{k+1}^{\mathrm{T}} \bar{P} \bar{X}_{k+1} \\ &= \bar{X}_k^{\mathrm{T}} \bar{Q} \bar{X}_k + u_k^{\mathrm{T}} R u_k + \left(\bar{A} \bar{X}_k + \bar{B} u_k\right)^{\mathrm{T}} \bar{P} \left(\bar{A} \bar{X}_k + \bar{B} u_k\right), \end{aligned}$$

or equivalently,

$$Q_{\bar{K}}(\bar{X}_k, u_k) = \begin{bmatrix} \bar{X}_k \\ u_k \end{bmatrix}^{\mathrm{T}} H \begin{bmatrix} \bar{X}_k \\ u_k \end{bmatrix}, \tag{5.16}$$

where

$$H \triangleq \begin{bmatrix} H_{XX} & H_{Xu} \\ H_{uX} & H_{uu} \end{bmatrix} = \begin{bmatrix} \bar{Q}+\bar{A}^{\mathrm{T}}\bar{P}\bar{A} & \bar{A}^{\mathrm{T}}\bar{P}\bar{B} \\ \bar{B}^{\mathrm{T}}\bar{P}\bar{A} & R+\bar{B}^{\mathrm{T}}\bar{P}\bar{B} \end{bmatrix}.$$

When $\bar{K} = \bar{K}^*$, we have $\bar{P} = \bar{P}^*$, $Q_{\bar{K}} = Q^*$, and the optimal LQR controller can be obtained by solving $\frac{\partial}{\partial u_k} Q^* = 0$ for u_k, which corresponds to (5.12).

It can be seen that the problem of finding an optimal controller boils down to finding the optimal matrix H^* or the optimal Q-function Q^*.

Q-learning is a model-free learning technique that estimates the optimal Q-function without requiring the knowledge of system dynamics. It does so by means of the following Bellman Q-learning equation,

$$Q_{\bar{K}}(\bar{X}_k, u_k) = \bar{X}_k^{\mathrm{T}}\bar{Q}\bar{X}_k + u_k^{\mathrm{T}}Ru_k + Q_K(\bar{X}_{k+1}, \bar{K}\bar{X}_{k+1}), \tag{5.17}$$

which is obtained by substituting $V_{\bar{K}}(\bar{X}_k) = Q_{\bar{K}}(\bar{X}_k, \bar{K}X_k)$ in (2.9). Let

$$z_k = \begin{bmatrix} \bar{X}_k \\ u_k \end{bmatrix}.$$

Then, by definition (2.11), we can write equation (2.23) as,

$$z_k^{\mathrm{T}} H z_k = \bar{X}_k^{\mathrm{T}}\bar{Q}\bar{X}_k + u_k^{\mathrm{T}}Ru_k + z_{k+1}^{\mathrm{T}} H z_{k+1}, \tag{5.18}$$

which is linear in the unknown matrix H. We can perform the following parametrization on this equation,

$$\begin{aligned} Q_{\bar{K}}(z_k) &\triangleq Q_{\bar{K}}(\bar{X}_k, u_k) \\ &= \bar{H}^{\mathrm{T}}\bar{z}_k, \end{aligned}$$

where

$$\begin{aligned} \bar{H} &= \mathrm{vec}(H) \\ &\triangleq [h_{11}, 2h_{12}, \cdots, 2h_{1l}, h_{22}, 2h_{23}, \cdots, 2h_{2l}, \cdots h_{ll}]^{\mathrm{T}} \in \mathbb{R}^{l(l+1)/2}, \end{aligned}$$

h_{ii} are the elements of matrix H and $l = n + n\bar{S} + m\bar{T} + m$. The regressor $\bar{z}_k \in \mathbb{R}^{l(l+1)/2}$ is defined as the following quadratic basis set,

$$\bar{z} = \left[z_1^2, z_1 z_2, \cdots, z_1 z_l, z_2^2, z_2 z_3, \cdots, z_2 z_l, \cdots z_l^2\right]^{\mathrm{T}}.$$

With this parametrization, we have, from (5.18), the following equation,

$$\bar{H}^{\mathrm{T}}(\bar{z}_k - \bar{z}_{k+1}) = \bar{X}_k^{\mathrm{T}}\bar{Q}\bar{X}_k + u_k^{\mathrm{T}}Ru_k. \tag{5.19}$$

In what follows, we present an iterative Q-learning algorithm to learn the optimal control parameters.

Algorithm 1: State Feedback Iterative Q-learning Value Iteration Algorithm for Time Delay Systems

Initialization. Start with an arbitrary policy $u_k^0 = v_k$ with v_k being the exploration signal. Set $H^0 \leftarrow 0$.

Collect Data. Apply the initial policy u^0 to collect L datasets of $(\bar{X}_k, u_k)$ along with their quadratic terms.

Value Update. Compute the least-squares solution of

$$\left(\bar{H}^j\right)^{\mathrm{T}} \bar{z}_k = \bar{X}_k^{\mathrm{T}}\bar{Q}\bar{X}_k + u_k^{\mathrm{T}}Ru_k + \left(\bar{H}^{j-1}\right)^{\mathrm{T}} \bar{z}_{k+1}. \tag{5.20}$$

Policy Update. Determine an improved policy using

$$u_k^{j+1} = -\left(H_{uu}^j\right)^{-1} H_{uX}^j \bar{X}_k. \tag{5.21}$$

Repeat and Stop. Then, for $j = 1, 2, \cdots$, repeat the steps in (5.20) and (5.21) until convergence,

$$\left\|\bar{H}^j - \bar{H}^{j-1}\right\| < \varepsilon, \tag{5.22}$$

for some small positive constant ε.

Algorithm 1 presents an iterative algorithm based on Q-learning. We apply an arbitrary initial control with an exploration signal v_k such that the rank condition (5.24) is satisfied. Based on this data, we solve the recursive Q-learning Bellman equation (5.20) in the value update step. In the next step, we perform a minimization of the Q-function with respect to u_k, which gives us a new policy. This policy will be evaluated in the next iteration. These iterations are carried out until we see no further updates in the estimate of the matrix H within a sufficiently small range specified by the positive constant ε. Once this criterion is met, the algorithm is considered to have converged to the (near) optimal solution. The data

matrices in the recursive iterations are given as,

$$\begin{aligned}\Phi &= \left[\bar{z}_{k-L+1},\ \bar{z}_{k-L+2},\ \cdots\ \bar{z}_k\right],\\ \Upsilon^{j-1} &= \Big[r(\bar{X}_{k-L+1},u_{k-L+1}) + \left(\bar{H}^{j-1}\right)^{\mathrm{T}}\bar{z}_{k-L+2}, r(\bar{X}_{k-L+2},u_{k-L+2}) + \left(\bar{H}^{j-1}\right)^{\mathrm{T}}\bar{z}_{k-L+3},\cdots,\\ &\qquad r(\bar{X}_k,u_k) + \left(\bar{H}^{j-1}\right)^{\mathrm{T}}\bar{z}_{k+1}\Big]^{\mathrm{T}}.\end{aligned}$$

The least-squares solution of (5.20) is given by,

$$\bar{H}^j = (\Phi\Phi^{\mathrm{T}})^{-1}\Phi\Upsilon^{j-1}, \tag{5.23}$$

It is important to note that since $u_k = \bar{K}\bar{X}_k$ is linearly dependent on $\bar{X}_k$, the least-squares problem cannot be solved unless we inject an independent exploration signal v_k in u_k in order to guarantee the invertibility of $\Phi\Phi^{\mathrm{T}}$ in (5.23). In other words, the following rank condition should hold,

$$\text{rank}(\Phi) = l(l+1)/2. \tag{5.24}$$

We next show the convergence of Algorithm 1.

