Ben-Gurion University of the Negev

The Faculty of Engineering

The Department of Mechanical Engineering

Model Free Output-Feedback Control with applications in Mixed Platoons

Thesis submitted in partial fulfillment of the requirements

for the Master of Sciences degree

By Eliran Elbaz

Under the supervision of Prof. Shai Arogeti

March 2025

Ben-Gurion University of the Negev

The Faculty of Engineering

The Department of Mechanical Engineering

Model Free Output-Feedback Control with applications in Mixed Platoons

Thesis submitted in partial fulfillment of the requirements

for the Master of Sciences degree

Under the supervision of Prof. Shai Arogeti

Signature of student: _____ Date: _____

Signature of supervisor: _____ Date: _____

Signature of chairperson of the

committee for graduate studies: _____ Date: _____

March 2025

Model Free Output-Feedback Control with applications in Mixed Platoons

Eliran Elbaz

Under the supervision of Prof. Shai Arogeti

Ben-Gurion University of the Negev 2025

Abstract

This study focuses on developing and improving control methods for mixed platoons consisting of autonomous and human-driven vehicles. The primary goal is to guarantee weak string stability and dynamic performance, even under uncertain conditions and partial system knowledge. A novel output-feedback control framework for mixed platoons is proposed based on Adaptive Dynamic Programming (ADP), enabling optimal control policies to be learned directly from input-output data without requiring complete knowledge of the system dynamics. The algorithm is further improved by significantly reducing the number of controller parameters involved in the learning process, reducing the computational complexity, and facilitating implementation in large-scale systems such as long-vehicle platoons.

A key strength of the proposed framework lies in its ability to guarantee desired string stability behavior, even in the presence of uncontrollable human-driven vehicles within the platoon. This structural property is achieved by formulating a $\mathcal{H}_\infty$ control problem, which explicitly incorporates weak-string stability into the design objectives. The simulation results demonstrate the algorithm's ability to learn stable and effective control policies while obtaining the required mixed platoon performance. In particular, we demonstrate the controller's ability to compensate for the string instability of human drivers involved in the mixed platoon. This research contributes to the development of intelligent and safe transportation systems, particularly in heterogeneous traffic environments. In addition, the proposed framework supports the reduction of congestion propagation along the platoon, thereby improving overall traffic stability and efficiency.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Vehicle Platooning

The California Program on Advanced Technology for the Highway (PATH) pioneered the concept of vehicle platooning, aiming to enhance road safety, efficiency, and fuel consumption through coordinated, automated vehicle control systems. According to [5], vehicle platooning involves multiple vehicles traveling closely together, managed by advanced control systems that regulate spacing and speed, ensuring tight formations without compromising safety. However, transitioning to fully connected autonomous vehicles (CAV) could take decades [6]. CAVs will coexist with human-driven vehicles (HDVs) during this transition period. Even today, cars are equipped with systems that support a driver while performing a specific driving task with safety, convenience, efficiency, and an improvement in the overall driver experience.

A key concept of traffic dynamics is string stability, which is defined for a platoon of vehicles along the longitudinal axes. It investigates the behavior of the entire platoon in response to disturbances, e.g., a sudden break of the platoon leader. The car-follow behavior is called string stable if the perturbation decays as it propagates upstream within the platoon [7]. Vehicle platooning, especially in mixed platoons involving autonomous and human-driven vehicles, represents a significant step towards a future with fully autonomous transportation.

In this context, an additional concept introduced in [8] and called *weak string stability* becomes relevant. In mixed platoons, where not all vehicles are controllable (e.g., HDVs), disturbances

often originate from the leader or the first CAV in the platoon. These disturbances are then amplified by the unstable and unpredictable behavior of the HDVs in the middle of the platoon. However, they can still be attenuated by the last CAV. *Weak string stability* refers to this partial attenuation of disturbances, where the overall system does not guarantee full decay throughout, but ensures that perturbations are eventually mitigated by the autonomous tail.

Vehicle platooning, especially in mixed platoons involving autonomous and human-driven vehicles, represents a significant step towards a future with fully autonomous transportation. Although there are challenges in achieving weak string stability in the mixed platoon, the benefits of enhanced safety, improved traffic flow, and environmental advantages make it a promising area for ongoing research and development. In [9], linear quadratic regulation (LQR) is used to obtain an optimal design of connected cruise control (CCC) for a mixed platoon. However, they assumed the connectivity of human cars, and the solution was model-based. In [10], the authors used Adaptive Dynamic Programming (ADP) techniques and Reinforcement Learning (RL) to solve the LQR problem of continuous-time systems and design adaptive cruise control (ACC) without knowing the system model. Still, the Policy Iteration (PI) algorithm used was full-state feedback based on the assumption that human cars are connected and all system state variables are known. Vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication technologies are still being developed. Most vehicles available to the general public are not connected, including those equipped with advanced driver assistance systems (ADAS), which rely primarily on sensors and radars onboard rather than communication between vehicles. In the study presented here, the challenge due to the unconnected HDV in a mixed platoon is addressed, assuming that all HDVs in the platoon are not communicating with the CAVs. From the control point of view, the meaning of this assumption is that not all of the platoon state variables are measurable, which does not allow the use of state-feedback controllers.

In [11], they propose a model-free solution to the LQR problem of continuous-time systems based on reinforcement learning using dynamic output feedback. The advantage of their method is that knowing the system states is unnecessary. Then, in [12], they extended their method to the linear zero-sum game (ZSG), which is the basis of a $\mathcal{H}_\infty$ controller. They demonstrated their algorithms for systems such as the F16 autopilot and a double integrator (which is open-loop unstable). The motivation of our study is to modify this dynamic output feedback algorithm

and implement the modified variant in the mixed platoon control problem. That means that two notable advantages are achieved:

- As mentioned before, we assume the HDVs are not connected, and therefore, it is impossible to design a controller that depends on all system state variables and uses the information from these vehicles. This assumption motivates the use of output-feedback strategies.
- In the linear ZSG-based optimal control, the weak string stability criterion can be embedded in the standard design criterion (this concept is explained in chapter 4).

During the implementation of the model-free dynamic output feedback algorithm, it was observed that increasing the number of output measurements significantly raises the number of unknowns, thus increasing computational complexity. To address this challenge, we developed a technique that unifies the output measurements into a linear combination of the measured system states. This transformation preserves the integrity of the algorithm result while reducing the number of variables involved in the computation. The proposed improvement is general and applicable to both the LQR and the linear ZSG (i.e., $\mathcal{H}_\infty$) control designs. In this study, we utilize the suggested improved design strategy for the control of a mixed platoon system.

1.2 Literature Review

First, we will review the development made in the field of longitudinal control in the automotive industry, starting with ADAS that are already integrated into vehicles today, particularly ACC. Later, we will review the concept of cooperative adaptive cruise control (CACC). Subsequently, we will present studies made in longitudinal control on automated platoons that consist of self-driving cars and their communication protocols. Then, the notion of mixed platoons will be covered and explained. Next, a historical overview of ADP development will be provided, including basic principles, key methods, and applications. Finally, the challenges and limitations of ADP will be discussed.

1.2.1 Mixed Traffic

ADAS supports a driver while performing a specific driving task with safety, convenience, efficiency, and an improvement in the overall driver experience [13]. Table 1.1 shows a list developed by the National Highway Traffic Safety Administration (NHTSA) for five eras of safety through the evolution of automated safety technologies [14].

Evolution Era	Automated Technologies
1950-2000	Safety convenience features Cruise control Seat belts Antilock brakes
2000-2010	Advanced safety features Electronic Stability Control Blind Spot Detection Forward Collision Warning Lane Departure Warning
2010-2016	Advanced driver assistance features Rear view Video Systems Automatic Emergency Braking Pedestrian Automatic Emergency Braking Rear Automatic Emergency Braking Rear Cross-Traffic Alert Lane Centering Assist
2016-2025	Partially automated safety features Lane-keeping assist Adaptive cruise control Traffic jam assist Self-park
2025+	Fully automated safety features Highway autopilot

Table 1.1: Available assistive driving.

The main focus is on Adaptive Cruise Control (ACC), which is widely implemented in modern vehicles and has become a standard component of advanced driver assistance systems. Due to its prevalence, ACC has a substantial impact on mixed traffic scenarios, particularly with respect to stop-and-go phenomena. These phenomena refer to traffic patterns characterized by alternating acceleration and deceleration phases, often without a clear external trigger. Such behavior can induce traffic instabilities, especially when disturbances propagate as waves through the vehicle string. These waves, sometimes referred to as traffic shockwaves, can amplify along the platoon and degrade both safety and road capacity. Understanding and mitigating these effects is,

therefore, essential when evaluating the performance of ACC-equipped vehicles in mixed traffic environments.

ACC systems use radar and cameras to monitor the vehicle ahead and automatically adjust the speed to maintain a safe following distance. If there is no vehicle ahead, the system works as a regular cruise control and maintains a fixed speed value set by the driver. In [15], the study assesses the string stability of seven 2018 model year ACC-equipped vehicles widely available on the US market. Seven different vehicle models are analyzed from two different vehicle makers using data collected from more than 1,200 miles of driving. Delay differential equation models of the vehicle under ACC control were calibrated with the following minimum and maximum settings. All vehicles with all the following settings were found to be string-instable.

1.2.2 Automated Platoons

To begin, it is important to clarify the distinction between two closely related yet often confused concepts that are central to this section: *connected vehicle (CV)* and *automated vehicle (AV)*. According to the definitions provided in [16], these terms refer to different technologies with different functionalities and implications.

CV options include:

- Vehicle-to-vehicle (V2V).
- Vehicle to infrastructure (V2I).
- Infrastructure to vehicle (I2V).
- Vehicle to pedestrian (V2P).
- Vehicle to anything (V2X).

V2V connectivity can be realized through WiFi, 4G/5G cellular networks, Bluetooth, etc. It is utilized to share data such as the speed of the vehicle in front, sudden deceleration of a far-ahead vehicle, etc. V2I and I2V can broadcast information about weather, future roadworks, hazards, etc. V2P includes any vulnerable road user who has a device that can connect to nearby vehicles. V2X represents connectivity to everything, in which virtually every device

could be connected to any other device. Our interest is in the V2V / V2I connectivity that enables CACC. The integrated connectivity allows for tighter vehicle follow-up control than the conventional ACC, with improved traffic flow stability.
AVs, on the other hand, are vehicles with the ability to perform an automatic task instead of a human driver, for example, maintaining a constant speed through a cruise control.

There is a strong synergy between connected and automated vehicles. Although these technologies have developed along separate paths, they offer complementary advantages in improving traffic safety and efficiency. Connectivity enables real-time data sharing between vehicles and infrastructure, while automation addresses human driver limitations. As noted in [16], the integration of CV and AV technologies allows enhanced perception, coordination, and control beyond what standalone AV systems can achieve.

For the design framework in this study, the concepts of AV and CV are combined into a connected autonomous vehicle (CAV).
In [17] the authors have used wireless communication (V2V) to add additional information about the leader of the platoon to a standard ACC system, thus being able to design a controller that reduces the effects of speed disturbance along the platoon, and managed to achieve string stability considering reasonable acceleration and deceleration. The advantages of V2V communication have been widely studied in the literature. In [18], the authors examined the effect of CACC on traffic throughput and showed that it increased the capacity of the highway near the lane drop (where the lane drop simulates a speed disturbance propagating through the platoon). The CACC system can improve traffic flow performance; however, this is only when 40% or more vehicles are equipped with it. In [1], the effect on traffic throughput of ACC systems compared to CACC systems and to Here I Am (HIA) systems was examined. HIA vehicles are driven manually, but are equipped with a dedicated short-range communications radio that frequently broadcasts a 'here I am' message with its location and speed. Vehicle Awareness Devices (VADs) can be used by a platoon leader or other CACC vehicles. It can be seen in Figure 1.1 that the lane capacity is a function of the market penetration of the CACC systems. In addition, the increase in road capacity is further improved when HIA devices are added.

Figure 1.2 shows the prediction of the lane capacity for the integration of ACC and CACC

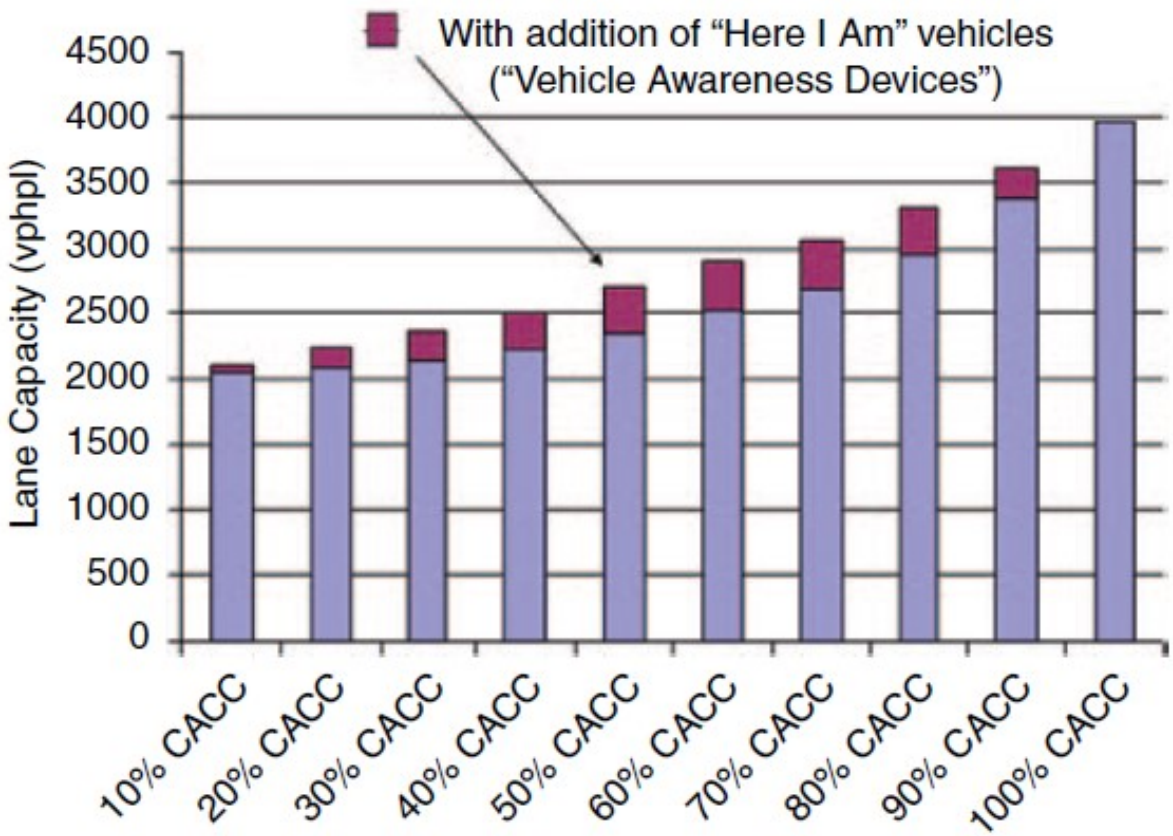

Figure 1.1: Highway lane capacity as a function of changes in CACC market penetration relative to manually driven vehicles or HIA vehicles, from [1].

vehicles into standard traffic. The figures show that with a penetration rate of 60% and a higher number of CACC systems, there is a significant improvement in the lane capacity.

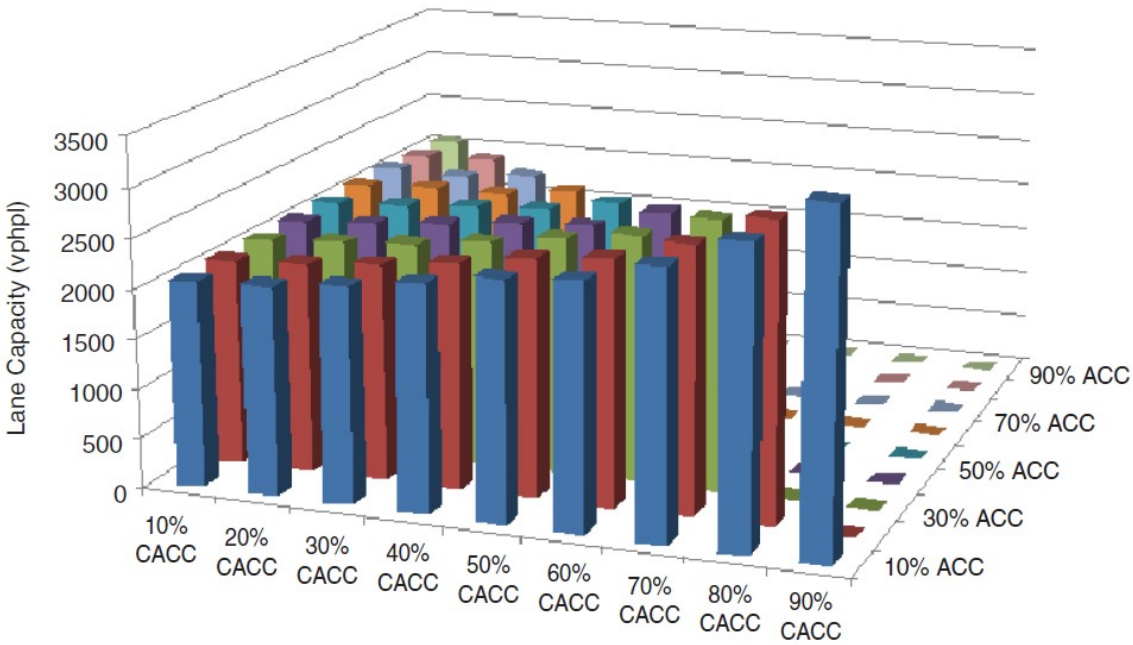

Figure 1.2: Updated prediction of lane capacity effects of ACC and CACC vehicles driven at time gaps chosen by drivers in field test (remaining vehicles manually driven), from [1].

1.2.3 Mixed Platoons

As demonstrated in the previous section, CACC enhances ACC's functionality by utilizing V2V communication. This enables vehicles to exchange real-time information on speed, acceleration, and braking behavior.

In the near future, HDVs, which lack connectivity capabilities, are expected to drive along with CAVs. Numerous studies have addressed the problem of longitudinal control in such mixed-traffic scenarios. These studies can generally be categorized into two main approaches, depending on how HDVs are treated within the control framework:

1. Manually driven vehicles have a means of communication.

2. Manually driven vehicles are without means of communication.

A representative study of the first approach is presented in [9], where the authors model a mixed platoon in state space form by linearizing the Intelligent Driver Model (IDM) to describe the behavior of HDVs. They formulate an LQR problem to derive the optimal control gains for

the CAVs and employ reinforcement learning and adaptive dynamic programming techniques to solve the Riccati equation without explicit knowledge of system dynamics. However, their method assumes the complete observability of the system, including access to all state variables such as HDV speed and distance from the previous vehicle.

The second approach is more relevant to our work, as discussed in the previous chapter, since it is reasonable to assume that real-time state information about HDVs is not available. In [19], the authors designed a model predictive control (MPC) framework based on real-time trajectory data to regulate the motion of CAVs within a mixed platoon, thus improving traffic stability and smoothness. In [3], mixed platoons and human driver behavior modeling are further refined. The authors utilize a $\mathcal{H}_\infty$ control framework to address model uncertainties and propose four types of controllers, ranging from a robust baseline controller to a fully integrated one.

1.2.4 Adaptive Dynamic Programming

The concept of learning-based control has its roots in early research on artificial intelligence, dating back to Minsky's Ph.D. dissertation [20]. In this seminal work, Minsky introduced the foundation of reinforcement learning (RL), inspired by the challenge of understanding learning, memory, and cognitive processes in the human brain. In RL, an agent interacts with an unknown environment by taking actions and receiving feedback as a reward. The agent's objective is to maximize its reward, which, in the context of optimal control, is typically framed as minimizing a cost function [21]. Reinforcement learning (RL) does not rely on a supervisor to instruct the agent in selecting the optimal action. Instead, it emphasizes how the agent, through interactions with an unknown environment, can adjust its actions progressively toward optimal behavior. As shown in Figure 1.3, a typical RL iteration consists of two key steps. First, the agent interacts with the environment to assess the cost associated with its current policy, a process known as policy evaluation. Then, using the evaluated cost, the agent updates its policy to further minimize the cost, a step known as policy improvement [2].