Theorem 5.3. *Under the stabilizability and detectability conditions on $(\bar{A},\bar{B})$ and $(\bar{A},\sqrt{\bar{Q}})$, respectively, the iterative Q-learning Algorithm 1 generates a sequence of controls $\{u_k^j, j=1,2,3,...\}$ that converges to the optimal feedback controller given in (5.12) as $j\to\infty$ if the rank condition* (5.24) *is satisfied.*

Proof: The proposed iterative algorithm is based on the following Q-learning equation,

$$\begin{bmatrix}\bar{X}_k\\ u_k\end{bmatrix}^{\mathrm{T}} H \begin{bmatrix}\bar{X}_k\\ u_k\end{bmatrix} = \bar{X}_k^{\mathrm{T}}\bar{Q}\bar{X}_k + u_k^{\mathrm{T}}Ru_k + \begin{bmatrix}\bar{X}_{k+1}\\ u_{k+1}\end{bmatrix}^{\mathrm{T}} H \begin{bmatrix}\bar{X}_{k+1}\\ u_{k+1}\end{bmatrix}.$$

Using the feedback policy $\bar{K}$ and the definition (5.16), we have

$$Q_{\bar{K}}(\bar{X}_k,\bar{K}\bar{X}_k) = \bar{X}_k^{\mathrm{T}}\bar{Q}\bar{X}_k + \bar{X}_k^{\mathrm{T}}\bar{K}^{\mathrm{T}}R\bar{K}\bar{X}_k + Q_{\bar{K}}(\bar{X}_{k+1},\bar{K}\bar{X}_{k+1}).$$

Noticing that $Q_{\bar{K}}(\bar{X}_k,\bar{K}\bar{X}_k) = V_{\bar{K}}(\bar{X}_k)$, we have the following Bellman equation,

$$V_{\bar{K}}(\bar{X}_k) = \bar{X}_k^{\mathrm{T}}\bar{Q}\bar{X}_k + \bar{X}_k^{\mathrm{T}}\bar{K}^{\mathrm{T}}R\bar{K}\bar{X}_k + V_{\bar{K}}(\bar{X}_{k+1}).$$

As the cost of following policy $\bar{K}$ is quadratic in the state as given by (5.14), we have

$$\bar{X}_k^{\mathrm{T}}\bar{P}\bar{X}_k = \bar{X}_k^{\mathrm{T}}\bar{Q}\bar{X}_k + \bar{X}_k^{\mathrm{T}}\bar{K}^{\mathrm{T}}R\bar{K}\bar{X}_k + \bar{X}_{k+1}^{\mathrm{T}}\bar{P}\bar{X}_{k+1},$$

or

$$\bar{X}_k^{\mathrm{T}}\bar{P}\bar{X}_k = \bar{X}_k^{\mathrm{T}}\bar{Q}\bar{X}_k + \bar{X}_k^{\mathrm{T}}\bar{K}^{\mathrm{T}}R\bar{K}\bar{X}_k + \bar{X}_k^{\mathrm{T}}(\bar{A}+\bar{B}\bar{K})^{\mathrm{T}}\bar{P}(\bar{A}+\bar{B}\bar{K})\bar{X}_k,$$

which holds for all $\bar{X}_k$. We thus have the following Lyapunov equation,

$$(\bar{A}+\bar{B}\bar{K})^{\mathrm{T}}\bar{P}(\bar{A}+\bar{B}\bar{K}) - \bar{P} + \bar{Q} + \bar{K}^{\mathrm{T}}R\bar{K} = 0. \tag{5.25}$$

On the other hand, by the definition of matrix H in (5.16), the policy update step computes the policy $\bar{K}$ as

$$\bar{K} = -(R + \bar{B}^{\mathrm{T}}\bar{P}\bar{B})^{-1}\bar{B}^{\mathrm{T}}\bar{P}\bar{A}. \tag{5.26}$$

Substituting (5.26) in (5.25) results in the ARE (5.13). If the rank condition (5.24) holds then the value update equation (5.20) is uniquely solvable. Then, the iterations on the value update and policy update steps corresponding to (5.20) and (5.21) are equivalent to the recursion on the following Riccati Difference Equation (RDE),

$$\bar{P}^{j+1} = \bar{A}^{\mathrm{T}}\bar{P}^j\bar{A} + \bar{Q} - \bar{A}^{\mathrm{T}}\bar{P}^j\bar{B}(R + \bar{B}^{\mathrm{T}}\bar{P}^j\bar{B})^{-1}\bar{B}^{\mathrm{T}}\bar{P}^j\bar{A}.$$

The recursions on the RDE converge to the solution of the ARE under the stabilizability and detectability conditions of $(\bar{A}, \bar{B})$ and $(\bar{A}, \sqrt{\bar{Q}})$, respectively [34] . Thus, the iterative Q-learning algorithm converges to the optimal stabilizing solution $\bar{K}^*$. As a result, we have $\bar{X}_k$ and x_k converging to zero as $k \to \infty$. This completes the proof. □

Remark 5.2. *Compared to the previous works [73], [74], the proposed scheme relaxes the assumption of the existence of a bicausal change of coordinates. Furthermore, unlike in [73], [74], the information of the state and input delays (both the number and lengths of the delays) is not needed in our proposed scheme.*

5.4. Q-learning Based Output Feedback Stabilization of Time Delay Systems

This section will present a Q-learning based control algorithm to stabilize the time delay system (5.1) using only the measurements of the system output instead of the full state. We recall from [38] the following lemma, which allows the reconstruction of the system state by means of the delayed measurements of the system inputs and outputs. Applying the parametrization (2.18) from Chapter 2 for the augmented system (5.5), we have

$$X_k = M_y Y_{k-1,k-N} + M_u u_{k-1,k-N}, \tag{5.27}$$

where $N \leq n(S+1) + mT$ is an upper bound on the system's observability index, $u_{k-1,k-N} \in \mathbb{R}^{mN}$ and $Y_{k-1,k-N} \in \mathbb{R}^{(p+m)N}$ are the input and augmented output data vectors and M_u and M_y are formed using the augmented system matrices $(A, B, \mathcal{C})$.

Remark 5.3. *The state parametrization* (5.27) *involves the invertability of the observability matrix. To ensure the observability of the extended augmented pair* $(\bar{A}, \bar{\mathcal{C}})$*, we will assume in this section that* $\bar{S} = S$*.*

For the extended augmented system (5.7), we have the following parametrization of the extended augmented state,

$$\bar{X}_k = \bar{M}_y \bar{Y}_{k-1,k-N} + \bar{M}_u u_{k-1,k-N}, \tag{5.28}$$

where $\bar{Y}_{k-1,k-N}$, $\bar{M}_u$ and $\bar{M}_y$ are formed using the extended augmented system matrices $(\bar{A}, \bar{B}, \bar{\mathcal{C}})$ and N now represents an upper bound on the extended augmented system's observability index. It can be easily verified that substitution of (5.28) in (5.16) results in,

$$\begin{aligned} Q_{\bar{K}} &= \begin{bmatrix} u_{k-1,k-N} \\ \bar{Y}_{k-1,k-N} \\ u_k \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathcal{H}_{\bar{u}\bar{u}} & \mathcal{H}_{\bar{u}\bar{y}} & \mathcal{H}_{\bar{u}u} \\ \mathcal{H}_{\bar{y}\bar{u}} & \mathcal{H}_{\bar{y}\bar{y}} & \mathcal{H}_{\bar{y}u} \\ \mathcal{H}_{u\bar{u}} & \mathcal{H}_{u\bar{y}} & \mathcal{H}_{uu} \end{bmatrix} \begin{bmatrix} u_{k-1,k-N} \\ \bar{Y}_{k-1,k-N} \\ u_k \end{bmatrix} \\ &\triangleq \zeta_k^{\mathrm{T}} \mathcal{H} \zeta_k, \end{aligned} \tag{5.29}$$

where

$$\begin{aligned} \zeta_k &= \begin{bmatrix} u_{k-1,k-N}^{\mathrm{T}} & \bar{Y}_{k-1,k-N}^{\mathrm{T}} & u_k^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}, \\ \mathcal{H} &= \mathcal{H}^{\mathrm{T}} \in \mathbb{R}^{(mN+(p+m)N+m)\times(mN+(p+m)N+m)}, \end{aligned}$$

and the partitioned matrices are defined as,

$$\begin{aligned}
\mathcal{H}_{\bar{u}\bar{u}} &= \bar{M}_u^{\mathrm{T}}(\bar{Q} + \bar{A}^{\mathrm{T}}\bar{P}\bar{A})\bar{M}_u \in \mathbb{R}^{mN\times mN},\\
\mathcal{H}_{\bar{u}\bar{y}} &= \bar{M}_u^{\mathrm{T}}(\bar{Q} + \bar{A}^{\mathrm{T}}\bar{P}\bar{A})\bar{M}_y \in \mathbb{R}^{mN\times (p+m)N},\\
\mathcal{H}_{\bar{u}u} &= \bar{M}_u^{\mathrm{T}}\bar{A}^{\mathrm{T}}\bar{P}\bar{B} \in \mathbb{R}^{mN\times m},\\
\mathcal{H}_{\bar{y}\bar{y}} &= \bar{M}_y^{\mathrm{T}}(\bar{Q} + \bar{A}^{\mathrm{T}}\bar{P}\bar{A})\bar{M}_y \in \mathbb{R}^{(p+m)N\times (p+m)N}\\
\mathcal{H}_{\bar{y}u} &= \bar{M}_y^{\mathrm{T}}\bar{A}^{\mathrm{T}}\bar{P}\bar{B} \in \mathbb{R}^{(p+m)N\times m},\\
\mathcal{H}_{uu} &= R + \bar{B}^{\mathrm{T}}\bar{P}\bar{B} \in \mathbb{R}^{m\times m}.
\end{aligned} \tag{5.30}$$

To obtain the optimal controller, we perform the minimization of the optimal Q-function $Q^*_{\bar{K}}$ with $\mathcal{H}^*$ being the optimal $\mathcal{H}$. Setting $\frac{\partial}{\partial u_k}Q^*_{\bar{K}} = 0$ and solving for u_k result in our output feedback LQR control law,

$$u_k^* = -(\mathcal{H}^*_{uu})^{-1}\left(\mathcal{H}^*_{u\bar{u}}u_{k-1,k-N} + \mathcal{H}^*_{u\bar{y}}\bar{Y}_{k-1,k-N}\right). \tag{5.31}$$

Now that we have an output feedback form of the Q-function for the extended augmented time delay system, the next step is to learn the optimal Q-function $Q^*_{\bar{K}}$ and the corresponding output feedback optimal controller (5.31).

Consider the state feedback Q-learning equation (5.18). We employ the equivalent output feedback Q-function to write this equation as,

$$\zeta_k^{\mathrm{T}}\mathcal{H}\zeta_k = \bar{Y}_k^{\mathrm{T}}\bar{Q}_y\bar{Y}_k + u_k^{\mathrm{T}}Ru_k + \zeta_{k+1}^{\mathrm{T}}\mathcal{H}\zeta_{k+1}. \tag{5.32}$$

It should be noted that in the output feedback learning, we apply the user-defined weighting matrix $\bar{Q}_y$ to the outputs. The term $\bar{X}_k^{\mathrm{T}}Q\bar{X}_k$ can be replaced with $\bar{Y}_k^{\mathrm{T}}\bar{Q}_y\bar{Y}_k$ without requiring the knowledge of $\bar{\mathcal{C}}$ when $\bar{Q} = \bar{\mathcal{C}}^{\mathrm{T}}\bar{Q}_y\bar{\mathcal{C}}$ and $\bar{Y}_k = \bar{\mathcal{C}}X_k$, where $\bar{Y}_k$ is measurable. Here, u_{k+1} is computed as,

$$u_{k+1} = -(\mathcal{H}_{uu})^{-1}\left(\mathcal{H}_{u\bar{u}}\bar{u}_{k,k-N+1} + \mathcal{H}_{u\bar{y}}\bar{Y}_{k,k-N+1}\right).$$

So, (5.32) is the Bellman equation for the Q-function in the output feedback form for which we will develop a reinforcement learning algorithm. Next, we will parameterize the Q-function in (5.29) so that we can separate the unknown matrix $\mathcal{H}$.