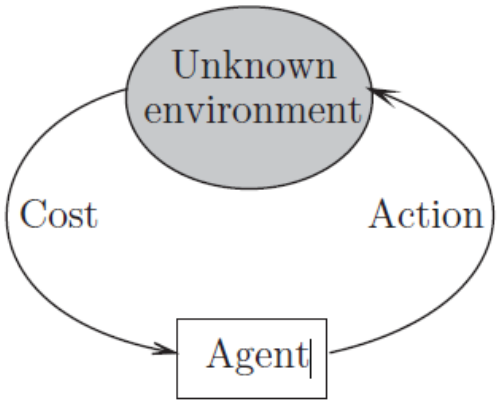

Figure 1.3: An illustration of RL: The agent selects an action to interact with an unknown environment and assesses the resulting cost. Based on this evaluation, the agent refines its actions to minimize the cost, from [2].

The field of reinforcement learning (RL) continues to grow and evolve, with numerous review articles and books highlighting its ongoing advances and untapped potential [22–25]. Dynamic programming (DP) [26], a well-known concept in optimal control, provides a theoretical framework to solve multistage decision-making problems. However, its practical application is often limited by significant computational complexity, commonly referred to as the curse of dimensionality.

The Bellman principle of dynamic programming is a cornerstone of optimal control theory, where the Hamilton–Jacobi–Bellman (HJB) equation plays a crucial role in solving optimal control problems. In most cases, however, the HJB is numerically intractable except in specific cases such as linear systems, where it reduces to solving an algebraic Riccati equation (ARE) [27]. Even then, the ARE remains non-linear, and solving it typically requires an offline procedure that assumes full model knowledge.

Among the main solution approaches to dynamic programming are policy iteration (PI) and value iteration (VI). Although both methods aim to compute optimal policies, this thesis focuses solely on PI-based algorithms, which are better suited for safety-critical systems such as mixed vehicular platoons. Consequently, VI is not addressed further in this work.

To address the challenges of solving the ARE in a model-free context, model-based iterative methods such as PI have been proposed [28]. These approaches rely on solving the Lyapunov equations and require that the closed-loop system is initialized with a stabilizing control policy. When this condition is satisfied, the associated cost is finite, and the algorithm constitutes the unique stabilizing solution. As such, PI-based algorithms that require an initial stabilizing controller are well-suited for control tasks with safety-critical constraints (such as control of mixed platoons).

Policy iteration (PI) algorithms improve a control policy through two alternating steps: **Policy Evaluation** and **Policy Improvement**. These steps are repeated iteratively until the policy converges. This thesis adopts the PI framework due to its suitability for safety-critical systems such as mixed vehicular platoons, as it requires a stabilizing controller for the complete learning process. The two main steps in the PI algorithm are

Model-free reinforcement learning (RL) and adaptive dynamic programming (ADP) methods have been developed to design the LQR controller for discrete-time systems [29]. Most of the progress in the field was made in discrete time because the continuous Bellman equation is based on the Hamiltonian, which contains the system information. Therefore, using RL, [22] suggested a design with only partial knowledge of the model. Later, [30] solved the problem, but it was still necessary to obtain partial knowledge of the system dynamics to be accurately known in the context of continuous-time systems. In [31], on the other hand, the authors introduced a policy iteration (PI) approach for continuous-time linear systems with entirely unknown

system dynamics. A value-based approach was proposed in [32] for an optimal adaptive design of continuous-time linear systems with unknown dynamics. In [33][34], Robust Adaptive Dynamic Programming (RADP) of the two-player zero-sum game (ZSG) for continuous-time linear systems has been developed. **Still, all of these algorithms are full-state feedback and require knowledge of all system variables**.

In [35], the authors suggested discrete-time policy and value iteration algorithms that converge to an optimal controller that requires only output feedback (OPFB). These algorithms need exploration noise to be added to the evaluated policy and suffer from bias due to that additional noise. The authors solved this problem by adding a discount factor that minimizes interference to a negligible effect [35]. In [36], it has been found that an incorrect choice of the discount factor can destabilize the system, so an upper limit must be defined. In addition, these solutions lead to sub-optimality. In [37][38], the authors solved the bias problem by presenting an RL-based dynamic output feedback algorithm for discrete and continuous time for the LQR and $\mathcal{H}_\infty$ problems. Later, the same authors extended their development to solve also the continuous-time LQR and $\mathcal{H}_\infty$ control design problems [11][12].

Although numerous algorithms have been proposed for continuous-time optimal control using the policy iteration (PI) and value iteration (VI) frameworks, they commonly rely on the assumption of full-state feedback. In contrast, this thesis focuses on extending PI-based methods to output-feedback scenarios, as detailed in subsequent chapters.

1.2.5 Thesis Scope and Contribution

This thesis focuses on the development of a model-free output feedback control strategy for mixed platoons consisting of human-driven vehicles (HDVs) and connected autonomous vehicles (CAVs). The main objectives of this research are as follows.

1. To model a mixed platoon that incorporates both HDVs and CAVs, while capturing the interactions and dynamics unique to heterogeneous traffic flow. Most importantly, formulating the mixed platoon model so that the string-stability criterion can be embedded in the optimal control design criterion.

2. Design an optimal controller for cooperative adaptive cruise control (CACC) without requiring prior knowledge of human driving model coefficients (i.e., f_s, f_v, $f_{\Delta v}$, in the linearized form of the Intelligent Driver Model (IDM)) and without relying on direct measurements of HDV speed or spacing to the preceding vehicle.

3. To ensure **weak string stability** within the platoon, employ a robust control approach that suppresses the amplification of velocity errors between adjacent CAVs, in accordance with the Information Flow Topology (IFT) structure, which will be discussed in later sections.

4. Implement dynamic output-feedback algorithms and modify their formulation to reduce computational complexity and improve scalability in large-scale systems with multiple measurements, and with a specific emphasis on application to vehicle platooning.

The analysis and simulations presented in this thesis are based on the following assumptions.

1. The number of HDVs in the platoon is known a priori.

2. There is no communication delay between the CAVs.

3. Lateral vehicle dynamics are neglected; only longitudinal motion is considered.

The main contributions of this thesis are summarized below, highlighting the key developments and insights introduced in this work:

- Proposing a model-free, output-feedback control framework suitable for mixed platoons with limited state knowledge and without prior knowledge of human driving models.
- Introducing a modified dynamic output feedback algorithm that significantly reduces computational complexity in large-scale systems. The suggested modification can be applied to both LQR and $\mathcal{H}_\infty$ output feedback model-free designs (based on ADP) and is not limited to the application of mixed platoons.
- Employing a $\mathcal{H}_\infty$ control strategy, formulated as a zero-sum game, to achieve weak string stability in a partially measurable mixed platoon, under a mixed platoon Information Flow Topology (IFT) structure. The advantage of our formulation based on $\mathcal{H}_\infty$ is the string stability criterion embedded in the control design criterion. A comparison is made with an LQR-based design to emphasize this advantage.
- Evaluating the proposed approach through simulations in various mixed platoon scenarios, with a focus on string stability performance.

Chapter 2

Human Driving Modeling and Simulation

2.1 Velocities of Interest

According to [39], the instability of the human driver platoons tends to occur mainly at low to medium speeds, especially when the average speed is around 10 to 70 km/h. In this range of speeds, traffic disturbances such as braking or small speed changes may intensify and spread along the line of vehicles, leading to the phenomenon of "stop and go waves".

2.2 Intelligent Driver Model (IDM)

In traffic flow modeling, the Intelligent Driver Model (IDM) is a car following model that describes the position and velocities of single-lane vehicles (driven by human drivers), and is used to simulate human driver behavior in urban traffic or on a highway [40]. The IDM car-following model (see Figure 2.1) is defined by: x_i is the position of the i vehicle, v_i is the velocity of the i vehicle, s_i is the front-bumper to front-bumper distance, and $i-1$ is the index of the preceding vehicle in the platoon. In addition, $s_i := x_{i-1} - x_i - l_{i-1}$ is the distance from the preceding vehicle. Note that the length of the preceding vehicle, l_{i-1}, is not taken into account later when developing dynamic equations, as it is assumed to be negligible. Lastly, $\Delta v := v_{i-1} - v_i$ is the speed difference between vehicle i and vehicle $i-1$.

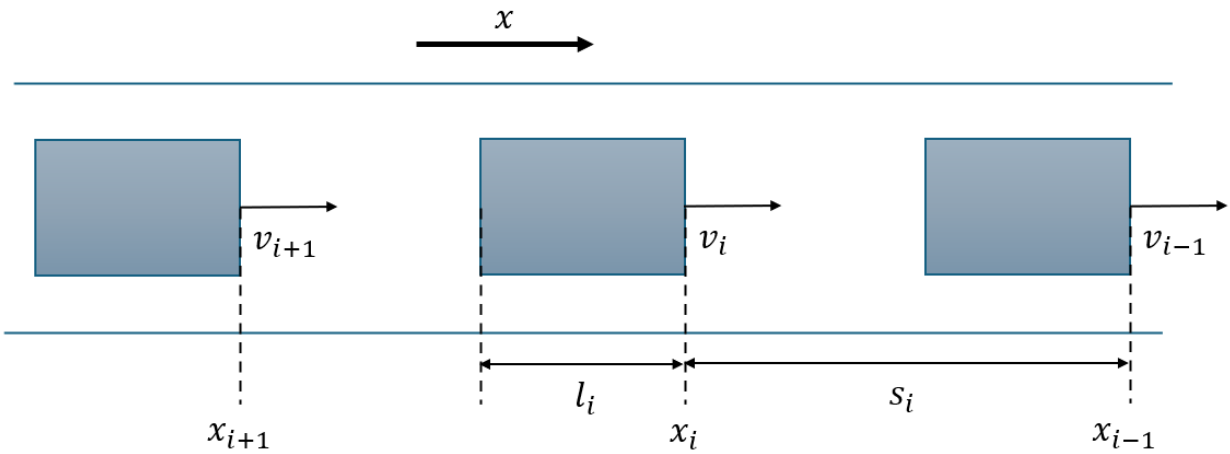

Figure 2.1: General car-following model set-up.

According to the IDM model, the acceleration of the HDV in a platoon is a function of v_i, s_i, Δv_i, and the desired distance s^*, with $s^*(v_i, \Delta v_i) = s_0 + v_i T + \frac{v_i \Delta v_i}{2\sqrt{ab}}$,

$$\dot{x}_i = v_i \tag{2.1}$$

$$a_{HDV} = \dot{v}_i = a_0 \left[1 - (\frac{v_i}{v_0})^\delta - \left(\frac{s^*(v_i, \Delta v_i)}{s_i} \right)^2 \right] \tag{2.2}$$

The HDV acceleration can be separated into the free road acceleration and the contribution of vehicle interaction.

$$a_i^{free} = a_0 \left[1 - (\frac{v_i}{v_0})^\delta \right] \qquad a_i^{interaction} = -a_0 \left(\frac{s^*(v_i, \Delta v_i)}{s_i} \right)^2 \tag{2.3}$$

The IDM model parameters are defined in Table 2.1.

Parameter	Symbol	Value
v_0	Desired speed	30 [m/s]
T	Safe time headway	1.5 [s]
a	Maximum acceleration	0.73 $[m/s^2]$
b	Comfortable Deceleration	1.67 $[m/s^2]$
δ	Acceleration exponent	4
s_0	Minimum distance	2 [m]
l	Vehicle length	5 [m]

Table 2.1: Calibrated IDM model

In order to describe the driver behavior by a linear model (for a linear control design framework), we need to linearize the IDM model. As we see in [41], the acceleration of the vehicle i is a

function three variables,

$$a_i = f(s_i, \Delta v, v_i) \tag{2.4}$$

Considering small perturbations around the equilibrium point $f(s(v_i), 0, v_i) = 0$, where the equilibrium velocity is $v_i = V$ and the equilibrium distance is $s_i(V)$, the acceleration function can be modeled as,

$$a_i = f_s \Delta s_i + f_{\Delta v_i} \Delta v_i - f_v e_{v,i} \tag{2.5}$$

In the linearized model, Δs_i is the distance deviation from equilibrium, $\Delta s_i = x_{i-1} - x_i - s(V)$, and $e_{v,i} = v_i - V$ is the velocity deviation from the equilibrium velocity (V that is equivalent to all platoon members).

Figure 2.2, taken from [3], shows the values of the linearized model coefficients, f_s, $f_{\Delta v_i}$, and f_v as a function of different equilibrium velocities.

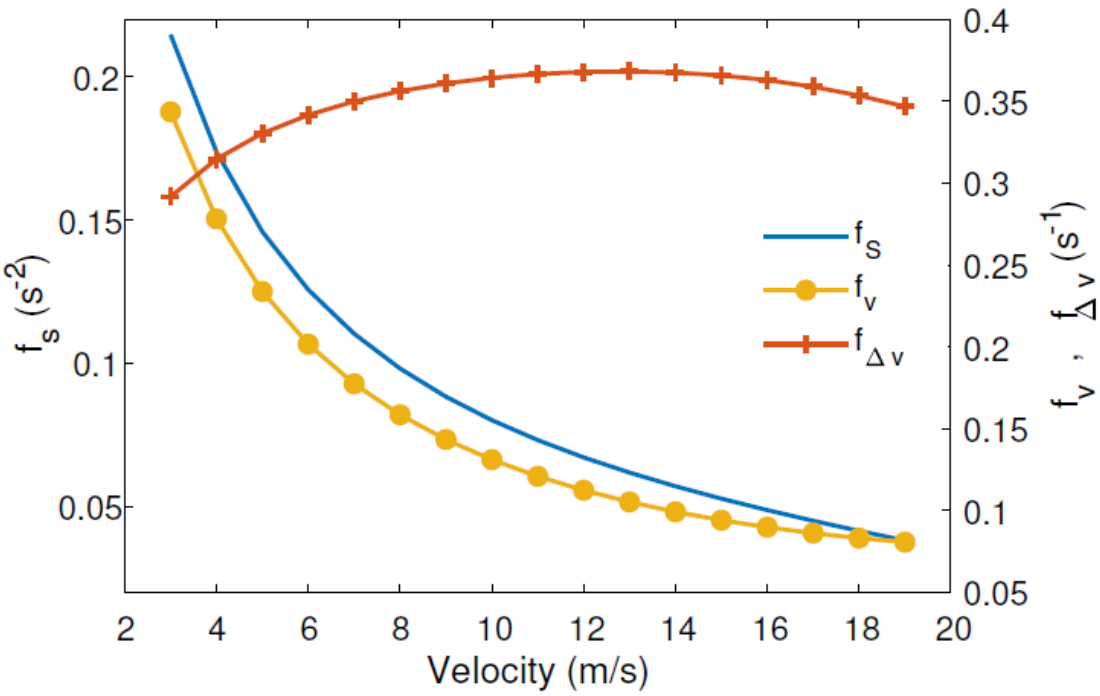

Figure 2.2: Linearized IDM model coefficients as a function of velocity, from [3].

2.3 Platoon Human Driving Simulation

The simulation was developed in the MATLAB environment, and the vehicle dynamics were modeled using Simulink (using the model structure in Figure 2.3). Note that the IDM model for simulations was implemented in its original nonlinear form, while the linearized model is only utilized for control design.

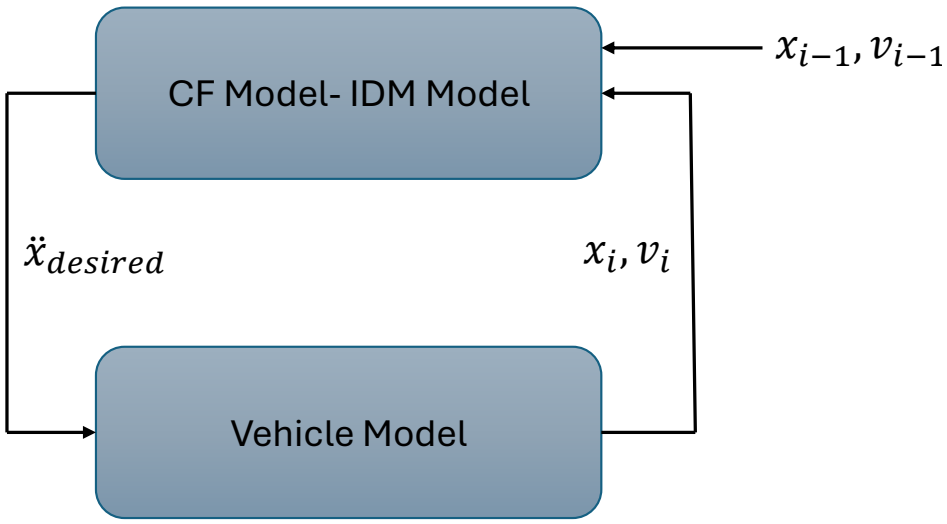

Figure 2.3: single HDV model.

The dynamics of the vehicle was modeled as a double integrator system [3], where the acceleration is defined as $a_i = \ddot{x}_i = \frac{1}{s^2}x_i$, and the velocity is $v_i = \dot{x}_i$. A simulation scenario involving a platoon of nine HDVs is presented in Figure 2.4. In this simulation, the lead vehicle performed two sudden braking maneuvers. The result shows that the disturbance propagation is amplified along the platoon, indicating the human driver's lack of string stability.

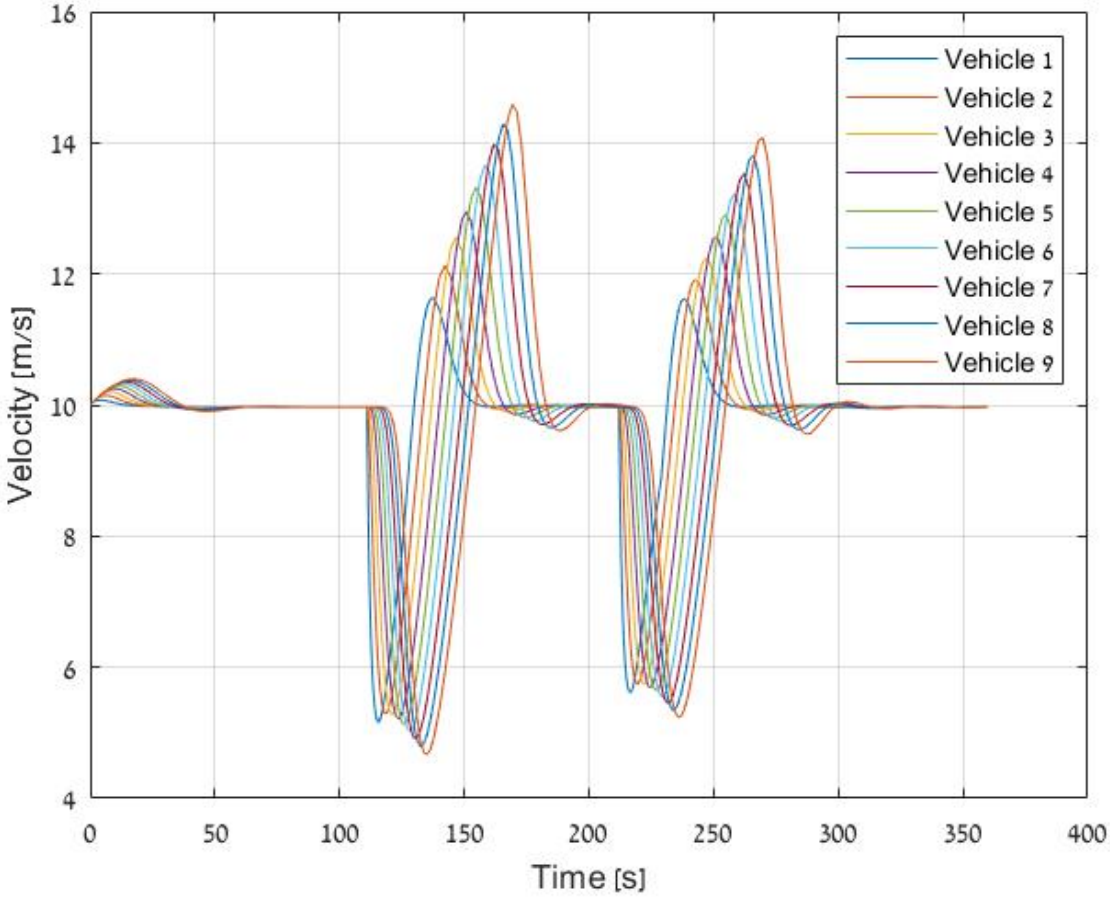

Figure 2.4: Simulation of HDVs platoon with disturbances.

Chapter 3

Mixed Platoon Model Based on IDM

This chapter outlines the formulation of a mixed vehicle platoon, in which the linearized IDM model is used to represent the dynamics of human-driven vehicles (HDV). This model is developed for control design purposes. A comparable methodology was previously introduced in [3].

3.1 Automated Platoon Control Components

This section outlines the core components of automated vehicle platooning to establish a foundation for mixed platoon control. A typical platoon consists of a leading vehicle (indexed $i = 0$) and a series of following vehicles (indexed $i = 1, 2, \ldots$). The control framework for such platoons can be categorized into the following four components [42]:

1. **Node Dynamics (ND)** refers to the mathematical representation of the vehicle's motion dynamics within the platoon. Although both second-[43]- and third-order models are commonly used, where the latter includes engine dynamics approximations [44][45], this work adopts a second-order model to describe vehicle behavior for simplicity and analytical tractability.

2. **Information Flow Topology (IFT)** determines the structure of information exchange between vehicles. A widely used topology is the predecessor-following with leader information, where each vehicle receives data from both its immediate predecessor and the

platoon leader.[43], [45].

3. **Formation Geometry (FG)** specifies the desired inter-vehicle spacing policy. Two common approaches are the constant spacing policy [43], [45], and the constant time headway policy, which maintains safe distances based on vehicle velocity. [44], [46].

4. **Distributed Controller (DC)** implements the control law for each vehicle in a decentralized manner. Most suggested controllers are linear. In this work, we adopt the $\mathcal{H}_\infty$ control strategy due to its disturbance attenuation capabilities. Relevant studies were made in [4], [44].

3.2 String Stability in Mixed Platoons

While basic stability ensures that each vehicle in the platoon eventually reaches the desired velocity and spacing, this criterion alone is insufficient to ensure safe and robust platoon behavior. An equally important property is ***string stability***, which refers to the system's ability to suppress the propagation of disturbances along the vehicle string. Specifically, a string-stable system prevents amplification of disturbances, such as sudden accelerations or decelerations, as they travel upstream from one vehicle to another.

Traditionally, string stability is analyzed in the frequency domain using various norms to quantify amplification. Among them, the $\mathcal{H}_\infty$ norm is commonly employed due to its favorable analytical properties and will also serve as the basis for our analysis.

In the context of mixed traffic, where both automated (CAV) and human-driven vehicles (HDV's) coexist, a relaxed notion called ***weak string stability*** has been proposed. This concept tolerates local amplification of disturbances by certain HDV's, as long as the disturbance does not grow when measured across non-adjacent CAV's in the platoon. Thus, *weak string stability* provides a more practical and flexible criterion suitable for heterogeneous platoon compositions.

3.3 IFT for mixed platoon

The structure of information exchange within a vehicle platoon, known as the Information Flow Topology (IFT), plays a critical role in ensuring weak string stability, particularly due to the importance of leader information [42]. In the context of mixed traffic, traditional IFTs designed for fully automated platoons may not be suitable, since HDVs cannot access or transmit information and are limited in reacting to the immediate preceding vehicle.

To address this, specialized IFT strategies have been proposed for mixed platoons [4]. These strategies partition the platoon into fully automated subplatoons, each separated by one or more HDVs (see Figure 3.1). Within this structure, the leading CAV that follows an HDV is designated as a *subleader*, while the remaining CAVs behind it are termed *subplatoon followers*.

As described in [4], two hierarchical IFT layers are introduced: *intra* and *inter* subplatoon topologies. The intra-sub-platoon IFT assumes a fully automated environment, allowing any established topologies from prior literature to be applied. The innovation of [4] lies in the inter-sub-platoon IFT, where only subleaders are responsible for acquiring information from sources external to their sub-platoon.

Each subleader collects data from its preceding HDV and possibly from an additional CAV called the *flow leader*. Various configurations exist for selecting the flow leader. For example, the flow leader could be (a) the global platoon leader, (b) the preceding subleader, or (c) the closest automated vehicle preceding the subleader. These three options are illustrated in Figure 3.2.

In this study, **we adopt the third configuration** for selecting the flow-leader (i.e., type (c)), in which the closest CAV preceding the sub-leader provides for inter-sub-platoon information (i.e., serves as local leader). This topology is further developed and analyzed throughout this thesis as it offers a practical compromise between information availability and implementation feasibility in mixed platoon environments. In addition, the chosen control strategy is specifically designed to address scenarios in which the leading vehicle introduces the worst-case disturbance, ensuring robust performance under such critical conditions.

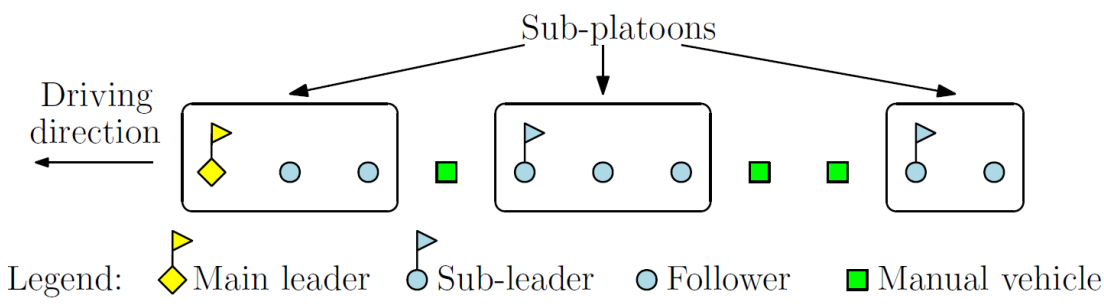

Figure 3.1: Division of a mixed-platoon into sub-platoons, from [4].

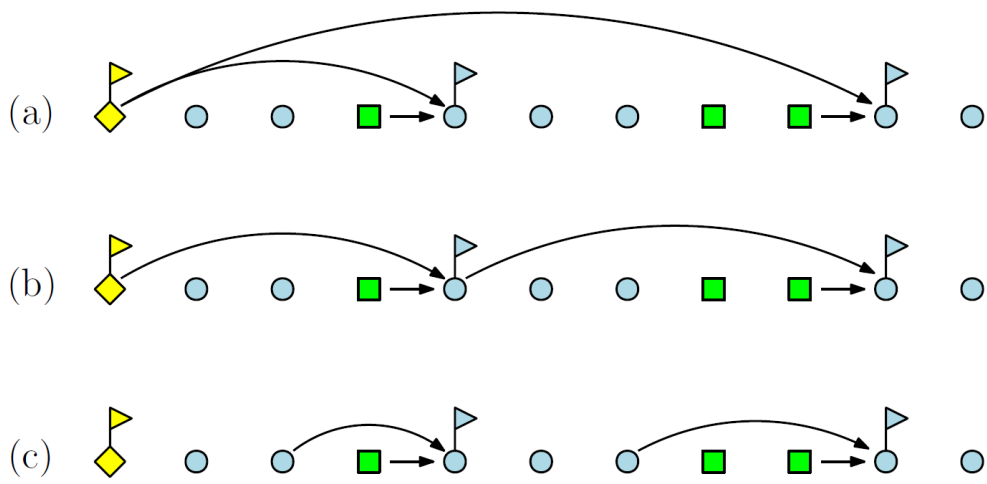

Figure 3.2: A demonstration of the three types of inter-sub-platoon IFTs for mixed platoons, from [4].

Focusing on a single subleader (see Figure 3.3), the involved dynamics (from the control point of view) comprises a flow leader followed by several HDVs, and a single CAV positioned at the tail of the platoon. Considering the other two IFT types (i.e., (a) and (b)), the configuration in Figure 3.3represents a conservative control scenario, often referred to as the 'worst case' IFT, since it excludes any CAVs between the controlled CAV and its flow leader. Unlike other IFT variants in which CAVs may be interleaved between HDVs, the proposed model ensures that the segment between the CAV and the flow-leader consists solely of HDVs. Since CAVs generally have a stabilizing effect, this assumption yields a less optimistic scenario [3], making any design designated to IFT type (c) also applicable to types (a) and (b).

The number of HDVs in front of the controlled subleader influences the dynamics required for its controller design. Fortunately, [3] proposed a method to estimate the number of preceding

HDVs between two communicating CAVs. It is based on the typical time headway of human drivers. Hence, we assume that the number of HDVs is known for our needs in this study.

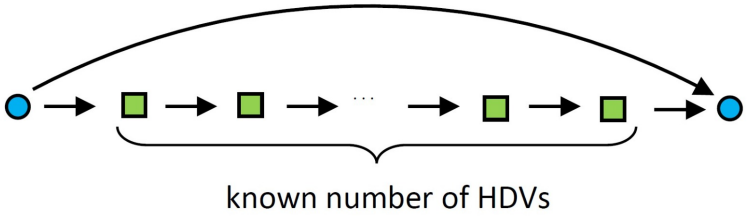

Figure 3.3: IFT of one CAV.

3.4 State-Space Modeling of Mixed Platoon

The state space representation of a mixed platoon consisting of HDVs and CAVs is based on the framework presented in [3]. In contrast to the original formulation, where external disturbances were embedded within the system dynamics, the current approach omits disturbances during the modeling phase. Instead, disturbances are incorporated later as part of the controller design process (see Chapter 4).

In this model, each HDV is represented by a linearized version of the Intelligent Driver Model (IDM), resulting in a second-order system with two state variables per vehicle. Each CAV, in contrast, is modeled as a double integrator with control input corresponding to acceleration, thereby contributing two states to the global state-space representation.

To both HDVs and CAVs, the deviation of the velocity of vehicle i from the constant platoon velocity V is denoted as

$$e_{v_i} = v_i - V \tag{3.1}$$

As described in Section 3.1, the inter-vehicle spacing policy described by the linearized HDVs model is the constant spacing policy, defined relative to a fixed desired spacing, and given by

$$\Delta s_i = x_{i-1} - x_i - s(V), \tag{3.2}$$

where $s(V)$ is the equilibrium spacing at the nominal speed V.

In contrast, a constant time-headway policy is adopted for CAVs. Under this policy, the desired

inter-vehicle spacing is a function of the vehicle's velocity. Accordingly, the space-headway error is defined as follows.

$$e_{s_i} = x_{i-1} - x_i - (s_0 + t_h \cdot v_i), \tag{3.3}$$

where s_0 is the standstill distance and t_h is the time headway parameter.

The HDV acceleration, as described by the linearized IDM model (2.5) is,

$$a_i = f_s \Delta s_i + f_{\Delta v} e_{v,i-1} - (f_{\Delta v} + f_v) e_{v,i} \tag{3.4}$$

The state-space realization of this model is,

$$\dot{e}_{h,i} = A_{h,1} e_{h,i} + A_{h,0} e_{h,i-1} \tag{3.5}$$

where,

$$e_{h,i} = \begin{bmatrix} \Delta s_i \\ \dot{e}_{v,i} \end{bmatrix}, A_{h,1} = \begin{bmatrix} 0 & -1 \\ f_s & -(f_{\Delta v} + f_v) \end{bmatrix}, A_{h,0} = \begin{bmatrix} 0 & 1 \\ 0 & f_{\Delta v} \end{bmatrix}$$

The state-space realization of the CAV is,

$$\dot{e}_{c,i} = A_{c,1} e_{c,i} + A_{c,0} e_{h,i-1} + b_c a_i \tag{3.6}$$

where,

$$e_{c,i} = \begin{bmatrix} e_{s,i} \\ \dot{e}_{v,i} \end{bmatrix}, A_{h,1} = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix}, A_{c,0} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, b_c = \begin{bmatrix} -t_h \\ 1 \end{bmatrix}$$

The combined state-space model of a mixed platoon consisting of n HDVs followed by a single CAV at the tail (as in the Figure 3.3)

$$\begin{bmatrix} \dot{e}_{h,1} \\ \dot{e}_{h,2} \\ \vdots \\ \dot{e}_{h,n} \\ \dot{e}_c \end{bmatrix} = \begin{bmatrix} A_{h0} & 0 & \dots & 0 & 0 & 0 \\ A_{h1} & A_{h0} & \dots & 0 & 0 & 0 \\ \vdots & & \ddots & & \vdots & \vdots \\ 0 & 0 & \dots & A_{h1} & A_{h0} & 0 \\ 0 & 0 & \dots & 0 & A_{c1} & A_{c0} \end{bmatrix} \begin{bmatrix} e_{h,1} \\ e_{h,2} \\ \vdots \\ e_{h,n} \\ e_c \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ b_c \end{bmatrix} u \tag{3.7}$$

This state-space model will be further augmented to include a disturbance input, which will be presented in the next chapter.

Chapter 4

Mixed Platoon Control

4.1 Overview

This section provides a concise overview of control design methods aimed at minimizing a predefined performance criterion. Two prominent approaches in this context are LQR and the $\mathcal{H}_\infty$ control methods.

The discussion begins by outlining the theoretical foundations for controller synthesis under the assumption of a **linear system**, aligning with the linear nature of the system addressed in this thesis. For a more comprehensive treatment of optimal control theory, the reader is referred to [27].

Understanding these classical control approaches is essential for the subsequent development of a control framework tailored to mixed vehicle platoons. In particular, the principles of optimal control discussed here will be leveraged in the next chapters to design stabilizing controllers capable of handling the uncertain dynamics inherent in heterogeneous platoon scenarios.

In addition, this background will serve as a theoretical foundation for the general control synthesis method proposed in Chapter 6, referred to as the *Reduced Dimension Output Matrix* approach. Although the method is illustrated through the mixed platoon control problem, it is formulated in the state-space domain and is thus applicable to **any system that can be represented in state-space form**. The proposed framework unifies the treatment of both LQR and $\mathcal{H}_\infty$ controllers, offering a flexible and generic approach to performance-driven control

design.

4.1.1 Linear Quadratic Regulator

In classical control theory, system stabilization is often achieved through pole placement, which involves shifting the poles of a system to desired locations to regulate specific states to zero and achieve certain performance objectives. However, when dealing with high-dimensional systems, it becomes challenging to determine which poles to move and how to position them to guarantee satisfactory performance across all states of the system.

The Linear Quadratic Regulator (LQR) approach addresses this challenge by formulating a performance criterion, typically a quadratic cost function, and deriving a control law that minimizes this criterion. This method provides a systematic and optimal way to balance state regulation and the control effort.

We begin by considering a linear time-invariant (LTI) system of the form

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{4.1}$$

where $x(t)$ is a vector of state variables, $u(t)$ is a vector of control inputs, and $y(t)$ is a vector of measured outputs. In the case of full state feedback, the matrix K represents the state-feedback gain, and the control input is defined as,

$$u = -Kx \tag{4.2}$$

4.1.1.1 Quadratic Performance Index

The goal of state regulation is to minimize deviations from the desired state over time, including those caused by initial conditions or external disturbances. This is achieved by selecting a control input $u(t)$ that minimizes the following quadratic cost function:

$$J = \int_0^\infty \left(x^T Q x + u^T R u\right)\, dt \tag{4.3}$$

where $Q \geq 0$ and $R > 0$ are weighting matrices.

4.1.1.2 Solving The LQR Problem

Define a symmetric positive-definite matrix $P = P^T$, and add and subtract the term $x_0^T P x_0$ from the cost function, then,

$$J = x_0^T P x_0 - x_0^T P x_0 + \int_0^\infty (x^T Q x + u^T R u) dt \tag{4.4}$$

Insert the term $-x_0^T P x_0$ into the integral, and use $[_{x^T P x}]_0^\infty = 0 - x_0^T P x_0$, it to get,

$$J = x_0^T P x_0 + \int_0^\infty (\frac{d}{dt}(x^T P x) + x^T Q x + u^T R u) dt \tag{4.5}$$

From,

$$\frac{d}{dt}(x^T P x) = \dot{x}^T P x + x^T P \dot{x} \tag{4.6}$$

and by substituting $\dot{x} = Ax + Bu$,

$$J = x_0^T P x_0 + \int_0^\infty [(Ax + Bu)^T P x + x^T P (Ax + Bu) + x^T Q x + u^T R u] dt \tag{4.7}$$

Collecting all $x^T(\cdot)x$ terms together,

$$J = x_0^T P x_0 + \int_0^\infty [x^T (A^T P + PA + Q) x + u^T R u + x^T P B u + u^T B^T P x] dt \tag{4.8}$$

The terms $x_0^T P x_0$ and $x^T(A^T P + PA + Q)x$ do not depend on the control input u, and therefore cannot be used to minimize the cost function J directly. In contrast, the expression involving u can be simplified by completing the square, which results in,

$$u^T R u + x^T P B u + u^T B^T P x = \left(u + R^{-1} B^T P x\right)^T R \left(u + R^{-1} B^T P x\right) - x^T P B R^{-1} B^T P x \tag{4.9}$$

By substituting this back in J,

$$J = x_0^T P x_0 + \int_0^\infty [x^T (A^T P + PA + Q - PBR^{-1} B^T P) x + \left(u + R^{-1} B^T P x\right)^T R \left(u + R^{-1} B^T P x\right)] dt \tag{4.10}$$

Hence, the minimizing (optimal) controller is,

$$u = -Kx = -R^{-1}B^T Px \qquad \Rightarrow \qquad K = R^{-1}B^T P \tag{4.11}$$

The gain matrix depends on the symmetric matrix P, which is calculated from the algebraic Riccati equation (ARE),

$$A^T P + PA + Q - PBR^{-1}B^T P = 0 \tag{4.12}$$

From these steps, the minimal value of J is $x_0^T P X_0$; hence, the cost is proportional to P. The positive-definite solution of the ARE is the smallest stabilizing P.

4.1.2 Two-Player Zero-Sum Games

Zero-sum game (ZSG) refers to a situation that involves two competing players. Inevitably, one player's gain leads to an equal loss for the other player, and vice versa. Nash solution of the 2-player ZS game provides a solution to the bounded problem L_2 gain, [47][48][49].
We now formulate the $\mathcal{H}_\infty$ control problem using the zero-sum differential game (ZSG) framework. This formulation plays an important role in the control design framework we suggest for mixed platoons. Let the system model be represented by,

$$\dot{x} = Ax + B_1 u_1 + B_2 u_2 \tag{4.13}$$

where u_1 is the control input (controller), and u_2 is the disturbance input. From a differential game point of view, this system is governed by two players u_1 and u_2.

We look for a controller u_1 that solves the zero-sum game (ZSG),

$$\min_{u_1} \max_{u_2} \int_0^\infty \left(x^T Q x + u_1^T R u_1 - \gamma^2 u_2^T u_2 \right) dt \tag{4.14}$$

To proceed, we define

$$J = \int_0^\infty \left(x^T Q x + u_1^T R u_1 - \gamma^2 u_2^T u_2 \right) dt \tag{4.15}$$

Following the same procedure as presented for the LQR, we get,

$$J = x_0^T P x_0 + \int_0^\infty \left(\dot{x}^T P x + x^T P \dot{x} + x^T Q x + u_1^T R u_1 - \gamma^2 u_2^T u_2\right) dt \tag{4.16}$$

By substituting the $\dot{x}$ from the system model,

$$\begin{aligned} J &= x_0^T P x_0 + \int_0^\infty \Big((Ax + B_1 u_1 + B_2 u_2)^T P x + x^T P (Ax + B_1 u_1 + B_2 u_2) \\ &\quad + x^T Q x + u_1^T R u_1 - \gamma^2 u_2^T u_2\Big) dt \\ &= x_0^T P x_0 + \int_0^\infty \Big(x^T (A^T P + PA + Q) x + u_1^T B_1^T P x + u_2^T B_2^T P x \\ &\quad + x^T P B_1 u_1 + x^T P B_2 u_2 + u_1^T R u_1 - \gamma^2 u_2^T u_2\Big) dt \end{aligned} \tag{4.17}$$

Here, we complete the square twice

$$\begin{aligned} u_1^T R u_1 + x^T P B_1 u_1 + u_1^T B_1^T P x &= (u_1 + R^{-1} B_1^T P x)^T R (u_1 + R^{-1} B_1^T P x) - x^T P B_1 R^{-1} B_1^T P x \\ u_2^T B_2^T P x + x^T P B_2 u_2 - \gamma^2 u_2^T u_2 &= -(u_2 - \gamma^{-2} B_2^T P x)^T \gamma^2 (u_2 - \gamma^{-2} B_2^T P x) + x^T P B_2 \gamma^{-2} B_2^T P x \end{aligned} \tag{4.18}$$

which reformulates J as,

$$\begin{aligned} J = x_0^T P x_0 + \int_0^\infty \Big(&x^T (A^T P + PA + Q - P B_1 R^{-1} B_1^T P + P B_2 \gamma^{-2} B_2^T P) x \\ &+ (u_1 + R^{-1} B_1^T P x)^T R (u_1 + R^{-1} B_1^T P x) \\ &- (u_2 - \gamma^{-2} B_2^T P x)^T \gamma^2 (u_2 - \gamma^{-2} B_2^T P x)\Big) dt \end{aligned} \tag{4.19}$$

Then, the minimizing u_1 is,

$$u_1 = -R^{-1} B_1^T P x \tag{4.20}$$

and the maximizing u_2 is,

$$u_2 = \gamma^{-2} B_2^T P x \tag{4.21}$$

Note that u_2 is in fact *the worst-case disturbance* (and not necessarily the actual disturbance).