Consider the output feedback Q-function in (2.19) which can be linearly parametrized as,

$$Q_{\bar{K}} = \bar{\mathcal{H}}^{\mathrm{T}}\bar{z}_k, \tag{5.33}$$

where $\bar{\mathcal{H}} = \text{vec}(\mathcal{H}) \in \mathbb{R}^{l(l+1)/2} \triangleq [\mathcal{H}_{11}, 2\mathcal{H}_{12}, \cdots, 2\mathcal{H}_{1l}, \mathcal{H}_{22}, 2\mathcal{H}_{23}, \cdots, 2\mathcal{H}_{2l}, \cdots \mathcal{H}_{ll}]^{\mathrm{T}}$ is the vector that contains the upper triangular portion of matrix $\mathcal{H}$, where $l = mN + (p+m)N + m$. Since $\mathcal{H}$ is symmetric, the off-diagonal entries are included as $2\mathcal{H}_{ij}$. The regression vector $\bar{\zeta}_k \in \mathbb{R}^{l(l+1)/2}$ is defined as the quadratic basis set,

$$\bar{\zeta} = \left[\zeta_1^2, \zeta_1\zeta_2, \cdots, \zeta_1\zeta_l, \zeta_2^2, \zeta_2\zeta_3, \cdots, \zeta_2\zeta_l, \cdots \zeta_l^2\right]^{\mathrm{T}}.$$

With the parametrization (5.33), we have the following equation,

$$\bar{\mathcal{H}}^{\mathrm{T}}\bar{\zeta}_k = \bar{Y}_k^{\mathrm{T}}\bar{Q}_y\bar{Y}_k + u_k^{\mathrm{T}}Ru_k + \bar{\mathcal{H}}^{\mathrm{T}}\bar{\zeta}_{k+1}. \tag{5.34}$$

We can now utilize iterative Q-learning to learn our output feedback Q-function matrix. In what follows, we present an iterative Q-learning algorithm to learn the optimal control parameters.

Algorithm 2: Output Feedback Iterative Q-learning Value Iteration Algorithm for Time Delay Systems

Initialization. Start with an arbitrary policy $u_k^0 = v_k$ with v_k being the exploration signal. Set $\mathcal{H}^0 \leftarrow I$.
Collect Data. Apply the initial policy u^0 to collect L datasets of $(\bar{Y}_k, u_k)$ along with their quadratic terms.
Value Update. Compute the least-squares solution of

$$\left(\bar{\mathcal{H}}^j\right)^{\mathrm{T}}\bar{z}_k = \bar{Y}_k^{\mathrm{T}}\bar{Q}_y\bar{Y}_k + u_k^{\mathrm{T}}Ru_k + \left(\bar{\mathcal{H}}^{j-1}\right)^{\mathrm{T}}\bar{\zeta}_{k+1}. \tag{5.35}$$

Policy Update. Determine an improved policy using

$$u_k^{j+1} = -\left(\mathcal{H}_{uu}^j\right)^{-1}\left(\mathcal{H}_{u\bar{u}}^j u_{k-1,k-N} + \mathcal{H}_{u\bar{y}}^j \bar{Y}_{k-1,k-N}\right). \tag{5.36}$$

Repeat and Stop. Then, for $j = 1, 2, \cdots$, repeat the steps in (5.20) and (5.21) until convergence,

$$\left\|\bar{\mathcal{H}}^j - \bar{\mathcal{H}}^{j-1}\right\| < \varepsilon, \tag{5.37}$$

for some small positive constant ε.

Algorithm 2 presents an output feedback iterative algorithm based on Q-learning. We apply an arbitrary initial control with an exploration signal v_k. Based on this data, we solve the recursive Q-learning Bellman equation (5.35) in the value update step. In the next step, we perform a minimization of the Q-function with respect to u_k, which gives us a new policy. This policy will be evaluated in the next iteration. These iterations are carried out until we see no further updates in the estimates of the matrix $\mathcal{H}$ for a sufficiently small positive constant ε. Once this criterion is met, the algorithm has converged to the (near) optimal solution. The data matrices in recursive iterations are given as,

$$\begin{aligned}\Phi &= \left[\bar{\zeta}_{k-L+1},\ \bar{\zeta}_{k-L+2},\ \cdots\ \bar{\zeta}_{k}\right],\\ \Upsilon^{j-1} &= \Big[r(\bar{Y}_{k-L+1},u_{k-L+1}) + \left(\bar{\mathcal{H}}^{j-1}\right)^{\mathrm{T}}\bar{\zeta}_{k-L+2}\ , r(\bar{Y}_{k-L+2},u_{k-L+2}) + \left(\bar{\mathcal{H}}^{j-1}\right)^{\mathrm{T}}\bar{\zeta}_{k-L+3},\cdots\\ &\qquad r(\bar{Y}_{k},u_{k}) + \left(\bar{\mathcal{H}}^{j-1}\right)^{\mathrm{T}}\bar{\zeta}_{k+1}\Big]^{\mathrm{T}}.\end{aligned}$$

The least-squares solution of (5.35) is given by,

$$\bar{\mathcal{H}}^{j} = (\Phi\Phi^{\mathrm{T}})^{-1}\Phi\Upsilon^{j-1}, \tag{5.38}$$

Similar to Algorithm 1, the following rank condition needs to be satisfied to solve the least-squares problem (5.38),

$$\text{rank}(\Phi) = l(l+1)/2. \tag{5.39}$$

Remark 5.4. *When $N < \bar{T}$, $\bar{Y}_{k-1,k-N}$ will contain entries from $u_{k-1,k-N}$ that will prevent the rank condition* (5.39) *from being satisfied even in the presence of an exploration signal v_k. However, in the proof of Theorem 5.2, we see that increasing T to an arbitrary large $\bar{T}$ does not affect the observability. Furthermore, the rank condition* (5.9) *corresponding to the original output y_k remains unchanged. This implies that the sum of the observability indices from the original output y_k also remains unchanged. Therefore, the augmented output vector $u_{k-\bar{T}}$ must be contributing to the observability of the extended states. This causes the sum of the observability indices corresponding to the augmented output vector $u_{k-\bar{T}}$ to increase to $m\bar{T}$, with each component contributing $\bar{T}$ equally. Thus, for an arbitrarily large $\bar{T}$, the observability index becomes $N = \bar{T}$ and the rank condition* (5.39) *can be satisfied.*

We next show the convergence of Algorithm 2.

Theorem 5.4. *Under the stabilizability and detectability conditions on $(\bar{A}, \bar{B})$ and $(\bar{A}, \sqrt{\bar{Q}})$, respectively,*

the iterative Q-learning Algorithm 2 generates a sequence of controls $\{u_k^j, j = 1, 2, 3, ...\}$ *that converges to the optimal feedback controller given in (5.12) as* $j \to \infty$ *if the rank condition* (5.39) *is satisfied.*

Proof: Consider the output feedback Q-learning equation

$$\zeta_k^{\mathrm{T}} \mathcal{H} \zeta_k = \bar{Y}_k^{\mathrm{T}} \bar{Q}_y \bar{Y}_k + u_k^{\mathrm{T}} R u_k + \zeta_{k+1}^{\mathrm{T}} \mathcal{H} \zeta_{k+1}.$$

Based on the definitions of ζ_k and $\mathcal{H}$ in (5.30), we have

$$\begin{bmatrix} \bar{u}_{k-1,k-N} \\ \bar{Y}_{k-1,k-N} \\ u_k \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} M & I \end{bmatrix}^{\mathrm{T}} H \begin{bmatrix} M & I \end{bmatrix} \begin{bmatrix} \bar{u}_{k-1,k-N} \\ \bar{Y}_{k-1,k-N} \\ u_k \end{bmatrix} = \bar{X}_k^{\mathrm{T}} \bar{Q} \bar{X}_k + u_k^{\mathrm{T}} R u_k$$
$$+ \begin{bmatrix} \bar{u}_{k,k-N+1} \\ \bar{Y}_{k,k-N+1} \\ u_{k+1} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} M & I \end{bmatrix}^{\mathrm{T}} H \begin{bmatrix} M & I \end{bmatrix} \begin{bmatrix} \bar{u}_{k,k-N+1} \\ \bar{Y}_{k,k-N+1} \\ u_{k+1} \end{bmatrix},$$

where $M = \begin{bmatrix} \bar{M}_u & \bar{M}_y \end{bmatrix}$ and $\bar{Q} = \bar{\mathcal{C}}^{\mathrm{T}} \bar{Q}_y \bar{\mathcal{C}}$.