Lastly, $P > 0$ is calculated from the Riccati equation,

$$A^T P + PA + Q - P B_1 R^{-1} B_1^T P + P B_2 \gamma^{-2} B_2^T P = 0 \tag{4.22}$$

and the ZSG results in,

$$\min_{u_1} \max_{u_2} J = x_0^T P x_0 \tag{4.23}$$

Note that this value of J is obtained from applying the optimal controller and the worst-case disturbance; it represents a Nash equilibrium.

We will now show that the solution of the ZS-Game also solves the $\mathcal{H}_\infty$ control design problem.

The $\mathcal{H}_\infty$ control problem (also known as the bounded L_2 gain design problem) is formulated as follows, For the system (4.13), we are interested in $u_1(t)$, such that for $x(0) = 0$ and $u_2(t) \in L_2[0, \infty)$ we have,

$$\frac{\int_0^\infty \|z(t)\|^2 dt}{\int_0^\infty \|u_2(t)\|^2 dt} = \frac{\int_0^\infty \left(x^T Q x + u_1^T R u_1\right) dt}{\int_0^\infty \|u_2(t)\|^2 dt} \leq \gamma^2 \tag{4.24}$$

Note that the L_2 norm of $u_2(t)$ is $\sqrt{\int_0^\infty \|u_2(t)\|^2 dt}$, and the requirement for $u_2(t) \in L_2[0, \infty)$ means that it has a finite L_2 norm. Fulfilling (4.24) demands also a finite L_2 gain of $x(t)$ and therefore stability of the closed loop. The signal $z(t)$ is referred to as the controlled output.

The significance of the design criterion (4.24) is that the obtained $u_1(t)$ limits the influence of the disturbance $u_2(t)$ on $z(t)$ (controlled output). In other words, the $\mathcal{H}_\infty$ controller guarantees a bound (of γ^2) on the system from $u_2(t)$ to $z(t)$, a feature that will be later utilized for guaranteeing string stability.

Note that the minimal possible γ^2 (i.e., $\gamma^2_{\min}$) is the $\mathcal{H}_\infty$ norm of the system. A design to obtain $\gamma^2_{\min}$ is the optimal $\mathcal{H}_\infty$ design; otherwise (from a $\mathcal{H}_\infty$ perspective), the design is suboptimal. Nevertheless, it is still a solution for a ZSG.

Recall the ZSG solution for the case where $x(0) = 0$,

$$\begin{aligned} J = &\int_0^\infty x^T (A^T P + PA + Q - PB_1 R^{-1} B_1^T P + PB_2 \gamma^{-2} B_2^T P) x \, dt \\ &+ \int_0^\infty \left((u_1 + R^{-1} B_1^T P x)^T R (u_1 + R^{-1} B_1^T P x) - (u_2 - \gamma^{-2} B_2^T P x)^T \gamma^2 (u_2 - \gamma^{-2} B_2^T P x) \right) dt \end{aligned} \tag{4.25}$$

Substituting $u_1 = -R^{-1} B_1^T P x$, where (from the relevant ARE),

$$A^T P + PA + Q - PB_1 R^{-1} B_1^T P + PB_2 \gamma^{-2} B_2^T P = 0 \tag{4.26}$$

we get,

$$J = -\int_0^\infty (u_2 - \gamma^{-2} B_2^T P x)^T \gamma^2 (u_2 - \gamma^{-2} B_2^T P x) dt \tag{4.27}$$

This relation holds for any disturbance $u_2(t)$, not necessarily the worst-case disturbance. Then, by specifying the expression of J,

$$\int_0^\infty \left(x^T Q x + u_1^T R u_1 - \gamma^2 u_2^T u_2\right) dt = -\int_0^\infty (u_2 - \gamma^{-2} B_2^T P x)^T \gamma^2 (u_2 - \gamma^{-2} B_2^T P x) dt \tag{4.28}$$

The right-hand side of the equality (above) is negative for any u_2 with a bounded L_2 norm u_2 (other than the worst case u_2, which causes this side to be zero). Then we can write,

$$\begin{aligned} &\int_0^\infty \left(x^T Q x + u_1^T R u_1 - \gamma^2 u_2^T u_2\right) dt \leq 0 \\ &\Rightarrow \\ &\frac{\int_0^\infty \left(x^T Q x + u_1^T R u_1\right) dt}{\int_0^\infty u_2^T u_2 dt} \leq \gamma^2, \quad \forall u_2(t) \in L_2[0, \infty) \end{aligned} \tag{4.29}$$

4.2 Controller Design for Mixed-Platoon by ZSG

In chapter 3, we presented a model of a mixed platoon (see Eq. 3.7), and in the previous section, we gave background on the zero-sum game; now, these two notions are combined together.

Consider a mixed platoon of vehicles, where the first vehicle is a CAV (serves as the leader), followed by several HDVs, and a controlled CAV that closes this platoon structure (for illustration, see Figure 3.3). Assume this mixed platoon structure is modeled by,

$$\begin{aligned} \dot{x} &= Ax + B_1 u_1 + B_2 u_2 \\ y &= Cx \end{aligned} \tag{4.30}$$

where u_1 is the control input of the closing CAV, and u_2 is a velocity disturbance of the leading CAV.

If the leading CAV applies a disturbance, its influence propagates through the HDVs and reaches the closing CAV. The closing (controlled) CAV applies a control input ($u_1(t)$), defined as its acceleration, to achieve two goals: 1) to regulate its spacing with the preceding HDV, and 2) to attenuate the disturbance (which has probably been amplified by the middle HDVs),

and obtain weak string stability between the two CAVs. Assuming the velocity tracking error of the controlled CAV is weighted in the controlled output $z(t)$ (which is a signal determined by the designer), then the following $\mathcal{H}_\infty$ control objective,

$$\frac{\int_0^\infty z^T(t)z(t)\,dt}{\int_0^\infty u_2^T(t)u_2(t)\,dt} \leq \gamma^2 = 1. \tag{4.31}$$

can serve as a formal demand in the design of $u_1(t)$ to guarantee weak string stability between the two CAVs. Using this formulation, the amplification (or attenuation) of the influence of a disturbance that is propagating along the platoon is measured by the ratio of L_2 norms of the disturbance and its influence (as reflected in $z(t)$)

Since the first HDV in the considered mixed-platoon structure is influenced directly from the leading CAV disturbance, for this HDV,

$$\begin{aligned} \Delta s_1 &= x_0 - x_1 - s(V) \\ e_{v,1} &= v_1 - V \end{aligned} \tag{4.32}$$

and,

$$\begin{aligned} \dot{\Delta} s_1 &= \dot{x}_0 - \dot{x}_1 = e_{v,0} - e_{v,1} \\ \dot{e}_{v,1} &= a_i = f_s e_{s,1} + f_\Delta v e_{v,0} - (f_\Delta v + f_v) e_{v,1} \end{aligned} \tag{4.33}$$

where we define,

$$e_{v,0} = v_0 - V = u_2 \tag{4.34}$$

as the input (velocity) disturbance. Hence in (4.30), the model matrix B_2 is given by,

$$B_2 = \begin{bmatrix} 1 \\ f_{\Delta v} \\ \vdots \\ 0 \\ 0 \end{bmatrix} \tag{4.35}$$

and we define,

$$b_h = \begin{bmatrix} 1 \\ f_{\Delta v} \end{bmatrix} \tag{4.36}$$

The rest of the platoon model has been explained in Chapter 3, and for a mixed-platoon of n-HDVs, the general model form is,

$$\begin{bmatrix} \dot{e}_{h,1} \\ \dot{e}_{h,2} \\ \vdots \\ \dot{e}_{h,n} \\ \dot{e}_c \end{bmatrix} = \begin{bmatrix} A_{h0} & 0 & \dots & 0 & 0 & 0 \\ A_{h1} & A_{h0} & \dots & 0 & 0 & 0 \\ \vdots & & \ddots & & \vdots & \vdots \\ 0 & 0 & \dots & A_{h1} & A_{h0} & 0 \\ 0 & 0 & \dots & 0 & A_{c1} & A_{c0} \end{bmatrix} \begin{bmatrix} e_{h,1} \\ e_{h,2} \\ \vdots \\ e_{h,n} \\ e_c \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ b_c \end{bmatrix} u_1 + \begin{bmatrix} b_h \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} u_2 \tag{4.37}$$

where the subscripts h and c refer to the HDVs and CAVs, respectively.

The output matrix C in (4.30) determines the measurement signal $y(t)$, which is the information available for control. We follow the approach presented in [3], where the mixed-platoon control is based on three tracking error signals. For that, we define,

$$\boldsymbol{E} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} x_{i-1} - x_i - (s_0 + t_h v_i) \\ v_{i-1} - v_i \\ V - v_i \end{bmatrix} \in \mathbb{R}^3 \tag{4.38}$$

where,

1. $e_1(t)$ is the spacing error of the controlled CAV with respect to the preceding HDV vehicle, computed under a constant time headway policy. This information is essential for the stability of the controlled CAV.

2. $e_2(t)$ is the velocity error of the controlled CAV relative to the preceding HDV vehicle. This information is also essential for the stability of the controlled CAV. Since in the state space model (3.7), all velocity tracking errors are formulated with respect to the platoon steady-state velocity V (as determined by the leading CAV), $e_2(t)$ can be expressed (in terms of the model state variables) as,
   $e_2 = e_{v,i-1} - e_{v,i} = (v_{i-1} - V) - (v_i - V) = v_{i-1} - v_i$.

3. $e_3(t)$ is the velocity error with respect to the first CAV (which serves here as a platoon leader and determines the steady-state platoon velocity V), then in terms of the model state variables, it is expressed as

$$e_3 = -(v_i - V) = -e_{v,i}.$$

Note that the definition of the mixed-platoon state variable is not unique, and our choice here is consistent with that of [3]. Also, in the definitions above, x_i and v_i denote the position and velocity of vehicle i, respectively, s_0 is the standstill distance, and t_h is the desired time headway. The reference velocity V is constant and dictated by the first CAV, not considering its injected disturbances.

Note that the distance between the controlled CAV and leading CAV is not explicitly controlled, as it may vary significantly depending on the middle HDV behavior, and as the HDVs are not communicating, this distance can not be controlled (the required information is not available). For a configuration consisting of n HDVs followed by a single CAV, the output equation of (4.30) is,

$$y = Cx \in \mathbb{R}^3, \quad \text{where} \quad C = \begin{bmatrix} 0 & \cdots & 0 & 1 & 0 \\ 0 & \cdots & 1 & 0 & -1 \\ 0 & \cdots & 0 & 0 & -1 \end{bmatrix}, \quad x = \begin{bmatrix} \vdots \\ \Delta s_{i-1} \\ e_{v,i-1} \\ e_{s,i} \\ e_{v,i} \end{bmatrix} \in \mathbb{R}^{2(n+1)} \tag{4.39}$$

Chapter 5

Adaptive Dynamic Programming Based on Reinforcement Learning

5.1 Introduction

This chapter provides a concise overview of three key algorithms that serve as foundational elements in the development of adaptive dynamic programming (ADP) control strategies. The discussion begins with the most basic approach: a model-based offline policy iteration (PI) algorithm for solving the Linear Quadratic Regulator (LQR) problem. This method assumes full knowledge of the system dynamics and access to all state variables (i.e., it is a state-feedback approach). Subsequently, we introduce a model-free PI algorithm for LQR that still relies on state feedback but does not require knowledge of the system model. Finally, we examine a model-free output-feedback PI algorithm for LQR that eliminates the need for full state measurements. In parallel, we present the corresponding PI algorithms developed for the $\mathcal{H}_\infty$ (ZSG) control framework. Value Iteration (VI) methods were intentionally not included in this study due to practical safety considerations. The advantage of VI is that no initial stabilizing controller is needed, but this is irrelevant for the platoon that has to be always stable. Our goal is not just to stabilize the platoon but also to optimize its performance and guarantee string stability.

5.2 Model-Based PI

Even for a known system model, there is still a difficulty in solving the ARE (4.12) due to its non-linear nature. In (4.11), we found that the optimal controller is of the form $u = -R^{-1}B^TPx = -Kx$, i.e., a state-feedback. For this development, we first assume a closed-loop linear system with any stabilizing state-feedback controller (where K is not necessarily the optimal),

$$\dot{x} = (A - BK)x(t), \quad x(0) = x_0 \tag{5.1}$$

Consequently,

$$x(t) = e^{(A-BK)t}x_0 \tag{5.2}$$

Substituting this in J gives,

$$\begin{aligned} J &= \int_0^\infty \left(x^T(t)Qx(t) + u^T(t)Ru(t)\right) dt \\ &= \int_0^\infty \left(x^T(t)Qx(t) + x^T(t)K^TRKx(t)\right) dt \\ &= \int_0^\infty x^T(t)\left(Q + K^TRK\right)x(t)dt \\ &= x_0^T\left(\int_0^\infty e^{(A-BK)^Tt}\left(Q + K^TRK\right)e^{(A-BK)t}dt\right)x_0 \end{aligned} \tag{5.3}$$

Since K has been assumed to be stabilizing, the integration result is finite, and the cost of the state-feedback controller ($u = -Kx$, not necessarily the optimal) is,

$$J = x_0^TPx_0, \quad \text{where} \quad P = \int_0^\infty e^{(A-BK)^Tt}\left(Q + K^TRK\right)e^{(A-BK)t}dt \tag{5.4}$$

and from the structure of P we know that it is a solution of a Lyapunov equation of the form,

$$(A - BK)^TP + P(A - BK) + Q + K^TRK = 0 \tag{5.5}$$

Solving the Lyapunov equation above gives P, which is a measure for the value (or cost) of $u = -Kx$. It is clear that the optimal controller is the one with the minimal $P > 0$, which is also the positive-definite solution of the ARE. The ARE is nonlinear with respect to P, while the Lyapunov equation (5.5) is linear.

We will now review the method of Kleinman's ([28]), which replaces the solution of the ARE by a series of solutions of Lyapunov equations. Though this method is model-based, it is the basis for the model-free method to be presented next.

For a given stabilizing feedback gain K_i (with a detectable $A, Q^{1/2}$ couple), the following Lyapunov equation,

$$(A - BK_i)^T P_i + P_i(A - BK_i) + Q + K_i^T R K_i = 0 \tag{5.6}$$

gives the corresponding $P_i > 0$.

Kleinman (1968) theorem:

Assume that P_i, $i = 1, 2, \ldots$, are positive-definite solutions of (5.6), and,

$$K_{i+1} = R^{-1} B^T P_i \tag{5.7}$$

with, K_0 a stabilizing gain (i.e., $A - BK_0$ is stable), then,

1. $P \leq P_i \leq P_{i-1} \leq \cdots, \quad i = 0, 1, \ldots$
2. $\lim_{i \to \infty} P_i = P$

The algorithm goes as follows,

Model-Based LQR Policy Iteration

Input: Model information, A, B, design performance, Q, R, convergence criterion ϵ, and a stabilizing control policy K_0, set $i \to 0$

Output: A gain matrix K (near optimal) and the corresponding P

1: **Loop**

2: **Policy Evaluation:** Evaluate the current policy K_i by solving P_i in (5.6)

3: **Policy Improvement:** Update the policy by (5.7)

4: **If** $||P_i - P_{i-1}|| < \epsilon$ **then**

return $P = P_i, K = K_{i+1}$

5: **else** $i \to$i+1

6 : **End loop**

where ϵ is a small positive constant and $i \in \mathbb{Z}^+$. Note that K calculated by the Model-Based LQR Policy Iteration algorithm can be made as near as desired to optimality, just by reducing ϵ , at the cost of more iterations.

For the ZSG control design, the algorithm is modified to incorporate a second player, as formulated in [50]. Consider the following game dynamics,

$$\begin{aligned} \dot{x} &= Ax + B_1 u_1 + B_2 u_2 \\ y &= Cx \end{aligned}$$

where $x \in \mathbb{R}^n$ refers to the state vector, $u_1 \in \mathbb{R}^{m_1}$ is the input vector of Player 1 (controller), $u_2 \in \mathbb{R}^{m_2}$ is the input vector of Player 2 (disturbance), and $y \in \mathbb{R}^p$ is the output vector. The design problem is to find u_1 and u_2 such that,

$$J = \min_{u_1} \max_{u_2} \int_0^\infty \left(x^T Q x + u_1^T R u_1 - \gamma^2 \left\| u_2 \right\|^2 \right) dt \tag{5.8}$$

In this case, two policies are solved simultaneously, one for u_1 (the minimizing controller) and one for u_2 (the maximizing disturbance). Following the development presented in Chapter 4, the optimal policies for the two players are given by,

$$\begin{aligned} u_1 &= -Kx = -R^{-1} B_1^T P x \\ u_2 &= Hx = \gamma^{-2} B_2^T P x \end{aligned} \tag{5.9}$$

where P is the positive-definite solution of the ZSG-ARE,

$$A^T P + PA + Q - PB_1 R^{-1} B_1^T P + PB_2 \gamma^{-2} B_2^T P = 0. \tag{5.10}$$

Now, assume K_i and H_i are not necessarily the optimal solutions (but still stabilize the closed loop), then their value is $J_i = x_0^T P_i x_0$ and $P_i > 0$ can be calculated from the following Lyapunov equation [50],

$$(A - B_1 K_i + B_2 H_i)^T P_i + P_i (A - B_1 K_i + B_2 H_i) + Q + K_i^T R K_i - \gamma^{-2} H_i^T H_i = 0 \tag{5.11}$$

Adding also the two updated policy equations,

$$\begin{aligned} K_{i+1} &= R^{-1}B_1^T P_i \\ H_{i+1} &= \gamma^{-2}B_2^T P_i \end{aligned} \tag{5.12}$$

The ZSG iterative solution algorithm is,

Model-Based ZSG Policy Iteration

Input: Model information (A, B_1, B_2), design performance (Q, R, γ), convergence criterion ϵ, and stabilizing initial policies K_0, any H_0. set $i \to 0$

Output: Gain matrices K and H (near optimal), and the corresponding P matrix.

1: **Loop**

2: **Policy Evaluation:** Evaluate the current policies K_i, H_i by solving P_i in (5.11)

3: **Policy improvement:** Update the policies using (5.12)

4: **If** $||P_i - P_{i-1}|| < \epsilon$ **then**

return $P = P_i, K = K_{i+1}, H = H_{i+1}$

5 : **else**i→i+1

6 : **End loop**

Note that the convergence criterion $\epsilon > 0$ can be chosen as small as desired (which generally will require more iterations). For an open-loop stable system, the algorithm can be initialized with $K_0 = H_0 = 0$.

5.3 Model-Free PI using State Feedback

A model-free, state-feedback adaptive dynamic programming (ADP) policy iteration (PI) algorithm was proposed in [31], building upon the iterative approach introduced in [28], which relies on solving a series of Lyapunov equations. To develop the model-free variant, the system model (4.1) is represented as follows (where we add and subtract Bu_i),

$$\dot{x} = Ax + Bu_i + B(u - u_i). \tag{5.13}$$

We now define a value function $V(x) = x^T P x$, and for a policy $u_i = -K_i x$ (not necessarily optimal), we have,

$$V_i(t) = x^T(t) P_i x(t) = \int_t^\infty \left(x^T Q x + u_i^T R u_i\right) dt = \int_t^\infty \left(x^T Q x + x^T K_i^T R K_i x\right) dt \tag{5.14}$$

where, P_i satisfies (5.6).