Since $(A, \mathcal{C})$ is observable, we have

$$\bar{X}_k = M \begin{bmatrix} u_{k-1,k-N}^{\mathrm{T}} & \bar{Y}_{k-1,k-N}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}},$$

which results in the state feedback equation,

$$\begin{bmatrix} \bar{X}_k \\ u_k \end{bmatrix}^{\mathrm{T}} H \begin{bmatrix} \bar{X}_k \\ u_k \end{bmatrix} = \bar{X}_k^{\mathrm{T}} \bar{Q} \bar{X}_k + u_k^{\mathrm{T}} R u_k + \begin{bmatrix} \bar{X}_{k+1} \\ u_{k+1} \end{bmatrix}^{\mathrm{T}} H \begin{bmatrix} \bar{X}_{k+1} \\ u_{k+1} \end{bmatrix}.$$

Let

$$\bar{\mathcal{K}} = (\mathcal{H}_{uu})^{-1} \begin{bmatrix} \mathcal{H}_{u\bar{u}} & \mathcal{H}_{u\bar{y}} \end{bmatrix}.$$

Then, by the definition of $\mathcal{H}$, we have $\bar{\mathcal{K}} = M \bar{K}$ and the policy

$$u_k = \mathcal{K} \begin{bmatrix} u_{k-1,k-N}^{\mathrm{T}} & \bar{Y}_{k-1,k-N}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$$

is equivalent to the state feedback policy $u_k = \bar{K} \bar{X}_k$. Using the definition (5.16), we have

$$Q_{\bar{K}}(\bar{X}_k, \bar{K} \bar{X}_k) = \bar{X}_k^{\mathrm{T}} \bar{Q} \bar{X}_k + \bar{X}_k^{\mathrm{T}} \bar{K}^{\mathrm{T}} R \bar{K} \bar{X}_k + Q_{\bar{K}}(\bar{X}_{k+1}, \bar{K} \bar{X}_{k+1}).$$

Then, the remainder of the proof follows the arguments of Theorem 5.3. This completes the proof. □

5.5. Results

In this section, we test the proposed scheme using numerical simulation. Consider the discrete-time system (5.1) with

$$A_0 = \begin{bmatrix} 0.6 & 0.3 \\ 0.2 & 0.5 \end{bmatrix}, A_1 = \begin{bmatrix} 0.2 & 0.5 \\ 0.4 & 0.1 \end{bmatrix}, B_1 = \begin{bmatrix} 0.6 \\ 0.4 \end{bmatrix}, B_2 = \begin{bmatrix} 0.1 \\ -0.1 \end{bmatrix}, C = [1 - 0.8].$$

There are two input delays and one state delay present in the system. Notice that, although the matrices A_0 and A_1 are both Schur stable, the combined system is unstable due to delays, which can be checked by finding the roots of the polynomial matrix $P(\lambda) = A_0\lambda + A_1 - \lambda^2 I$ or by evaluating the eigenvalues of the augmented matrix A as defined in (5.5). It can be verified that the controllability condition of the combined delayed system $\rho \begin{bmatrix} \sum_{i=0}^{S} A_i \lambda^{S-i} - \lambda^{S+1} I & \sum_{0}^{T} B_i \lambda^{T-i} \end{bmatrix} = n$ holds. Hence, the extended augmented system (5.6) is controllable. The maximum input and state delays present in the system are $T = 2$ and $S = 1$, respectively. We first validate the state feedback Algorithm 1. Let $\bar{T} = 3$ and $\bar{S} = 1$ be the upper bounds on the input and state delays, respectively. We specify the user-defined performance index as $Q = I$ and $R = 1$. The nominal optimal feedback control matrix as obtained by solving the ARE with known delays is,

$$\bar{K}^* = \begin{bmatrix} 0.7991 & 0.8376 & 0.3622 & 0.3643 & 0.0007 & 0.6191 \end{bmatrix}.$$

The convergence criterion of $\varepsilon = 0.001$ was selected on the controller parameters. Since $l = n(\bar{S}+1) + m(\bar{T}+1) = 8$, we need at least $l(l+1)/2 = 36$ data samples to satisfy the rank condition (5.24) to solve (5.20). These data samples are collected by applying some sinusoidal signals of different frequencies and magnitudes in the control. The state response is shown in Fig. 5.1. It takes 9 iterations to converge to the optimal controller as shown in Fig. 5.2. It can be seen in Fig. 5.1 that the controller is able to stabilize the unstable system. The final estimate of the control matrix is,

$$\hat{\bar{K}} = \begin{bmatrix} 0.7996 & 0.8380 & 0.3625 & 0.3645 & 0 & 0.0007 & 0.6194 \end{bmatrix}.$$

It can be seen that the estimated control parameters correspond only to the actual delays present in the system while the one term corresponding to the extra state is 0. In other words, the final control is equal

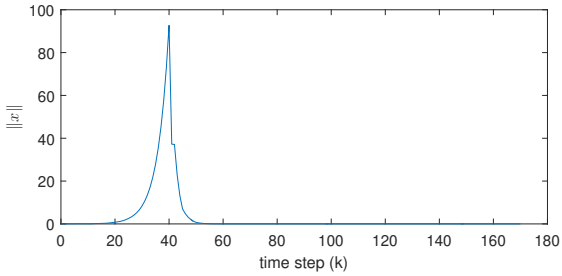

Fig. 5.1: Algorithm 1: State trajectory of the closed-loop system under state feedback.

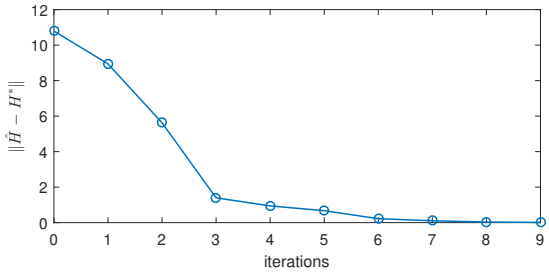

Fig. 5.2: Algorithm 1: Convergence of the parameter estimates under state feedback.

to the one obtained using the exact knowledge of the delays and system dynamics. Moreover, the rank condition (5.24) is no longer needed once the convergence criterion is met.