Differentiating the value function (above) with respect to time and substituting $\dot{x}$ from (5.13),

$$\dot{V}_i = 2x^T P_i \left(Ax + Bu_i + B(u - u_i)\right), \tag{5.15}$$

and with $u_i = -K_i x$ it becomes,

$$\dot{V}_i = 2x^T P_i (A - BK_i) x + 2x^T P_i B (u + K_i x) \tag{5.16}$$

Since the elements in (5.16) are scalars (that are not changed by the transpose operation), we write,

$$\dot{V}_i = x^T \left((A - BK_i) P_i + P_i (A - BK_i)\right) x + 2(u + K_i x)^T B^T P_i x \tag{5.17}$$

By replacing $(A - BK_i)P_i + P_i(A - BK_i)$ with $-Q - K_i^T R K_i$ from (5.6),

$$\begin{aligned} \dot{V}_i &= -x^T \left(Q + (K_i)^T R K_i\right) x + 2(u + K_i x)^T B^T P_i x \\ &= -x^T Q_i x + 2(u + K_i x)^T B^T P_i x \end{aligned}$$

where, $Q_i = Q + (K_i)^T R K_i$.

The last step allowed the removal of the unknown model matrix A from the equation. To also eliminate the unknown model matrix B, we use $B^T P_i = R K_{i+1}$ from Eq.(5.7), to get,

$$\dot{V}_i = -x^T Q_i x + 2(u + K_i x)^T R K_{i+1} x = -x^T Q_i x + 2u^T R K_{i+1} x + 2x^T K_i^T R K_{i+1} x \tag{5.18}$$

Finally, we integrate on both sides from t to $t+T$,

$$\int_t^{t+T} \dot{V}_i \, d\tau = -\int_t^{t+T} x^T Q_i x \, d\tau + 2\int_t^{t+T} u^T R K_{i+1} x \, d\tau + 2\int_t^{t+T} x^T K_i^T R K_{i+1} x \, d\tau \tag{5.19}$$

ans use (for the left-hand side of the equation),

$$\int_t^{t+T} \dot{V}_i \, d\tau = V_i(t+T) - V_i(t) = x^T(t+T) P_i x(t+T) - x^T(t) P_i x(t) \tag{5.20}$$

The result, including (5.19) and (5.20), is the learning equation,

$$\begin{aligned} x^T(t+T) P_i x(t+T) - x^T(t) P_i x(t) = &-\int_t^{t+T} x^T(\tau) Q_i x(\tau) \, d\tau \\ &+ 2\int_t^{t+T} u^T(\tau) R K_{i+1} x(\tau) \, d\tau \\ &+ 2\int_t^{t+T} x^T(\tau) K_i^T R K_{i+1} x(\tau) \, d\tau \end{aligned} \tag{5.21}$$

from which P_i and K_{i+1} can be solved. The learning equation depends on data sets of $x(t)$ and $u(t)$. Since the learning equation (5.21) is scalar, and the number of unknowns in P_i and K_{i+1} is greater than 1, the data sets should be split into several time sections. For example, if $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$, then in order to uniquely identify the unknown parameters, it is necessary to obtain at least,

$$l \geq \frac{n(n+1)}{2} + mn \tag{5.22}$$

independent equations. This condition ensures that the number of equations matches or exceeds the number of unknowns. Still, the learning equation (5.21) is not easy to solve because the unknowns are matrices. Some Kronecker-product manipulations are necessary to extract the unknowns. These manipulations arrange the scalars of the unknown matrices as vectors.

The vec$(\cdot)$ operator transforms a matrix into a vector of its entries (ordered by the column vectors of the matrix). Assuming some general A, B, and C matrices of appropriate dimensions, then,

$$\mathrm{vec}(ABC) = (C^T \otimes A)\,\mathrm{vec}(B) \tag{5.23}$$

For a raw vector A and a column vector C, the product ABC is a scalar; hence, it becomes,

$$ABC = (C^T \otimes A)\,\mathrm{vec}(B) \tag{5.24}$$

Using the identity (5.24) above, we can write,

$$2\int_t^{t+T} u^T R K_{i+1} x \, d\tau = 2\int_t^{t+T} \left(x^T \otimes u^T R\right) d\tau \,\mathrm{vec}(K_{i+1}) \tag{5.25}$$

Utilizing also the following Kronecker-product property,

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD), \tag{5.26}$$

we have,

$$2\int_t^{t+T} u^T R K_{i+1} x \, d\tau = 2\int_t^{t+T} \left(x^T \otimes u^T\right) d\tau \,(I_n \otimes R)\,\mathrm{vec}(K_{i+1}) \tag{5.27}$$

In (5.27), only the data is being integrated, while the unknowns (i.e., the elements of K_{i+1}) are represented as a vector on the right-hand side of the equation. Similarly, we write,

$$2\int_t^{t+T} x^T K_i^T R K_{i+1} x \, d\tau = 2\int_t^{t+T} \left(x^T \otimes x^T\right) d\tau \left(I_n \otimes K_i^T R\right) \mathrm{vec}(K_{i+1}) \tag{5.28}$$

and also,

$$\int_t^{t+T} x^T Q_i x \, d\tau = \left(\int_t^{t+T} x^T \otimes x^T \, d\tau\right) \mathrm{vec}(Q_i). \tag{5.29}$$

Since symmetric matrices (such as the unknown $P = P^T$) have a special structure, it is not necessary to solve for all the matrix entries. Hence, a vec$(\cdot)$ operator dedicated to symmetric matrices (denoted by svec$(\cdot)$) will be defined.

Let $P \in \mathbb{R}^{n\times n}$ be a symmetric matrix ($P = P^T$), so that,

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots \\ * & p_{22} & \cdots \\ * & * & \ddots \end{bmatrix}.$$

The lower triangle of the matrix, marked by $*$ is determined by the upper triangle. The matrix P has n^2 entries, but only $\frac{n(n+1)}{2}$ unknowns. The operator $\text{svec}(P)$ arranges the elements of the matrix $P = P^T$ into a vector, but does not include the redundant entries due to symmetry.

$$\text{vec}(P) = N\,\text{svec}(P), \quad N \in \mathbb{R}^{n^2 \times \frac{n(n+1)}{2}} \tag{5.30}$$

The matrix N transforms between $\text{svec}(\cdot)$ and $\text{vec}(\cdot)$ representations. This concept is demonstrated by a simple example, let,

$$P = P^T = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix}$$

Then,

$$\text{svec}(P) = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix} \quad \text{and} \quad \text{vec}(P) = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 2 \\ 4 \\ 5 \\ 3 \\ 5 \\ 6 \end{bmatrix}$$

and,

$$\begin{bmatrix} 1 \\ 2 \\ 3 \\ 2 \\ 4 \\ 5 \\ 3 \\ 5 \\ 6 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{N} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}$$

Applying this procedure (and the identity (5.24)) to left-hand side of Eq.(5.21), we get,

$$x^T(t+T)P_i x(t+T) - x^T(t)P_i x(t) = \left[x^T(t+T) \otimes x^T(t+T) - x^T(t) \otimes x^T(t)\right] N \,\mathrm{svec}(P_i). \quad (5.31)$$

Now, we assume that the data has been collected over a sufficiently long time interval $[t_0, t_l]$. For the required number of independent learning equations, we divide the time interval into l sub-intervals, such that

$$l > \frac{n(n+1)}{2} + mn \quad (5.32)$$

Then we define,

$$\delta_i = [x(t_{i+1}) \otimes x(t_{i+1}) - x(t_i) \otimes x(t_i)], \quad (5.33)$$

$$\delta_{xx} = [\delta_1, \delta_2, \cdots, \delta_{l-1}]^T, \quad (5.34)$$

$$I_{xx} = \left[\int_{t_0}^{t_1} x \otimes x \, d\tau, \quad \int_{t_1}^{t_2} x \otimes x \, d\tau, \quad \cdots, \quad \int_{t_{l-1}}^{t_l} x \otimes x \, d\tau\right]^T, \quad (5.35)$$

$$I_{xu} = \left[\int_{t_0}^{t_1} x \otimes u \, d\tau, \quad \int_{t_1}^{t_2} x \otimes u \, d\tau, \quad \cdots, \quad \int_{t_{l-1}}^{t_l} x \otimes u \, d\tau\right]^T. \quad (5.36)$$

Based on these definitions, we can write the following linear-system of equations,

$$X_i \Theta_i = Y_i \quad (5.37)$$

where,

$$X_i = \left[-\delta_{xx}N, \quad 2I_{xu}(I_n \otimes R) + 2I_{xx}(I_n \otimes K_i^T R)\right], \tag{5.38}$$

$$Y_i = I_{xx}\,\mathrm{vec}(Q_i), \tag{5.39}$$

$$\Theta_i = \begin{bmatrix} \mathrm{svec}(P_i) \\ \mathrm{vec}(K_{i+1}) \end{bmatrix}. \tag{5.40}$$

The vector of unknowns, Θ_i, is recalculated iteratively until convergence. Since we typically look for more equations than there are unknowns (to overcome the influence of measurement noise), the solution is obtained in the least-squares sense,

$$\Theta_i = (X_i^T X_i)^{-1} X_i^T Y_i \tag{5.41}$$

The algorithm proceeds as follows: a stabilizing initial controller is applied to collect data, after which policy evaluation and improvement steps are performed iteratively until convergence. The complete procedure is formulated here as an algorithm and illustrated in the flowchart in Figure 5.1.

Model-Free PI for LQR Using State-Feedback

Input: Design performance (Q, R), convergence criterion ϵ, and a stabilizing initial policy K_0.

Output: Gain matrix K (near optimal), and the corresponding P matrix.

1: **Initialize:** Construct an excitation initial input $u^0 = K_0 x + \nu$ where ν is exploration noise. Set $i \to 0$.

2: **Acquire Data.** Apply u_0 during $t \in [t_0, t_l]$ and collect input-state (u, x) data sets, divided into the time intervals $t_i - t_{i-1} = T_i, i = 1, 2, ..., l$.

3: **Loop**

4: **Policy Evaluation and Policy Improvement:** Solve for K_{i+1} and P_i in (5.37).

5: **If** $||P_i - P_{i-1}|| < \epsilon$ **then**

return P_i, K_{i+1}

6: **else** $i \to$i+1

6 : **End loop**

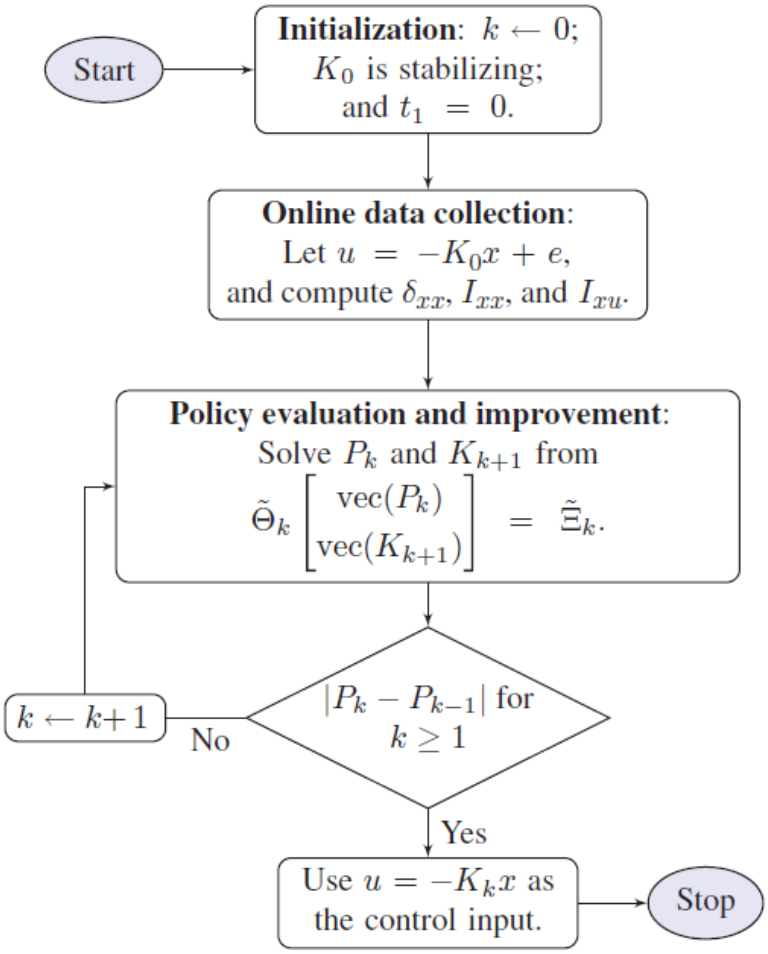

Figure 5.1: Flowchart of PI algorithm for finding on- line adaptive optimal controllers for continuous-time linear systems with completely unknown system dynamics, from [2].

The zero-sum game (ZSG) ADP variant has been suggested in [50]. The derivation follows the same procedure as was presented here for the LQR case. From this procedure,

$$\begin{aligned} x^T(t+T)P_i x(t+T) - x^T(t)P_i x(t) = &- \int_t^{t+T} x^T(Q_i - \gamma^2 H_i^T H_i)x \, d\tau \\ &- 2\int_t^{t+T} (u_1 + K_i x)^T K_{i+1} x \, d\tau \\ &+ 2\gamma^2 \int_t^{t+T} (u_2 + H_i x)^T H_{i+1} x \, d\tau \end{aligned} \tag{5.42}$$

and by Kronecker-product manipulations (in addition to (5.27), (5.28) and (5.29)),

$$\int_t^{t+T} \left(x^T \gamma^2 H_i^T H_i x\right) d\tau = \gamma^2 \int_t^{t+T} \left(x^T \otimes x^T\right) d\tau \; \mathrm{vec}(H_i^T H_i) \tag{5.43}$$

$$2\gamma^2 \int_t^{t+T} \left(u_2^T H_{i+1} x\right) d\tau = 2\gamma^2 \int_t^{t+T} \left(x^T \otimes u_2^T\right) d\tau \; \mathrm{vec}(H_{i+1}) \tag{5.44}$$

$$2\gamma^2 \int_t^{t+T} \left(u_2^T H_{i+1} x\right) d\tau = 2\gamma^2 \int_t^{t+T} \left(x^T \otimes u_2^T\right) d\tau \, (I_n \otimes H_i^T) \, \mathrm{vec}(H_{i+1}) \tag{5.45}$$

For the case where $x \in \mathbb{R}^n$, $u_1 \in \mathbb{R}^{m_1}$, and $u_2 \in \mathbb{R}^{m_2}$, the number of collected data sets l must satisfy the following condition,

$$l \geq \frac{n(n+1)}{2} + mn + mq \tag{5.46}$$

Similar to what was shown before, we define,

$$\delta_i = [x(t_{i+1}) \otimes x(t_{i+1}) - x(t_i) \otimes x(t_i)] \tag{5.47}$$

$$\delta_{xx} = [\delta_1, \delta_2, \cdots, \delta_{l-1}]^T \tag{5.48}$$

$$I_{xx} = \left[\int_{t_0}^{t_1} x \otimes x \, d\tau, \quad \int_{t_1}^{t_2} x \otimes x \, d\tau, \quad \cdots, \quad \int_{t_{l-1}}^{t_l} x \otimes x \, d\tau \right]^T \tag{5.49}$$

$$I_{xu_1} = \left[\int_{t_0}^{t_1} x \otimes u_1 \, d\tau, \quad \int_{t_1}^{t_2} x \otimes u_1 \, d\tau, \quad \cdots, \quad \int_{t_{l-1}}^{t_l} x \otimes u_1 \, d\tau \right]^T \tag{5.50}$$

$$I_{xu_2} = \left[\int_{t_0}^{t_1} x \otimes u_2 \, d\tau, \quad \int_{t_1}^{t_2} x \otimes u_2 \, d\tau, \quad \cdots, \quad \int_{t_{l-1}}^{t_l} x \otimes u_2 \, d\tau \right]^T \tag{5.51}$$

Based on these definitions, we can write the following system of equations,

$$X_i \Theta_i = Y_i$$

where,

$$X_i = \left[-\delta_{xx} N, \quad 2I_{xu}(I_n \otimes R) + 2I_{xx}(I_n \otimes K_i^T R), -2\gamma^2 I_{xu_2} - 2\gamma^2 I_{xu_2}(I_n \otimes H_i^T) \right] \tag{5.52}$$

$$Y_i = I_{xx} \, \mathrm{vec}(Q_i - \gamma^2 H_i^T H_i) \tag{5.53}$$

$$\Theta_i = \begin{bmatrix} \mathrm{svec}(P_i) \\ \mathrm{vec}(K_{i+1}) \\ \mathrm{vec}(H_{i+1}) \end{bmatrix} \tag{5.54}$$

that is solved in the least-squares sense by,

$$\Theta_i = (X_i^T X_i)^{-1} X_i^T Y_i$$

The algorithm is formulated as follows,

Model-Free PI for ZSG Using State-Feedback

Input: Design performance (Q, R, γ), convergence criterion ϵ, and a stabilizing initial policy K_0.

Output: Gain matrices K and H (near optimal), and the corresponding P matrix.

1: **Initialize:** Construct excitation initial inputs $u_1^0 = -K_0 x + \nu_1$ and $u_2^0 = H_0 x + \nu_2$ where ν_1, ν_2 are exploration noise signals. Set $i \to 0$.

2: **Acquire Data.** Apply u_1^0 and u_2^0 during $t \in [t_0, t_l]$, and collect input-state (u_1, u_2, x) data sets, divided into the time intervals $t_i - t_{i-1} = T_i, i = 1, 2, ..., l$.

3: **Loop**

4: **Policy Evaluation and Policy Improvement:** Solve for P_i, K_{i+1} and H_{i+1} in (5.42).

5: **If** $||P_i - P_{i-1}|| < \epsilon$ **then**

return P_i, K_{i+1}, H_{i+1}

6: **else** $i \to$i+1

6 : **End loop**

Note that only the gain matrix K is required for control (i.e., to be applied in u_1). The gain matrix H represents the worst-case disturbance, and it is only used in the design phase. The anticipated actual disturbance (u_2) is uncertain.

5.4 Model-Free PI using Output Feedback

In most practical systems, measuring all the state variables directly is not feasible. To address this limitation, [11] proposed a method that circumvents the need for full state information.

Their approach involves parameterizing the internal state in terms of filtered inputs and outputs of the system.

5.5 A filtered input–output state parametrization

Consider the following observable system (4.1), $\begin{aligned} \dot{x} &= Ax + Bu, \\ y &= Cx \end{aligned}$

where,

$$x \in \mathbb{R}^n, \quad u \in \mathbb{R}^m, \quad y \in \mathbb{R}^p.$$

An observer with a desired characteristic polynomial can always be designed; it has the following structure,

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x}). \tag{5.55}$$

The estimation error $e = x - \hat{x}$ obeys,

$$\dot{e} = (A - LC)e \tag{5.56}$$

where

$$\det(sI - (A - LC)) = s^n + \alpha_{n-1}s^{n-1} + \alpha_{n-2}s^{n-2} + \cdots + \alpha_1 s + \alpha_0 = \Lambda(s) \tag{5.57}$$

is the desired characteristic polynomial of the observer.

Let,

$$\hat{x} = \begin{bmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_n \end{bmatrix}, \quad u = \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ \vdots \\ y_p \end{bmatrix}, \tag{5.58}$$

and rewrite the observer as,

$$\dot{\hat{x}} = (A - LC)\hat{x} + Bu + Ly \tag{5.59}$$

This is a linear time-invariant system with two inputs, u and y. The system poles are the roots of the $\Lambda(s)$ in (5.57). Hence, a transfer function can be formulated from any element of u (or

y) to any element in $\hat{x}$. All of these transfer functions will have $\Lambda(s)$ as their characteristic polynomial (i.e., their denominator), but their numerator could be different (or individual). Additionally, because of the structure of (5.59), which characterizes a causal system, all these transfer functions are strictly proper (i.e., they have more poles than zeros). Based on these observations, the transfer function from u_i to $\hat{x}_1$ is,

$$\begin{aligned}\frac{\hat{x}_1(s)}{u_i(s)} &= \frac{a^1_{i,n-1}s^{n-1} + a^1_{i,n-2}s^{n-2} + \cdots + a^1_{i,1}s + a^1_{i,0}}{s^n + \alpha_{n-1}s^{n-1} + \alpha_{n-2}s^{n-2} + \cdots + \alpha_1 s + \alpha_0} \\ &= \frac{1}{\Lambda(s)} \begin{bmatrix} a^1_{i,n-1} & a^1_{i,n-2} & \cdots & a^1_{i,1} & a^1_{i,0} \end{bmatrix} \begin{bmatrix} s^{n-1} \\ s^{n-2} \\ \vdots \\ s \\ 1 \end{bmatrix}\end{aligned} \tag{5.60}$$

By defining a basis of linear filters,

$$F_0(s) = \frac{1}{\Lambda(s)}, \quad F_1(s) = \frac{s}{\Lambda(s)}, \quad \ldots, \quad F_{n-2}(s) = \frac{s^{n-2}}{\Lambda(s)}, \quad F_{n-1}(s) = \frac{s^{n-1}}{\Lambda(s)} \tag{5.61}$$

The transfer function from u_i to $\hat{x}_1$ can also be represented as,

$$\begin{aligned}\frac{\hat{x}_1(s)}{u_i(s)} &= \begin{bmatrix} a^1_{i,n-1} & a^1_{i,n-2} & \cdots & a^1_{i,1} & a^1_{i,0} \end{bmatrix} \begin{bmatrix} F_{n-1}(s) \\ F_{n-2}(s) \\ \vdots \\ F_1(s) \\ F_0(s) \end{bmatrix} \\ &= \begin{bmatrix} a^1_{i,n-1} & a^1_{i,n-2} & \cdots & a^1_{i,1} & a^1_{i,0} \end{bmatrix} F(s)\end{aligned} \tag{5.62}$$

where $F(s)$ is a vector containing all the basis filters. Continuing with the same notation, the

response of $\hat{x}$ due to u is,

$$\begin{bmatrix} \hat{x}_1(s) \\ \hat{x}_2(s) \\ \vdots \\ \hat{x}_n(s) \end{bmatrix} = \begin{bmatrix} a^1_{1,n-1} & a^1_{1,n-2} & \cdots & a^1_{1,1} & a^1_{1,0} \\ a^2_{1,n-1} & a^2_{1,n-2} & \cdots & a^2_{1,1} & a^2_{1,0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a^n_{1,n-1} & a^n_{1,n-2} & \cdots & a^n_{1,1} & a^n_{1,0} \end{bmatrix} \begin{bmatrix} F_{n-1}(s) \\ F_{n-2}(s) \\ \vdots \\ F_1(s) \\ F_0(s) \end{bmatrix} u_1(s)$$
$$+ \cdots + \begin{bmatrix} a^1_{m,n-1} & a^1_{m,n-2} & \cdots & a^1_{m,1} & a^1_{m,0} \\ a^2_{m,n-1} & a^2_{m,n-2} & \cdots & a^2_{m,1} & a^2_{m,0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a^n_{m,n-1} & a^n_{m,n-2} & \cdots & a^n_{m,1} & a^n_{m,0} \end{bmatrix} \begin{bmatrix} F_{n-1}(s) \\ F_{n-2}(s) \\ \vdots \\ F_1(s) \\ F_0(s) \end{bmatrix} u_m(s) \tag{5.63}$$

which can be compactly written as:

$$\hat{x}(s) = M^1_u u^1_F(s) + \cdots + M^m_u u^m_F(s) = \sum_{i=1}^{m} M^i_u u^i_F(s) \tag{5.64}$$

Adding also the influence of y (on $\hat{x}$) amounts to,

$$\hat{x}(s) = \sum_{i=1}^{m} M^i_u u^i_F(s) + \sum_{i=1}^{p} M^i_y y^i_F(s) \tag{5.65}$$

where

$$u^i_F(s) = F(s)u_i(s), \quad y^i_F(s) = F(s)y_i(s).$$

Reorganizing the expression of $\hat{x}(s)$ gives,

$$\hat{x}(s) = M_u u_F(s) + M_y y_F(s) = \begin{bmatrix} M_u & M_y \end{bmatrix} \begin{bmatrix} u_F(s) \\ y_F(s) \end{bmatrix}, \quad \hat{x}(t) = \mathcal{L}^{-1}\{\hat{x}(s)\} \tag{5.66}$$

where

$$M_u = \begin{bmatrix} M^1_u & \cdots & M^m_u \end{bmatrix}, \quad M_y = \begin{bmatrix} M^1_y & \cdots & M^p_y \end{bmatrix},$$

$$u_F(s) = \begin{bmatrix} u_F^1(s) \\ \vdots \\ u_F^m(s) \end{bmatrix}, \quad y_F(s) = \begin{bmatrix} y_F^1(s) \\ \vdots \\ y_F^p(s) \end{bmatrix}.$$

Then, $\hat{x}(t)$ (above) replaces $x(t)$ for the learning process after convergence of the observer. The convergence rate is determined by the design of $\Lambda(s)$.

Adding the influence of initial conditions, Eq. (5.66) can be rewritten in the time domain as,

$$\hat{x}(t) = M_u u_F(t) + M_y y_F(t) + e^{(A-LC)t}\hat{x}(0). \tag{5.67}$$

and the estimation error dynamics,

$$e(t) = x(t) - \hat{x}(t) = e^{(A-LC)t}e(0), \tag{5.68}$$

Because the proposed algorithm is based on collecting information from the system's response, it is crucial to ensure that the data reflects the steady-state behavior of the observer. Since $A - LC$ is Hurwitz, the term $e^{(A-LC)t}$ vanishes as $t \to \infty$, guaranteeing the convergence of the estimate $\hat{x}(t)$ to the true state $x(t)$. To ensure that transient effects do not corrupt the collected data, it is necessary to begin the data collection only after the observer dynamics have sufficiently settled. The rate at which the transient decays is determined by the characteristic polynomial $\Lambda(s)$, and specifically by the eigenvalues of $A - LC$. The dominant (slowest) time constant, defined as the inverse of the smallest (in magnitude) real part of the eigenvalues, provides a measure of how quickly the estimation error $e(t)$ converges to zero. Typically, waiting approximately 3τ to 5τ after initialization, where τ is the dominant time constant, ensures that the effect of the initial conditions becomes negligible. Using (5.67), the value function is approximated as,

$$V = x^T P x \cong \hat{x}^T P \hat{x} = \begin{bmatrix} u_F(t) \\ y_F(t) \end{bmatrix}^T [M_u \;\; M_y]^T P [M_u \;\; M_y] \begin{bmatrix} u_F(t) \\ y_F(t) \end{bmatrix} \triangleq z^T(t)\bar{P}z(t), \tag{5.69}$$

where $z(t)$ is an augmented vector, defined as $z(t) = \begin{bmatrix} u_F^T(t) & y_F^T(t) \end{bmatrix}^T \in \mathbb{R}^{mn+pn}$, and the matrix

$\bar{P}$, given by,

$$\bar{P} = \bar{P}^T = \begin{bmatrix} M_u^T P M_u & M_u^T P M_y \\ M_y^T P M_u & M_y^T P M_y \end{bmatrix}. \tag{5.70}$$

is composed of P (solution of), and the state parameterization coefficients M_u and M_y. In a model-free design scenario, all elements of $\bar{P}$ are unknown. By approximating also the state-feedback control law, using (5.67), it is reformulated as a dynamic output-feedback controller,

$$u(x) = Kx(t) \cong K\hat{x}(t) = K[M_u \;\; M_y] \begin{bmatrix} u_F(t) \\ y_F(t) \end{bmatrix} \triangleq \bar{K}z(t), \tag{5.71}$$

where $\bar{K} = K[M_u \;\; M_y] \in \mathbb{R}^{m\times(mn+pn)}$.

The unknown coefficients M_u and M_y are now embedded in $\bar{K}$. Consequently, the approximated optimal controller using the state parametrization would be,

$$u(z) = \bar{K}z(t), \tag{5.72}$$

with $\bar{K}$ and the associated $\bar{P}$ (given in (5.70)) with $P > 0$ that is the solution of the Riccati equation (4.12).

Based on (5.69) and (5.71), the learning equation (5.21) presented earlier for the model-free LQR design procedure can be reformulated for learning the dynamic output feedback controller (5.72); the result is,

$$\begin{aligned} & z^T(t)\bar{P}_i z(t) - z^T(t-T)\bar{P}_i z(t-T) = \\ & \quad - \int_{t-T}^{t} y^T(\tau) Q_y y(\tau)\, d\tau - \int_{t-T}^{t} z^T(\tau)\bar{K}_i^T R \bar{K}_i z(\tau)\, d\tau \\ & \quad - 2\int_{t-T}^{t} \big(u(\tau) - \bar{K}_i z(\tau)\big)^T R \bar{K}_{i+1} z(\tau)\, d\tau. \end{aligned} \tag{5.73}$$

This formulation includes the unknown matrices $\bar{P}_i$ and $\bar{K}_{i+1}$, which can be solved from system data (y, u) (and consequently z), via least-squares optimization, in a similar fashion to the state feedback case. Define,

$$\delta_{zz} = \begin{bmatrix} \bar{z}^T(t_1) - \bar{z}^T(t_0) & \bar{z}^T(t_2) - \bar{z}^T(t_1) & \cdots & \bar{z}^T(t_l) - \bar{z}^T(t_{l-1}) \end{bmatrix}^T \tag{5.74}$$

$$I_{zu} = \left[\int_{t_0}^{t_1} (z(\tau) \otimes u(\tau))\, d\tau \quad \int_{t_1}^{t_2} (z(\tau) \otimes u(\tau))\, d\tau \quad \cdots \quad \int_{t_{l-1}}^{t_l} (z(\tau) \otimes u(\tau))\, d\tau \right]^T \tag{5.75}$$

$$I_{zz} = \left[\int_{t_0}^{t_1} (z(\tau) \otimes z(\tau))\, d\tau \quad \int_{t_1}^{t_2} (z(\tau) \otimes z(\tau))\, d\tau \quad \cdots \quad \int_{t_{l-1}}^{t_l} (z(\tau) \otimes z(\tau))\, d\tau \right]^T \tag{5.76}$$

$$I_{yy} = \left[\int_{t_0}^{t_1} (y(\tau) \otimes y(\tau))\, d\tau \quad \int_{t_1}^{t_2} (y(\tau) \otimes y(\tau))\, d\tau \quad \cdots \quad \int_{t_{l-1}}^{t_l} (y(\tau) \otimes y(\tau))\, d\tau \right]^T . \tag{5.77}$$

Based on these definitions, we can write the following system of equations,

$$X_i \Theta_i = Y_i \tag{5.78}$$

where,

$$X_i = \left[\delta_{zz} N, \quad 2I_{zu}(I_n \otimes R) - 2I_{zz}(I_n \otimes K_i^T R) \right] \tag{5.79}$$

$$Y_i = I_{zz} \operatorname{vec}(Q_i) - I_{yy} \operatorname{vec}(Q_y) \tag{5.80}$$

$$\Theta_i = \begin{bmatrix} \operatorname{svec}(\bar{P}_i) \\ \operatorname{vec}(\bar{K}_{i+1}) \end{bmatrix} \tag{5.81}$$

The algorithm goes as follow:

Model-Free PI for LQR Using Output-Feedback

Input: Design performance Q, R, and $\Lambda(s)$, convergence criterion ϵ, and a stabilizing initial policy $\bar{K}_0$

Output: Gain matrix $\bar{K}$ (near optimal), and the corresponding $\bar{P}$ matrix.

1: **Initialize:** Construct an excitation initial input $u^0 = \bar{K}_0 + \nu$ where ν is exploration noise. Set $i \to 0$.

2: **Acquire Data.** Apply u_0 during $t \in [t_0, t_l]$ and collect input-output (u, y) data sets, divided into the time intervals $t_i - t_{i-1} = T_i, i = 1, 2, ..., l$.

3: **Loop**

4: **Policy Evaluation and Policy Improvement:** Solve for $\bar{K}_{i+1}$ and $\bar{P}_i$ in (5.73).

5: **If** $||\bar{P}_i - \bar{P}_{i-1}|| < \epsilon$ **then**

return $\bar{P}_i, \bar{K}_{i+1}$

6: **else** $i \rightarrow$i+1

6 : **End loop**

Since there are: $(mn+pn)(mn+pn+1)/2$ and $m(mn+pn)$ unknowns in $\bar{P}_i$ and $\bar{K}_{i+1}$, one needs,

$$\ell \geq \frac{(mn+pn)(mn+pn+1)}{2} + m(mn+pn) \tag{5.82}$$

data sets to solve (5.78).

5.6 Extension to the ZSG Design Problem

For the ZSG model-free output-feedback design problem, we refer to [12]. The system dynamics is observable, and (as in (4.30)) is given by,

$$\dot{x} = Ax + B_1u_1 + B_2u_2,$$
$$y = Cx$$

$$x \in \mathbb{R}^n, \quad u_1 \in \mathbb{R}^{m_1}, \quad u_2 \in \mathbb{R}^{m_2}, \quad y \in \mathbb{R}^p$$

where,

$$\hat{x} = \begin{bmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_n \end{bmatrix}, \quad u_1 = \begin{bmatrix} u_{1,1} \\ \vdots \\ u_{1,m_1} \end{bmatrix}, \quad u_2 = \begin{bmatrix} u_{2,1} \\ \vdots \\ u_{2,m_2} \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ \vdots \\ y_p \end{bmatrix}. \tag{5.83}$$

The state observer (which, for the learning process, also takes into account the influence of the second player u_2) is,

$$\dot{\hat{x}} = A\hat{x} + B_1u_1 + B_2u_2 + L(y - C\hat{x}) \tag{5.84}$$

Following the same process presented earlier, we get,

$$\hat{x}(s) = M_{u_1} u_{1,F}(s) + M_{u_2} u_{2,F}(s) + M_y y_F(s)$$

$$= \begin{bmatrix} M_{u_1} & M_{u_2} & M_y \end{bmatrix} \begin{bmatrix} u_{1,F}(s) \\ u_{2,F}(s) \\ y_F(s) \end{bmatrix} \tag{5.85}$$

where,

$$u_{1,F}(s) = \begin{bmatrix} u_{1,F}^1(s) \\ \vdots \\ u_{1,F}^{m_1}(s) \end{bmatrix}, \quad u_{2,F}(s) = \begin{bmatrix} u_{2,F}^1(s) \\ \vdots \\ u_{2,F}^{m_2}(s) \end{bmatrix}, \tag{5.86}$$

$$u_{1,F}^i(s) = F(s) u_{1,i}(s), \quad u_{2,F}^i(s) = F(s) u_{2,i}(s)$$

To express the cost function in terms of the system's input-output behavior, we apply the state parametrization defined earlier. Substituting this into the steady-state cost expression yields,

$$V = \begin{bmatrix} u_{F1} \\ u_{F2} \\ y_F \end{bmatrix}^T \begin{bmatrix} M_{u1} & M_{u2} & M_y \end{bmatrix}^T P \begin{bmatrix} M_{u1} & M_{u2} & M_y \end{bmatrix} \begin{bmatrix} u_{F1} \\ u_{F2} \\ y_F \end{bmatrix} \triangleq z^T \bar{P} z, \tag{5.87}$$

where $z = [u_{F1}^T \; u_{F2}^T \; y_F^T]^\top \in \mathbb{R}^N$ with $N = m_1 n + m_2 n + pn$, and $M = [M_{u1} \; M_{u2} \; M_y] \in \mathbb{R}^{n \times N}$. The matrix $\bar{P} = \bar{P}^T$ is defined as,

$$\bar{P} = \begin{bmatrix} M_{u_1}^T P M_{u_1} & M_{u_1}^T P M_{u_2} & W_{u_1}^T P M_y \\ M_{u_2}^T P M_{u_1} & M_{u_2}^T P M_{u_2} & M_{u_2}^T P M_y \\ M_y^T P M_{u_1} & M_y^T P M_{u_2} & M_y^T P M_y \end{bmatrix} \in \mathbb{R}^{N \times N}. \tag{5.88}$$

This representation enables the formulation of the two policies as output-feedback,

$$u_1 = -K \begin{bmatrix} M_{u1} & M_{u2} & M_y \end{bmatrix} \begin{bmatrix} u_{F1} \\ u_{F2} \\ y_F \end{bmatrix} \triangleq -\bar{K}z, \tag{5.89}$$

$$u_2 = H \begin{bmatrix} M_{u1} & M_{u2} & M_y \end{bmatrix} \begin{bmatrix} u_{F1} \\ u_{F2} \\ y_F \end{bmatrix} \triangleq \bar{H}z, \tag{5.90}$$

with $\bar{K} \in \mathbb{R}^{m_1 \times N}$ and $\bar{H} \in \mathbb{R}^{m_2 \times N}$.

Consequently, the augmented solution to the ZSG-ARE is $\bar{P}$, and from this solution, the approximated optimal policies are,

$$u_1 = -\bar{K}z, \tag{5.91}$$

$$u_2 = \bar{H}z. \tag{5.92}$$

The gain matrices above can be solved from the following learning equation (that is based on (5.73),

$$\begin{aligned} & z^T(t)\bar{P}_i z(t) - z^T(t-T)\bar{P}_i z(t-T) \\ &= -\int_{t-T}^{t} y^T(\tau)Q_y y(\tau)d\tau - \int_{t-T}^{t} z^T(\tau)(\bar{K}_i)^T(\bar{K}_i)z(\tau)d\tau \\ &\quad + \gamma^2 \int_{t-T}^{t} z^T(\tau)(\bar{H}_i)^T(\bar{H}_i)z(\tau)d\tau \\ &\quad - 2\int_{t-T}^{t} (u_1(\tau) - \bar{K}_i z(\tau))^T \bar{K}_{i+1} z(\tau)d\tau \\ &\quad + 2\gamma^2 \int_{t-T}^{t} (u_2(\tau) - \bar{H}_i z(\tau))^T \bar{H}_{i+1} z(\tau)d\tau \end{aligned} \tag{5.93}$$

This formulation includes the unknown matrices $\bar{P}_i$, $\bar{K}_{i+1}$ and $\bar{H}_{i+1}$, which can be solved from system data (y, u_1, u_2) (and consequently z), via least-squares optimization, in a similar fashion to the state feedback case.

Define,

$$\delta_{zz} = \begin{bmatrix} \bar{z}^T(t_1) - \bar{z}^T(t_0) & \bar{z}^T(t_2) - \bar{z}^T(t_1) & \cdots & \bar{z}^T(t_l) - \bar{z}^T(t_{l-1}) \end{bmatrix}^T$$

$$I_{zu_1} = \left[\int_{t_0}^{t_1} (z(\tau) \otimes u_1(\tau))\, d\tau \quad \int_{t_1}^{t_2} (z(\tau) \otimes u_1(\tau))\, d\tau \quad \cdots \quad \int_{t_{l-1}}^{t_l} (z(\tau) \otimes u_1(\tau))\, d\tau \right]^T \tag{5.94}$$

$$I_{zu_2} = \left[\int_{t_0}^{t_1} (z(\tau) \otimes u_2(\tau))\, d\tau \quad \int_{t_1}^{t_2} (z(\tau) \otimes u_2(\tau))\, d\tau \quad \cdots \quad \int_{t_{l-1}}^{t_l} (z(\tau) \otimes u_2(\tau))\, d\tau \right]^T \tag{5.95}$$

$$I_{zz} = \left[\int_{t_0}^{t_1} (z(\tau) \otimes z(\tau))\, d\tau \quad \int_{t_1}^{t_2} (z(\tau) \otimes z(\tau))\, d\tau \quad \cdots \quad \int_{t_{l-1}}^{t_l} (z(\tau) \otimes z(\tau))\, d\tau \right]^T$$

$$I_{yy} = \left[\int_{t_0}^{t_1} (y(\tau) \otimes y(\tau))\, d\tau \quad \int_{t_1}^{t_2} (y(\tau) \otimes y(\tau))\, d\tau \quad \cdots \quad \int_{t_{l-1}}^{t_l} (y(\tau) \otimes y(\tau))\, d\tau \right]^T$$

Based on these definitions, we can write the following system of equations (5.37):

$$X_i \Theta_i = Y_i$$

where:

$$X_i = \left[\delta_{zz} N, \quad 2I_{zu_1} - 2I_{zz}(I_n \otimes K_i^T R), -2\gamma^2 I_{zu_2} + 2\gamma^2 I_{zz}(I_n \otimes H_i^T) \right] \tag{5.96}$$

$$Y_i = I_{zz} \operatorname{vec}(Q_i) - I_{yy} \operatorname{vec}(Q_y)$$

$$\Theta_i = \begin{bmatrix} \operatorname{svec}(\bar{P}_i) \\ \operatorname{vec}(\bar{K}_{i+1}) \\ \operatorname{vec}(\bar{H}_{i+1}) \end{bmatrix} \tag{5.97}$$

The algorithm goes as follow:

Model-Free PI for ZSG Using Output-Feedback

Input: Design performance Q, R, γ, and $\Lambda(s)$, convergence criterion ϵ, and a stabilizing initial policy $\bar{K}_0$

Output: Gain matrices $\bar{K}, \bar{H}$ (near optimal), and the corresponding $\bar{P}$ matrix

1: **Initialize:** Choose stabilizing policies $u_1^0 = \bar{K}_0 x + \nu_1$ where ν_1 is the exploration noise and any policy $u_2^0 = \bar{H}_0 x + \nu_2$ where ν_2 is a different exploration noise , set $i \to 0$.