We next validate the output feedback Algorithm 2. It can be verified that the two observability conditions in Theorem 5.2 hold. We specify the user-defined performance index as $\bar{Q}_y = I$ and $R = 1$. The bound on the state delay is the same but the bound on the input delay has been increased to $\bar{T} = 4$ to make the observability index $N = \bar{T}$ in order to satisfy the output feedback rank condition (5.39). The nominal output feedback optimal control parameters as obtained by solving the ARE with known delays are,

$$\mathcal{H}^*_{u\bar{y}} = \begin{bmatrix} 12.1348 & -1.0313 & 2.1086 & 0 & 1.9331 & 0 & 1.7188 & 0 \end{bmatrix},$$
$$\mathcal{H}^*_{u\bar{u}} = \begin{bmatrix} 1.7316 & 2.1151 & -0.7301 & -2.0765 \end{bmatrix}, \mathcal{H}^*_{uu} = 3.3954.$$

The convergence criterion of $\varepsilon = 0.001$ was selected on the controller parameters. Since $l = mN + (p + m)N + m = 13$, we need at least $l(l + 1)/2 = 91$ data samples to satisfy the rank condition (5.39) to solve (5.35). These data samples are collected by applying some sinusoidal signals of different frequencies and magnitudes in the control. The state response is shown in Fig. 5.3. It takes around 18 iterations to converge to the optimal controller as shown in Fig. 5.4. It can be seen in Fig. 5.3 that

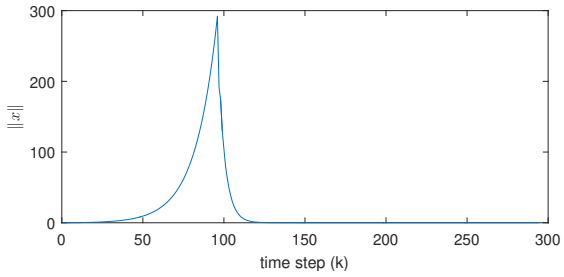

Fig. 5.3: Algorithm 2: State trajectory of the closed-loop system under output feedback.

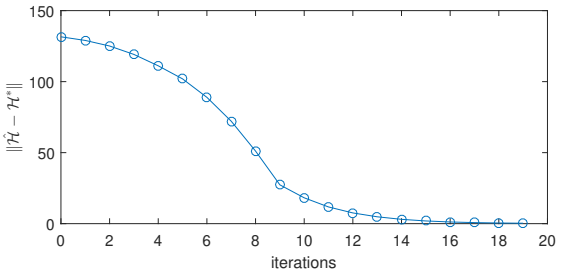

Fig. 5.4: Algorithm 2: Convergence of the parameter estimates under output feedback.

controller is able to stabilize the unstable system. The final estimates of the control parameters are,

$$\hat{\mathcal{H}}_{u\bar{y}} = \begin{bmatrix} 12.1077 & -1.0289 & 2.1040 & 0 & 1.9287 & 0 & 1.7149 & 0 \end{bmatrix},$$
$$\hat{\mathcal{H}}_{u\bar{u}} = \begin{bmatrix} 1.7277 & 2.1102 & -0.7286 & -2.0719 \end{bmatrix}, \hat{\mathcal{H}}_{uu} = 3.3927.$$

As can be seen in the state feedback and output feedback results, while the transient performance of the state feedback algorithm is superior due to fewer unknown parameters to be learnt, the output feedback algorithm has the advantage that it does not require the measurement of the full state feedback.

5.6. Summary

In this chapter, we presented an application of reinforcement learning to solve the model-free optimal stabilization problem for linear time delay systems with multiple unknown state and input delays and unknown system dynamics. To handle the unknown delays, an extended state augmentation approach was presented that transforms the system into an extended delay-free system. The upper bounds on the state and input delays were used in the augmentation process, in which the delayed state and control inputs were treated as additional virtual states of the systems. The transformation of the system to the extended delay-free form does not require any bicausal change of coordinates, as needed in the previous works. Furthermore, it was shown that the controllability property of the system is also preserved in the extended augmentation process. State feedback Q-learning scheme was employed to learn the optimal control parameters. In the second part of this chapter, we focused on the design of the output feedback Q-learning controller in order to circumvent the requirement of full state for feedback. Observability conditions of the augmented and extended augmented systems were presented. Unlike the controllability result, it was found that while the augmented delay-free system (with known maximum state delay and input delay) can recover observability, the extended augmented systems (with unknown maximum state delay and input delay) can achieve at most detectability. Under the special case when the maximum state delay is known, the extended augmented system becomes observable even when the maximum input delay is unknown. An output feedback Q-function was presented based on the parametrization of the state vector using delayed input and output measurements. An iterative Q-learning scheme was presented to learn the optimal output feedback Q-function using only the measured input and output data. Simulation results were used to demonstrate the effectiveness of both the state feedback and the output feedback algorithms.

6. Conclusions and Future Work

This dissertation presents new reinforcement learning algorithms for the model-free optimal control of linear systems using dynamic output feedback. In contrast to their state feedback counterparts, these output feedback algorithms require fewer sensors, and thereby, make the RL control framework more feasible from a practical stand point. The primary challenges in developing output feedback RL algorithms include the issue of exploration bias, the need of discounted cost functions and the unavailability of models to design state estimators. Although exploration is essential in RL algorithms to ensure parameter convergence, it is also linked to the estimation bias problem. Discounted cost functions provide a way to suppress this problem but they also bring instabilities in the design. Furthermore, state estimation based control techniques are difficult to design together with RL control algorithms. To tackle these difficulties, this work develops new learning equations based on dynamic output feedback. These new learning equations incorporate the exploration component of the control signal, and thereby, prevent the estimation bias from occurring in the first place. As a result, the need of discounted cost functions is eliminated and closed-stability and optimality of the solution are ensured. The output feedback control parameters are learned directly without ever estimating the internal state, which would otherwise complicate the design. These ideas are materialized first in the form of model-free algorithms for the classical linear quadratic regulator optimal control problem for both discrete-time and continuous-time systems.

This work has also developed new algorithms to address more advanced control problems. H-infinity based dynamic output feedback control algorithms are developed to reject external disturbances, where the ideas of game theory are employed to solve these problems. The dissertation also focuses on solving control constrained problems. Ideas of low gain feedback and parameterized Riccati equations are presented in a model-free setting to solve these problems. Finally, attention is also given to handle time delays present in the control loop, and for this purpose, an extended state augmentation approach is presented to solve the optimal control problems in the presence of delays whose information is not known.

This work, while addressing some RL control challenges, also gives rise to some new questions and interesting problems for future work, such as,

1) Solving general optimal tracking problems in the presence of disturbances.
2) Extensions to multi-agent distributed reinforcement learning control.

3) Developing disturbance rejection RL algorithms that do not require access to the disturbance channel.
4) Relaxing the exploration requirements for safe learning practices.
5) Solving continuous-time delay problems using reinforcement learning.
6) Extensions of the work to solve nonlinear output feedback control problems.

REFERENCES

[1] W. Aangenent, D. Kostic, B. de Jager, R. van de Molengraft, and M. Steinbuch, "Data-based optimal control," in *Proceedings of the 2005 American Control Conference*, 2005, pp. 1460–1465.