2: **Acquire Data.** Apply u_1^0 and u_2^0 during $t \in [t_0, t_l]$, and collect input-output (u_1, u_2, y) data sets, divided into the time intervals of length $t_i - t_{i-1} = T_i, i = 1, 2, ..., l$. Iterate for $i = 0, 1, ...$

3: **Loop**

4: **Policy Evaluation and Policy Improvement:** Solve for $\bar{K}_{i+1}$, $\bar{H}_{i+1}$ and $\bar{P}_i$ in (5.93).

5: **If** $||\bar{P}_i - \bar{P}_{i-1}|| < \epsilon$ **then**

return $\bar{P}_i, \bar{K}_{i+1}$ and $\bar{H}_{i+1}$

6: **else** $i \rightarrow$i+1

6 : **End loop**

Since $N = m_1 n + m_2 n + pn$, there are $N(N+1)/2 + (m_1 + m_2)N$ unknowns in $\bar{P}_i$, $\bar{K}_{i+1}$ and $\bar{H}_{i+1}$ so we need:

$$l \geq N(N+1)/2 + (m_1 + m_2)N \tag{5.98}$$

data sets.

Chapter 6

Model Free Reduced Dimension Output Feedback

Throughout the previous chapter, we reviewed several recently developed algorithms, including model-free dynamic output feedback, which were employed in this study to design an optimal controller for a mixed platoon. During the application of the last presented approach, i.e., model-free dynamic output-feedback, we encountered an inherent challenge related to the parameterization process. To illustrate this issue, we present an example adapted from [11]. The solution presented later in the chapter is general (i.e., not limited to the platoon dynamics) and applicable to both LQR and ZSG.

6.1 Example - load frequency control model of power systems

Given the load frequency control model of a power system (taken from [11], and presented here to support the discussion). Although power systems are inherently nonlinear, under typical operating conditions, a linear model can be utilized to design optimal controllers. The primary challenge is accurately identifying the plant parameters required for controller design. This limitation motivates the adoption of model-free optimal control strategies. Therefore, we employ model-free PI using the output-feedback algorithm, given that the system is open-loop stable.

Under normal conditions, the model can be linearized,

$$A = \begin{bmatrix} -0.0665 & 8 & 0 & 0 \\ 0 & -3.663 & 3.663 & 0 \\ -6.86 & 0 & -13.736 & -13.736 \\ 0.6 & 0 & 0 & 0 \end{bmatrix} \tag{6.1}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 13.736 \\ 0 \end{bmatrix} \tag{6.2}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \tag{6.3}$$

The authors [11] choose the performance index parameters as $Q_y = 1$ and $R = 1$. All observer (5.59) poles are placed at -1, which means, $\Lambda(s) = (s+1)^4$. The optimal controller obtained by solving the ARE (4.12) is,

$$P^* = \begin{bmatrix} 0.3135 & 0.2864 & 0.0509 & 0.1912 \\ 0.2864 & 0.4156 & 0.0903 & 0.0789 \\ 0.0509 & 0.0903 & 0.0210 & 0 \\ 0.1912 & 0.0789 & 0 & 1.1868 \end{bmatrix} \tag{6.4}$$

$$K^* = \begin{bmatrix} 0.6994 & 1.2404 & 0.2890 & 0 \end{bmatrix} \tag{6.5}$$

and from the state parameterization, one gets,

$$M_u = \begin{bmatrix} 0 & 402.5 & 0 & 0 \\ 0 & -673.9 & 50.3 & 0 \\ 0 & 1918.5 & -133.6 & 13.7 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{6.6}$$

$$M_y = \begin{bmatrix} -240.5 & -200.4 & -45.5 & -13.5 \\ 404.3 & 312.1 & 61.2 & 23.6 \\ -1151.1 & -893.7 & -185.1 & -72.2 \\ 1.9 & 3.5 & 2.4 & 0.6 \end{bmatrix} \tag{6.7}$$

where, for this case study, because of the scalar input u and the scalar measurement y, the dimensions of the unknown extended matrices $\bar{K}$ and $\bar{P}$ are,

$$\begin{aligned} & u \in \mathbb{R}^{1\times 1}, \quad y \in \mathbb{R}^{1\times 1}, \quad x \in \mathbb{R}^{4\times 1}, \quad M_{u/y} \in \mathbb{R}^{4\times 4} \\ & \Rightarrow \\ & \bar{K} \in \mathbb{R}^{1\times 8}, \quad \bar{P} = \bar{P}^T \in \mathbb{R}^{8\times 8} \end{aligned}$$

These dimensions determine the number of unknown (scalars) to be solved from the data, and consequently, the number of learning equations needed.

Since there are $\frac{(1\cdot 4+1\cdot 4)(1\cdot 4+1\cdot 4+1)}{2} = 36$ unknown parameters in the symmetric matrix $\bar{P}$, and an additional $1\cdot(1\cdot 4+1\cdot 4) = 8$ unknowns in $\bar{K}$, the total number of unknown scalars to be solved is 44. Therefore, in order to fully solve the system (5.37), we require at least 44 independent data sets (i.e., data of 44 time sections that can generate 44 independent equations).

For comparison, in the case of full-state feedback, the dimensions of the optimal matrices are $P^* \in \mathbb{R}^{4\times 4}$ and $K^* \in \mathbb{R}^{1\times 4}$. Therefore, the number of data sets required to solve for all unknowns is at least (5.32):

$$l \geq \frac{4(4+1)}{2} + 1\cdot 4 = 10 + 4 = 14. \tag{6.8}$$

So, in the case of full-state feedback, we need 14 data sets to solve Eq. (5.37).

Again, we emphasize that more than three times the number of unknowns is typically required to ensure accurate and robust estimation from data. To demonstrate the difficulty inherent in the method presented in [11], we consider now the case of a second measurement (this has not been shown in [11]). In particular, we assume that both x_1 and x_2 are measured. In this case,

the output matrix C takes the form,

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{6.9}$$

According to the parametrization method described in chapter 5.6, we need to add one more matrix of unknown coefficients, $M_{y2} \in \mathbb{R}^{4\times 4}$. By updating the dimensions of the matrices of the design problem, we have,

$$\begin{aligned} &u \in \mathbb{R}^{1\times 1}, \quad y \in \mathbb{R}^{1\times 1}, \quad x \in \mathbb{R}^{4\times 1}, \quad M_{u/y_1/y_2} \in \mathbb{R}^{4\times 4} \\ &\bar{K} \in \mathbb{R}^{1\times 12}, \quad \bar{P} = \bar{P}^T \in \mathbb{R}^{12\times 12} \end{aligned}$$

Hence, to fully solve the system (5.37), we require at least (5.82),

$$\ell \geq \frac{(1\cdot 4 + 2\cdot 4)(1\cdot 4 + 2\cdot 4 + 1)}{2} + 1\cdot(1\cdot 4 + 2\cdot 4) = 90 \tag{6.10}$$

independent equations (each is generated from a dataset on a certain time section). From this particular example, we see that adding a second measurement (which generally enhances the available information) more than doubles the number of unknown variables. As a result, a larger amount of data is required to maintain a solvable system of equations. This makes the design problem more complex, and from our experience, it can influence the feasibility of the solution. In general, as we add more measurements, the problem complexity grows intensively. This is counterintuitive because adding more measurements makes the design closer to the state-feedback design problem, where the number of unknowns to solve is the lowest.

6.2 Model Free Reduced Dimension Output Feedback

The more unknowns there are, the greater the computational burden is. All system modes need to be excited by the injected exploration noise. Suppose that the exploration noise is insufficient, in the sense that some of the equations in the system of linear equations (5.37) are dependent, the system rank will be deficient. This problem will get worse if the number of measurements increases. We offer a solution to this problem based on the pole placement

method. Specifically, by applying a linear combination of original outputs, while ensuring that the system remains observable, we effectively reduce the dimension of the output matrix C, and the unknowns to be solved.

This approach is hereafter demonstrated for the LQR case, but also holds for the $\mathcal{H}_\infty$ (ZSG) controller. In the next chapter, this method is applied to the control problem of a mixed platoon, using $\mathcal{H}_\infty$ control formulation. As mentioned in Chapter 5.4 (5.55), to understand the method, we would like to briefly review what a state estimator is, and its design based on pole placement. For the LQR design, the system model is (4.1),

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

The observer equation is given by,

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x}) \tag{6.11}$$

It is desirable to choose a gain matrix L such that the estimation error,

$$e = x - \hat{x} \tag{6.12}$$

converge to zero. The estimation error dynamics obey,

$$\begin{aligned} \dot{e} &= \dot{x} - \dot{\hat{x}} \\ &= Ax + Bu - (A\hat{x} + Bu + L(y - C\hat{x})) \\ &= A(x - \hat{x}) - L(Cx - C\hat{x}) \\ &= (A - LC)e \end{aligned} \tag{6.13}$$

Hence, the error dynamics is governed by the matrix $A - LC$. To guarantee convergence of the error to zero, we design L such that the eigenvalues of $A - LC$ are all in the left-half complex plane, and the characteristic polynomial of $A - LC$ is $\Lambda(s)$, a desired characteristic polynomial. However, in the case of multiple measurements (i.e., y is not scalar), this design is redundant,

and there is certain freedom in the design of L. Let,

$$C \in \mathbb{R}^{a\times b}, \quad L \in \mathbb{R}^{b\times a}$$

then, the gain matrix L,

$$L = \begin{bmatrix} L_{11} & L_{12} & \cdots & L_{1a} \\ L_{21} & L_{22} & \cdots & L_{2a} \\ \vdots & \vdots & \ddots & \vdots \\ L_{b1} & L_{b2} & \cdots & L_{ba} \end{bmatrix} \tag{6.14}$$

The corresponding characteristic polynomial of the closed-loop system is given by:

$$\begin{aligned} \det(sI - A + LC) = \det \Bigg(& \begin{bmatrix} s_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & s_{bb} \end{bmatrix} - \begin{bmatrix} A_{11} & \cdots & A_{1b} \\ \vdots & \ddots & \vdots \\ A_{b1} & \cdots & A_{bb} \end{bmatrix} \\ & + \begin{bmatrix} L_{11} & \cdots & L_{1a} \\ \vdots & \ddots & \vdots \\ L_{b1} & \cdots & L_{ba} \end{bmatrix} \begin{bmatrix} C_{11} & \cdots & C_{1b} \\ \vdots & \ddots & \vdots \\ C_{a1} & \cdots & C_{ab} \end{bmatrix} \Bigg) \end{aligned} \tag{6.15}$$

Consider the desired polynomial:

$$\Lambda_\alpha(s) = (s-\lambda_1)(s-\lambda_2)\cdots(s-\lambda_b) = s^b + \alpha_{b-1}s^{b-1} + \cdots + \alpha_1 s + \alpha_0 \tag{6.16}$$

There exists a unique solution subspace for the gain matrix L that ensures the desired pole placement and achieves the required performance, although the solution is not necessarily unique element-wise.

For the multi-measurement case, by comparing the characteristic polynomial of the closed-loop system with the desired one, it becomes evident that infinitely many solutions exist. This redundancy can be exploited either to satisfy additional design criteria or by arbitrarily selecting some of the free parameters and solving for the rest.

We proceed with the example presented earlier (in this chapter). The relevant matrices were

defined by,

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad L = \begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \\ L_{31} & L_{32} \\ L_{41} & L_{42} \end{bmatrix} \tag{6.17}$$

where $C \in \mathbb{R}^{2\times 4}$ and $L \in \mathbb{R}^{4\times 2}$. The result of the matrix product is,

$$LC = \begin{bmatrix} L_{11} & L_{12} & 0 & 0 \\ L_{21} & L_{22} & 0 & 0 \\ L_{31} & L_{32} & 0 & 0 \\ L_{41} & L_{42} & 0 & 0 \end{bmatrix} \tag{6.18}$$

The characteristic polynomial of the closed-loop system is defined as the determinant of the matrix $(sI - A + LC)$, which gives,

$$\begin{aligned} \det(sI - A + LC) = \det \Bigg(& \begin{bmatrix} s_1 & 0 & 0 & 0 \\ 0 & s_2 & 0 & 0 \\ 0 & 0 & s_3 & 0 \\ 0 & 0 & 0 & s_4 \end{bmatrix} - \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix} \\ & + \begin{bmatrix} L_{11} & L_{12} & 0 & 0 \\ L_{21} & L_{22} & 0 & 0 \\ L_{31} & L_{32} & 0 & 0 \\ L_{41} & L_{42} & 0 & 0 \end{bmatrix} \Bigg) \end{aligned} \tag{6.19}$$

This is the actual characteristic polynomial that is a function of L; it should be compared to $\Lambda_\alpha(s)$. There are eight unknowns but only four constraints; hence, the design problem is redundant. At this stage, we incorporate additional constraints into the gain matrix L in order to enforce specific design requirements,

$$L_{12} = cL_{11}, \quad L_{22} = cL_{21}, \quad L_{32} = cL_{31}, \quad L_{42} = cL_{41} \tag{6.20}$$

Then the matrix L becomes,

$$L = \begin{bmatrix} L_{11} & cL_{11} \\ L_{21} & cL_{21} \\ L_{31} & cL_{31} \\ L_{41} & cL_{41} \end{bmatrix}$$

which can be reformulated as,

$$L = \begin{bmatrix} L_{11} \\ L_{21} \\ L_{31} \\ L_{41} \end{bmatrix} \cdot \begin{bmatrix} 1 & c \end{bmatrix}$$

From the result above, we define,

$$L_{\text{new}} = \begin{bmatrix} L_{11} \\ L_{21} \\ L_{31} \\ L_{41} \end{bmatrix} \qquad C_{\text{new}} = \begin{bmatrix} 1 & c \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & c & 0 & 0 \end{bmatrix} \tag{6.21}$$

We now perform the multiplication,

$$L_{\text{new}} C_{\text{new}} = \begin{bmatrix} L_{11} & cL_{11} & 0 & 0 \\ L_{21} & cL_{21} & 0 & 0 \\ L_{31} & cL_{31} & 0 & 0 \\ L_{41} & cL_{41} & 0 & 0 \end{bmatrix} \tag{6.22}$$

That is identical to (6.18) with the added constraints (6.20). As a result, the output matrix C can be replaced with a reduced-dimensional form represented by a single row vector C_{new}. Observing (6.21), reveals that the constraints chosen in (6.20) are equivalent to designing a new scalar output, y_{new} that is a linear combination of the elements of the original system outputs y. For this case, if,

$$y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \tag{6.23}$$

then,

$$y_{\text{new}} = y_1 + cy_2 = \begin{bmatrix} 1 & c \end{bmatrix} y = c_y y \tag{6.24}$$

This new output is a linear combination of the original measured output y that is determined by the combination vector c_y. The resulting new output y_{new} is a scalar; hence, the number of unknowns to solve during the learning process is kept to a minimum. Most importantly, the solution of the redundancy is predetermined by the choice of y_{new}, and it is kept consistent during the iterative learning process. This consistency plays an important role in the stability of the learning process, which is critical for the convergence to the near-optimal policy. The reduced dimension output feedback that has been presented here through the simple example can be extended to any multi-output system of any dimension. The choice of c_y allows for emphasis on measurements with a higher fidelity, by weighting them in y_{new} with larger weights (chosen in c_y). An important assumption is that the system maintains its observability also with new output y_{new}. However, this cannot be guaranteed without precise knowledge of the model. A relevant comment and a possible solution are presented at the end of this chapter.

6.3 Numerical Example: Load Frequency Control Model with a Reduced-Dimension Output Matrix

Given the load frequency control model (6.1), (6.2) and the reduced dimension output matrix C (6.21) with $c = 1$ (i.e., $y = y_1 + y_2$), then,

$$C_{\text{new}} = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}$$

and A and B are given in (6.1) and (6.2). As before, the performance index is chosen with $Q_y = 1$ and $R = 1$, and all eigenvalues of the observer are placed at -1. The optimal control

parameters, as obtained by solving the ARE (4.12) with respect to y_{new} are,

$$P^* = \begin{bmatrix} 0.2335 & 0.3082 & 0.0524 & 0.2234 \\ 0.3082 & 0.5207 & 0.1059 & 0.0908 \\ 0.0524 & 0.1059 & 0.0242 & 0 \\ 0.2234 & 0.0908 & 0 & 1.2244 \end{bmatrix} \tag{6.25}$$

$$K^* = \begin{bmatrix} 0.7198 & 1.4547 & 0.3326 & 0 \end{bmatrix} \tag{6.26}$$

$$M_u = \begin{bmatrix} 0 & -335.5579 & 0 & 0 \\ 0 & 741.4236 & 50.3150 & 0 \\ 0 & -2375.9 & -133.7337 & 13.7360 \\ 1.0000 & 25.0833 & 0 & 0 \end{bmatrix} \tag{6.27}$$

$$M_y = \begin{bmatrix} 200.6667 & 148.0588 & 30.1513 & 14.6691 \\ -441.1786 & -348.4302 & -75.6233 & -28.1346 \\ 1413.8 & 1100.9 & 224.5996 & 85.3686 \\ -13.0493 & -7.5694 & 0.0945 & -0.4985 \end{bmatrix} \tag{6.28}$$

It is important to note that after the parametrization process, even though an additional measurement was incorporated, the dimensions of M_y remained unchanged. Therefore, it was not necessary to introduce separate matrices such as M_{y1} and M_{y2}, and the parametrization structure was preserved. Consequently, the dimensions of the matrices $\bar{K}$ and $\bar{P}$ remained the same.

For the learning process, the initial stabilizing controller was taken to be $0.1K^*$, where K^* denotes the optimal gain (this choice is only for the demonstration). It is worth noting that the controller could have alternatively been initialized with zero control, since the system is open-loop stable. A total of 400 data-set intervals were selected to ensure sufficiently rich information for learning and convergence.

Table 6.1 presents a detailed comparison between the estimated gain vectors over the first six iterations of the learning process and the optimal gain vector, denoted by $\bar{K}^*$, that is computed from the model. The relative error for each component is computed. As observed, the estimates

converge quickly towards the optimal values, with negligible errors for most components after the fourth iteration.

Component	$\bar{K}^*$	$\bar{K}_1$	$\bar{K}_2$	$\bar{K}_3$	$\bar{K}_4$	$\bar{K}_5$	$\bar{K}_6$
1	0	0	0	0	0	0	0
2	46.7494	23.4114	30.6204	42.4401	46.4258	46.6296	46.6299
3	28.7112	65.5369	38.6087	30.0634	28.7490	28.7046	28.7045
4	4.5687	14.7324	7.8550	5.1391	4.5917	4.5691	4.5691
5	-27.0898	-12.3877	-17.2913	-24.5022	-26.8966	-27.0185	-27.0187
6	-34.1301	-42.4273	-32.1755	-32.9701	-34.0103	-34.0747	-34.0748
7	-13.6016	-27.5792	-16.9903	-13.9888	-13.6010	-13.5902	-13.5902
8	-1.9740	-3.0981	-2.1331	-1.9668	-1.9687	-1.9696	-1.9696

Relative Error	Error1	Error2	Error3	Error4	Error5	Error6
1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2	0.4992	0.3450	0.0922	0.0069	0.0026	0.0026
3	1.2826	0.3447	0.0471	0.0013	0.0002	0.0002
4	2.2246	0.7193	0.1248	0.0050	0.0001	0.0001
5	0.5427	0.3617	0.0955	0.0071	0.0026	0.0026
6	0.2427	0.0570	0.0339	0.0035	0.0016	0.0016
7	1.0271	0.2488	0.0285	0.0000	0.0008	0.0008
8	0.5701	0.0805	0.0036	0.0027	0.0022	0.0022

Table 6.1: Comparison of K^* and K_i over six iterations in Example 1 (load frequency control model with reduced-dimension output-feedback). The first row was omitted due to negligible numerical values.

The graph in Figure 6.1 shows the convergence of the parameters of the reduced-dimension output feedback gains, measured by the normalized difference between the result of each iteration and the optimal result from the ARE.