[2] M. I. Abouheaf, F. L. Lewis, K. G. Vamvoudakis, S. Haesaert, and R. Babuska, "Multi-agent discrete-time graphical games and reinforcement learning solutions," *Automatica*, vol. 50, no. 12, pp. 3038–3053, 2014.

[3] M. Abu-Khalaf and F. L. Lewis, "Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach," *Automatica*, vol. 41, no. 5, pp. 779–791, 2005.

[4] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control," *Automatica*, vol. 43, no. 3, pp. 473–481, 2007.

[5] A. Al-Tamimi, F. L. Lewis, and Y. Wang, "Model-free H-infinity load-frequency controller design for power systems," in *Proceedings of the 22nd International Symposium on Intelligent Control*, 2007, pp. 118–125.

[6] E. Alpaydin, *Introduction to machine learning*. MIT press, 2009.

[7] T. Bacsar and P. Bernhard, *H-infinity Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach*. Springer Science & Business Media, 2005.

[8] M. Bardi and I. Capuzzo-Dolcetta, *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*. Springer Science & Business Media, 2008.

[9] G. Barles and E. R. Jakobsen, "On the convergence rate of approximation schemes for Hamilton-Jacobi-Bellman equations," *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 36, no. 1, pp. 33–54, 2002.

[10] R. W. Beard, G. N. Saridis, and J. T. Wen, "Galerkin approximations of the generalized Hamilton-Jacobi-Bellman equation," *Automatica*, vol. 33, no. 12, pp. 2159–2177, 1997.

[11] R. W. Beard, G. N. Saridis, and J. T. Wen, "Approximate solutions to the time-invariant Hamilton–Jacobi–Bellman equation," *Journal of Optimization theory and Applications*, vol. 96, no. 3, pp. 589–626, 1998.

[12] R. E. Bellman, *Dynamic Programming*. Princeton University Press, 1957.

[13] T. Bian and Z.-P. Jiang, "Data-driven robust optimal control design for uncertain cascaded systems using value iteration," in *Proceedings of the 54th Annual Conference on Decision and Control (CDC)*, 2015, pp. 7610–7615.

[14] T. Bian and Z.-P. Jiang, "Value iteration and adaptive dynamic programming for data-driven adaptive optimal control design," *Automatica*, vol. 71, pp. 348–360, 2016.

[15] V. Boltyanskii, R. Gamkrelidze, and L. Pontryagin, "On the theory of optimal processes," *Sci. USSR*, vol. 110, no. 1, pp. 71–0, 1956.

[16] S. J. Bradtke, B. E. Ydstie, and A. G. Barto, "Adaptive linear quadratic control using policy iteration," in *Proceedings of the 1994 American Control Conference*, 1994, pp. 3475–3479.

[17] L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Transactions on Systems, Man, And Cybernetics-Part C: Applications and Reviews*, vol. 38, no. 2, 2008.

[18] B. M. Chen, *Robust and H-infinity Control*. Springer Science & Business Media, 2013.

[19] L. Dong, X. Zhong, C. Sun, and H. He, "Adaptive event-triggered control based on heuristic dynamic programming for nonlinear discrete-time systems," *IEEE Transactions on Neural Networks and Learning Systems*, to appear.

[20] B. Frapard and C. Champetier, "H-infinity techniques - from research to industrial applications," in *Proceedings of the 3rd ESA International Conference on Spacecraft Guidance, Navigation and Control Systems*, 1997, pp. 231–237.

[21] W. Gao and Z. P. Jiang, "Data-driven adaptive optimal output-feedback control of a 2-DOF helicopter," in *Proceedings of the 2016 American Control Conference*, 2016, pp. 2512–2517.

[22] K. Gu, J. Chen, and V. L. Kharitonov, *Stability of Time-Delay Systems*. Springer Science & Business Media, 2003.

[23] P. He and S. Jagannathan, "Reinforcement learning-based output feedback control of nonlinear systems with input constraints," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 1, pp. 150–154, 2005.

[24] G. Hewer, "An iterative technique for the computation of the steady state gains for the discrete optimal regulator," *IEEE Transactions on Automatic Control*, vol. 16, no. 4, pp. 382–384, 1971.

[25] R. Howard, *Dynamic Programming and Markov Processes*. MIT Press: Cambridge, MA, USA, 1960.

[26] R. A. Hyde, K. Glover, and G. T. Shanks, "VSTOL first flight on an h-infinity control law," *Computing Control Engineering Journal*, vol. 6, no. 1, pp. 11–16, Feb 1995.

[27] Y. Jiang and Z.-P. Jiang, "Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics," *Automatica*, vol. 48, no. 10, pp. 2699–2704, 2012.

[28] B. Kiumarsi and F. L. Lewis, "Actor-critic-based optimal tracking for partially unknown nonlinear discrete-time systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 1, pp. 140–151, 2015.

[29] B. Kiumarsi and F. L. Lewis, "Output synchronization of heterogeneous discrete-time systems: A model-free optimal approach," *Automatica*, vol. 84, pp. 86–94, 2017.

[30] B. Kiumarsi, F. L. Lewis, and Z.-P. Jiang, "H_∞ control of linear discrete-time systems: Off-policy reinforcement learning," *Automatica*, vol. 78, pp. 144–152, 2017.

[31] B. Kiumarsi, F. L. Lewis, H. Modares, A. Karimpour, and M.-B. Naghibi-Sistani, "Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics," *Automatica*, vol. 50, no. 4, pp. 1167–1175, 2014.

[32] B. Kiumarsi, F. L. Lewis, M.-B. Naghibi-Sistani, and A. Karimpour, "Optimal tracking control of unknown discrete-time linear systems using input-output measured data," *IEEE Transactions on Cybernetics*, vol. 45, no. 12, pp. 2770–2779, 2015.

[33] D. Kleinman, "On an iterative technique for Riccati equation computations," *IEEE Transactions on Automatic Control*, vol. 13, no. 1, pp. 114–115, 1968.

[34] P. Lancaster and L. Rodman, *Algebraic Riccati Equations*. Clarendon press, 1995.

[35] T. Landelius, "Reinforcement learning and distributed local model synthesis," Ph.D. dissertation, Linköping University Electronic Press, 1997.

[36] F. L. Lewis and D. Liu, *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*. John Wiley & Sons, 2013, vol. 17.

[37] F. L. Lewis and V. L. Syrmos, *Optimal Control*. John Wiley & Sons, 1995.

[38] F. L. Lewis and K. G. Vamvoudakis, "Reinforcement learning for partially observable dynamic processes: Adaptive dynamic programming using measured output data," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 1, pp. 14–25, 2011.

[39] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits and Systems Magazine*, vol. 9, no. 3, pp. 32–50, 2009.

[40] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, "Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers," *IEEE Control Systems Magazine*, vol. 32, no. 6, pp. 76–105, 2012.

[41] H. Li, D. Liu, D. Wang, and X. Yang, "Integral reinforcement learning for linear continuous-time zero-sum games with completely unknown dynamics." *IEEE Trans. Automation Science and Engineering*, vol. 11, no. 3, pp. 706–714, 2014.

[42] Z. Lin, *Low Gain Feedback*. Springer, 1999.

[43] D. Liu, Y. Huang, D. Wang, and Q. Wei, "Neural-network-observer-based optimal control for unknown nonlinear systems using adaptive dynamic programming," *International Journal of Control*, vol. 86, no. 9, pp. 1554–1566, 2013.

[44] S. Lyashevskiy, "Control of linear dynamic systems with constraints: optimization issues and applications of nonquadratic functionals," in *Proceedings of the 35th IEEE Conference on Decision and Control, 1996*, vol. 3. IEEE, 1996, pp. 3206–3211.

[45] S. E. Lyshevski, "Optimal control of nonlinear continuous-time systems: design of bounded controllers via generalized nonquadratic functionals," in *Proceedings of the 1998 American Control Conference*, 1998, pp. 205–209.

[46] R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, *Machine Learning: An Artificial Intelligence Approach*. Springer Science & Business Media, 2013.

[47] H. Modares and F. L. Lewis, "Linear quadratic tracking control of partially-unknown continuous-time systems using reinforcement learning," *IEEE Transactions on Automatic control*, vol. 59, no. 11, pp. 3051–3056, 2014.

[48] H. Modares and F. L. Lewis, "Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning," *Automatica*, vol. 50, no. 7, pp. 1780–1792, 2014.

[49] H. Modares, F. L. Lewis, and Z. P. Jiang, "Optimal output-feedback control of unknown continuous-time linear systems using off-policy reinforcement learning," *IEEE Transactions on Cybernetics*, vol. 46, no. 11, pp. 2401–2410, 2016.

[50] H. Modares, F. L. Lewis, and M.-B. Naghibi-Sistani, "Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems," *Automatica*, vol. 50, no. 1, pp. 193–202, 2014.

[51] H. Modares, S. P. Nageshrao, G. A. D. Lopes, R. Babuška, and F. L. Lewis, "Optimal model-free output synchronization of heterogeneous systems using off-policy reinforcement learning," *Automatica*, vol. 71, pp. 334–341, 2016.

[52] C. Mu, D. Wang, and H. He, "Novel iterative neural dynamic programming for data-based approximate optimal control design," *Automatica*, vol. 81, pp. 240–252, 2017.

[53] I. Postlethwaite, A. Smerlas, D. Walker, A. Gubbels, S. Baillie, M. Strange, and J. Howitt, "H-infinity control of the nrc bell 205 fly-by-wire helicopter," *Journal of the American Helicopter Society*, vol. 44, no. 4, pp. 276–284, 1999.

[54] R. Postoyan, L. Busoniu, D. Nesic, and J. Daafouz, "Stability analysis of discrete-time infinite-horizon optimal control with discounted cost," *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 2736–2749, 2017.

[55] Y. Shoham, R. Powers, and T. Grenager, "Multi-agent reinforcement learning: a critical survey," Technical report, Stanford University, Tech. Rep., 2003.

[56] E. D. Sontag and H. J. Sussmann, "Nonlinear output feedback design for linear systems with saturating controls," in *Proceedings of the 29th IEEE Conference on Decision and Control*, 1990, pp. 3414–3416.

[57] G. Strub, S. Theodoulis, V. Gassmann, S. Dobre, and M. Basset, "Pitch axis control for a guided projectile in a wind tunnel hardware-in-the-loop setup," *Journal of Spacecraft and Rockets*, vol. 52, no. 6, pp. 1614–1626.

[58] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT press Cambridge, 1998.

[59] R. S. Sutton, A. G. Barto, and R. J. Williams, "Reinforcement learning is direct adaptive optimal control," *IEEE Control Systems*, vol. 12, no. 2, pp. 19–22, 1992.

[60] G. Tao, *Adaptive Control Design and Analysis*. John Wiley & Sons, 2003.

[61] K. G. Vamvoudakis, F. L. Lewis, and G. R. Hudas, "Multi-agent differential graphical games: Online adaptive learning solution for synchronization with optimality," *Automatica*, vol. 48, no. 8, pp. 1598–1611, 2012.

[62] D. Vrabie, "Online adaptive optimal control for continuous-time systems," Ph.D. dissertation, The University of Texas at Arlington, 2009.

[63] D. Vrabie and F. Lewis, "Adaptive dynamic programming for online solution of a zero-sum differential game," *Journal of Control Theory and Applications*, vol. 9, no. 3, pp. 353–360, 2011.

[64] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. L. Lewis, "Adaptive optimal control for continuous-time linear systems based on policy iteration," *Automatica*, vol. 45, no. 2, pp. 477–484, 2009.

[65] L. Y. Wang, C. Li, G. G. Yin, L. Guo, and C.-Z. Xu, "State observability and observers of linear-time-invariant systems under irregular sampling and sensor limitations," *IEEE Transactions on Automatic Control*, vol. 56, no. 11, pp. 2639–2654, 2011.

[66] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. dissertation, King's College, Cambridge, 1989.

[67] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[68] P. Werbos, "Beyond regression:" new tools for prediction and analysis in the behavioral sciences," Ph.D. dissertation, Harvard University, 1974.

[69] P. Werbos, "Approximate dynamic programming for realtime control and neural modelling," *Handbook of intelligent control: neural, fuzzy and adaptive approaches*, pp. 493–525, 1992.

[70] P. J. Werbos, "Neural networks for control and system identification," in *Proceedings of the 28th IEEE Conference on Decision and Control*, 1989, pp. 260–265.

[71] P. J. Werbos, "A menu of designs for reinforcement learning over time," *Neural networks for control*, pp. 67–95, 1990.

[72] H. Zhang, L. Cui, X. Zhang, and Y. Luo, "Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 2226–2236, 2011.

[73] H. Zhang, Y. Liu, G. Xiao, and H. Jiang, "Data-based adaptive dynamic programming for a class of discrete-time systems with multiple delays," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, no. 99, pp. 1–10, 2017.

[74] J. Zhang, H. Zhang, Y. Luo, and T. Feng, "Model-free optimal control design for a class of linear discrete-time systems with multiple delays using adaptive dynamic programming," *Neurocomputing*, vol. 135, pp. 163–170, 2014.

[75] Q. Zhao, H. Xu, and S. Jagannathan, "Near optimal output feedback control of nonlinear discrete-time systems based on reinforcement neural network learning," *IEEE/CAA Journal of Automatica Sinica*, vol. 1, no. 4, pp. 372–384, 2014.

[76] X. Zhong and H. He, "An event-triggered ADP control approach for continuous-time system with unknown internal states," *IEEE Transactions on Cybernetics*, vol. 47, no. 3, pp. 683–694, 2017.

[77] L. M. Zhu, H. Modares, G. O. Peen, F. L. Lewis, and B. Yue, "Adaptive suboptimal output-feedback control for linear systems using integral reinforcement learning," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 264–273, 2015.