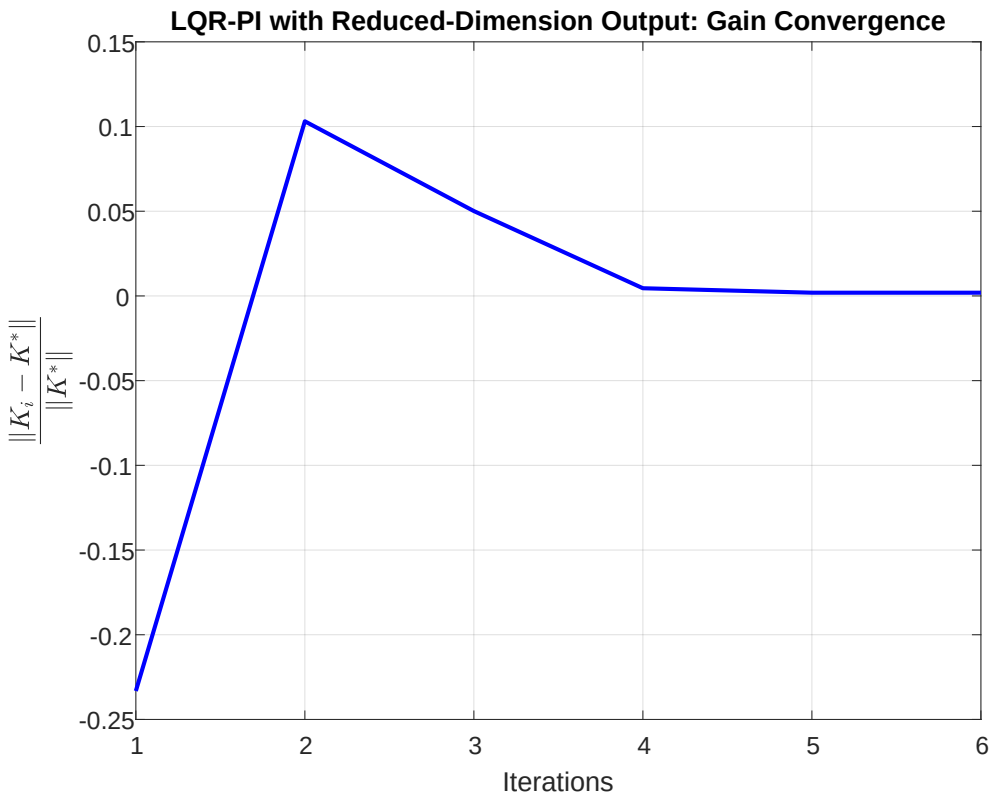

Figure 6.1: Convergence of the normalized error of K_i and the optimal K^*, in Example 1: load frequency control model with reduced-dimension output-feedback.

6.4 Example 2: Double Integrator System with a Reduced-Dimension Output Matrix

We now present an additional example, adapted from [11], involving an unstable double integrator system. This example further illustrates the applicability and performance of the proposed approach in scenarios where the open-loop dynamics are inherently unstable. Consider the system,

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}. \tag{6.29}$$

The performance index weights are $Q_y = 1$ and $R = 1$. The eigenvalues of the observer are placed at -2. The optimal controller as obtained by solving the ARE (4.12) is,

$$P^* = \begin{bmatrix} 1.4142 & 1.0000 \\ 1.0000 & 1.4142 \end{bmatrix} \tag{6.30}$$

$$K^* = \begin{bmatrix} 1.0000 & 1.4142 \end{bmatrix} \tag{6.31}$$

$$M_u = \begin{bmatrix} 1 & 0 \\ 4 & 1 \end{bmatrix} \tag{6.32}$$

$$M_y = \begin{bmatrix} 4 & 4 \\ 0 & 4 \end{bmatrix} \tag{6.33}$$

Datasets were collected over 99 intervals, with white noise as the exploration signal. The controller was initialized with 0.1*the optimal gain values, and the ending threshold for the learning process is $\epsilon = 10^{-3}$.

Table 6.2 shows the evolution of the gain vector elements in selected iterations alongside their relative errors with respect to the final gain K_{16}. The corresponding convergence trend is illustrated in Figure 6.2.

Elements	$\bar{K}^*$	$\bar{K}_1$	$\bar{K}_2$	$\bar{K}_4$	$\bar{K}_8$	$\bar{K}_{12}$	$\bar{K}_{16}$
1	6.6565	0.0000	4.2153	6.5647	6.5671	6.6611	6.6565
2	1.4141	1.0000	0.8178	1.4044	1.3947	1.4149	1.4141
3	4.0000	0.0000	3.7566	3.7891	3.9526	4.0047	4.0000
4	9.6567	-0.0000	7.0405	9.4064	9.5317	9.6648	9.6567

Relative Error	Error1	Error2	Error4	Error8	Error12	Error16
1	1.0000	0.3667	0.0138	0.0134	0.0007	0.0000
2	0.2928	0.4217	0.0069	0.0137	0.0006	0.0000
3	1.0000	0.0608	0.0527	0.0119	0.0012	0.0000
4	1.0000	0.2709	0.0259	0.0129	0.0008	0.0000

Table 6.2: Comparison of gain vector components and their relative error over selected iterations with respect to the final gain $\bar{K}^* = \bar{K}_{16}$, in Example 2: double integrator system.

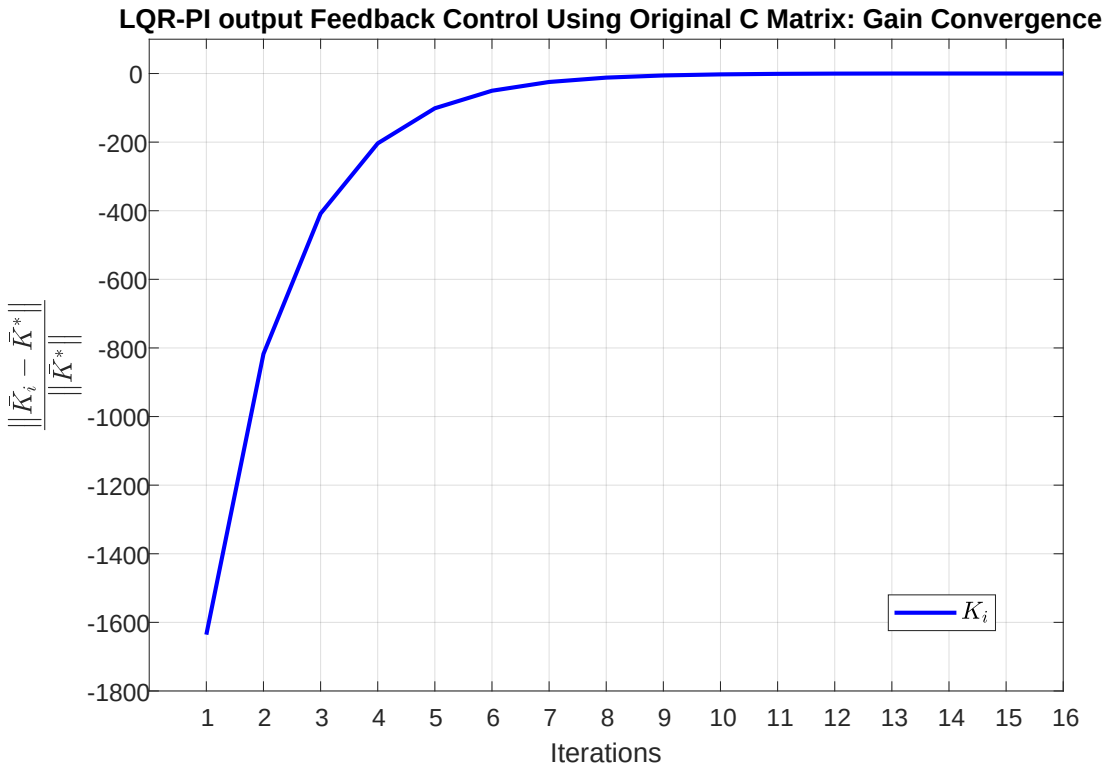

Figure 6.2: Convergence of the normalized parameters K_i towards the optimal gain K^*, in Example 2: double integrator system.

To explore the effect of an additional measurement (i.e., richer information), the output matrix C is modified to include separate measurements of both state variables. This represents a scenario in which the full state of the system is directly observable; hence, the output-feedback concept is only shown here for the demonstration of the order reduction method. The modified model is,

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{6.34}$$

and the product of L and C is,

$$\begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{6.35}$$

At this stage, structural constraints are imposed to cancel the redundancy. here we choose $y_{\text{new}} = y_1 + y_2$, then,

$$L_{11} = L_{12}, \quad L_{21} = L_{22} \tag{6.36}$$

what leads to,

$$\begin{bmatrix} L_{11} \\ L_{21} \end{bmatrix} \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{6.37}$$

and therefore:

$$C_{\text{new}} = \begin{bmatrix} 1 & 1 \end{bmatrix}. \tag{6.38}$$

Now, the reduced output system is,

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C_{new} = \begin{bmatrix} 1 & 1 \end{bmatrix}. \tag{6.39}$$

The performance index parameters are chosen as before. Solving the ARE (4.12) gives,

$$P^* = \begin{bmatrix} 0.7321 & 1.0000 \\ 1.0000 & 1.7321 \end{bmatrix} \tag{6.40}$$

$$K^* = \begin{bmatrix} 1.0000 & 1.7321 \end{bmatrix} \tag{6.41}$$

$$M_u = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{6.42}$$

$$M_y = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix} \tag{6.43}$$

Table 6.3 presents the evolution of the gain vector elements in selected iterations, along with their relative errors with respect to the final value in iteration K_{15}. In contrast to the previous example, with C based on a single measurement, the current case introduces additional

information into the learning process (by including also y_2). This enriched information case resulted in a slightly faster convergence toward the optimal gain values, which emphasizes the contribution of the additional measurement (if it is available). The trend observed in the table is consistent with the convergence behavior illustrated in Figure 6.3.

Component	$\bar{K}^*$	$\bar{K}_1$	$\bar{K}_2$	$\bar{K}_4$	$\bar{K}_8$	$\bar{K}_{12}$	$\bar{K}_{15}$
1	6.6565	0.0000	4.2153	6.5647	6.5671	6.6611	6.6565
2	1.4141	1.0000	0.8178	1.4044	1.3947	1.4149	1.4141
3	4.0000	0.0000	3.7566	3.7891	3.9526	4.0047	4.0000
4	9.6567	-0.0000	7.0405	9.4064	9.5317	9.6648	9.6567

Relative Error	Error1	Error2	Error4	Error8	Error12	Error15
1	1.0000	0.3667	0.0138	0.0134	0.0007	0.0000
2	0.2928	0.4217	0.0069	0.0137	0.0006	0.0000
3	1.0000	0.0608	0.0527	0.0119	0.0012	0.0000
4	1.0000	0.2709	0.0259	0.0129	0.0008	0.0000

Table 6.3: Comparison between gain vector components over selected iterations and their relative error with respect to the final gain vector $\bar{K}^* = \bar{K}_{15}$, in Example 2: double integrator system with reduced-dimension C matrix.

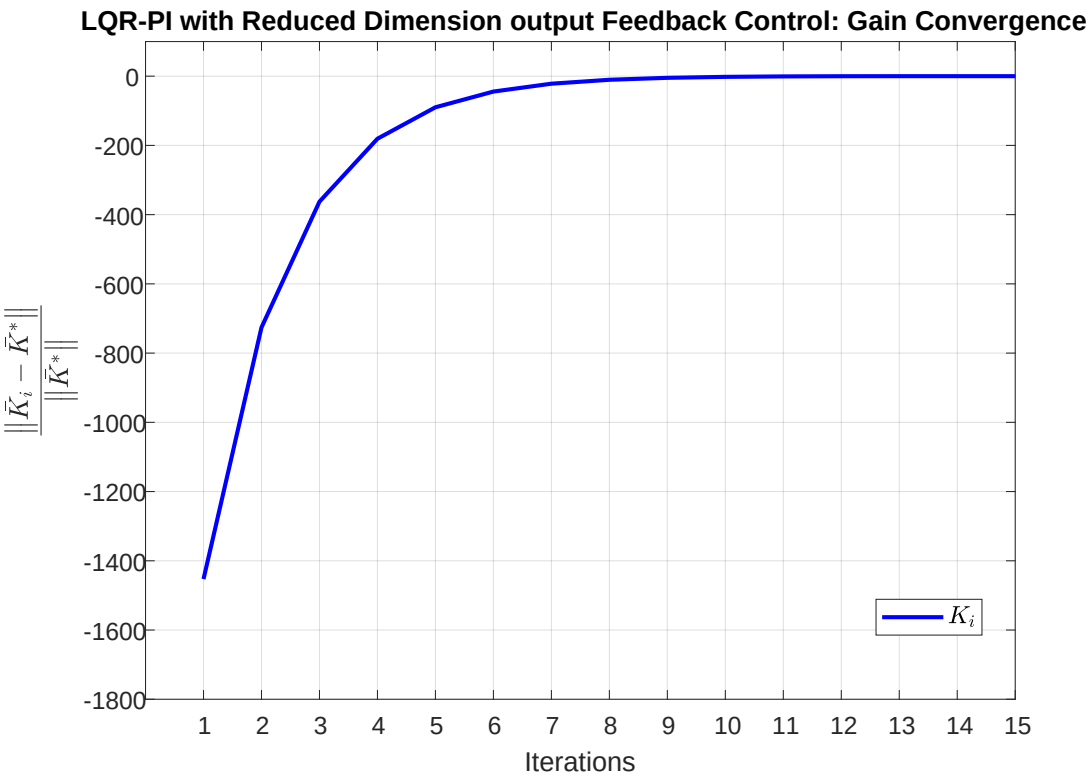

Figure 6.3: Convergence of the normalized parameters K_i towards the optimal gain K^*, in Example 2: double integrator system with reduced-dimension C matrix.

6.5 Orthogonality in the State Space and the Influence on Observability

From a theory by Popov, Belevitch, and Hautus (the PBH test), a linear system is unobservable if any of the eigenvectors v_i of A is orthogonal to C (i.e., $Cv_i = 0$). Hence, it is possible that for a system that is originally observable (based on the couple (A, C)), the definition of y_{new} as suggested in this chapter will cause an unobservable couple (A, C_{new}). For a model-based design, this situation can be easily avoided, but if the model matrix A is unknown, such a situation (although unlikely) is possible. Consequently, the system response (in the state space) is not fully spanned by C_{new} (i.e., some behavior is unseen by y_{new}). As to the state parameterization technique based on pole-placement (i.e., determining the eigenvalues of $A-LC$), a solution does not exist for an arbitrarily $\Lambda_\alpha(s)$ that represents the desired characteristic polynomial of the observer. Nevertheless, in the model-free variant (i.e., the output-feedback PI design), the observer gain, L, is not designed explicitly. Instead, the parametrization coefficients are embedded in $\bar{P}$ and $\bar{K}$, and the data is filtered by a basis of filters with $\Lambda_\alpha(s)$ as their denominator.

When running the algorithm, for example, in MATLAB, this situation typically manifests itself as a *rank deficiency* warning, indicating linear dependence among the generated equations. This rank deficiency can arise due to two main reasons (where in both, the data is not sufficiently reach):

1. Insufficient excitation of system modes due to poorly selected exploration noise.

2. The couple (A, C_{new}) is unobservable (i.e., not all system modes are observable through y_{new}).

To address this issue, system excitation should be enriched by adjusting exploration noise to better excite system dynamics. If this action has no influence on the rank of X_i (using the notation $X_i\Theta_i = Y_i$, as in (5.78)), alternative constraints on L should be selected (resulting in the generation of a new output matrix C_{new2}). The algorithm is then rerun using C_{new2}, and this procedure is repeated, modifying the excitation and adjusting the constraints, until the rank deficiency is resolved and the algorithm successfully converges.

Chapter 7

Simulation Results

In this chapter, we present the simulation results that illustrate the application of the reduced-dimension dynamic output-feedback method and the $\mathcal{H}_\infty$ control approach discussed in the preceding chapters. The analysis begins with a series of mixed platoon configurations, starting with a single HDV and incrementally increasing to a three-HDV platoon. Since the proposed control strategy is based on the $\mathcal{H}_\infty$ framework, it is assumed that the external disturbance originates from a CAV located at the front of the platoon, which is defined as the 'platoon-leader'. The actual controller of this 'platoon-leader' is not in the scope of this thesis (which focuses on the controller of the CAV in the back of the mixed-platoon unit).

In all scenarios, we used the PI learning algorithm, which requires an initial stabilizing controller to ensure safety throughout the learning process. This stabilizing controller guarantees that the system remains bounded during policy iterations, particularly during the early stages of adaptation. We emphasize that our main design goal is not to stabilize the mixed platoon, but to improve its string-stability performance, which we do by enforcing an $\mathcal{H}_\infty$ design criterion.

Following the simulation results, we assess whether weak string stability has been achieved under the proposed control framework. Weak string stability is a concept intended to characterize string stability for platoons where not all vehicles can be assured to be string stable (e.g., an HDV). Hence, by the notation of weak string stability, we demand disturbance attenuation only between CAVs. For the mixed platoons analyzed in this study, we require that the CAV at the back compensates for the string instability of the HDVs in front.

Finally, a comparative performance analysis is conducted between the standard LQR-based controller and the zero-sum game-based $\mathcal{H}_\infty$ approach.

7.1 Reduced-Dimension Control in Mixed Platoons

7.1.1 Case 1: One HDV in the platoon

In this scenario, a mixed platoon composed of one HDV followed by a CAV is considered, resulting in the following configuration: Leader–HDV–CAV. The leader determines the normal platoon velocity, and deviations from that velocity are represented as a disturbance input.

Based on the formulation provided in Chapter 3, we formulate the **error dynamics** of the mixed-platoon. The state vector is,

$$x = \begin{bmatrix} \Delta s_1 \\ e_{v,1} \\ e_{s,2} \\ e_{v,2} \end{bmatrix} \tag{7.1}$$

where $e_{s,i}$ and $e_{v,i}$ represent the spacing and velocity errors of vehicle i, respectively.

As developed in Chapter 3, the error dynamics is represented by (4.13),

$$\dot{x} = Ax + B_1 u_1 + B_2 u_2$$

with,

$$A = \begin{bmatrix} 0 & -1 & 0 & 0 \\ f_s & -(f_{\Delta v} + f_v) & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 \\ 0 \\ -t_h \\ 1 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 1 \\ f_{\Delta v} \\ 0 \\ 0 \end{bmatrix} \tag{7.2}$$

where:

- f_s f_v and $f_{\Delta v}$ are positive constants derived from an IDM model for a specific equilibrium point.
- t_h is the time headway.

- u_1 is the control input applied to the last vehicle, which is the CAV.
- u_2 represents an external disturbance (leader's velocity perturbations).

This model enables analysis of how disturbances (u_2) propagate through the platoon and how feedback control (u_1) can be designed to maintain desired inter-vehicle spacing and velocity synchronization. In particular, it provides the basis for studying **weak string stability** and the effect of mixed control schemes.

Substituting the values from Fig. 2.2 for $20[\frac{m}{s}]$: $f_s = 0.05$, $f_v = 0.08$, $f_{\Delta v} = 0.34$, $t_h = 0.5[s]$, into the system matrices defined in (7.2), gives,

$$A = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0.08 & -0.48 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 \\ 0 \\ -0.5 \\ 1 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 1 \\ 0.35 \\ 0 \\ 0 \end{bmatrix} \tag{7.3}$$

Based on the formulation in (4.39), the output matrix is constructed as follows,

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & -1 \end{bmatrix} \tag{7.4}$$

In accordance with the dimension of (7.4), the observer matrix gain L is,

$$L = \begin{bmatrix} L_{11} & L_{12} & L_{13} \\ L_{21} & L_{22} & L_{23} \\ L_{31} & L_{32} & L_{33} \\ L_{41} & L_{42} & L_{43} \end{bmatrix} \tag{7.5}$$

To reduce the dimension of y using the method of chapter 6, we apply the following constraints,

$$\begin{aligned} L_{12} &= L_{11}, & L_{13} &= \tfrac{3}{2}L_{11}, \\ L_{22} &= L_{21}, & L_{23} &= \tfrac{3}{2}L_{21}, \\ L_{32} &= L_{31}, & L_{33} &= \tfrac{3}{2}L_{31}, \\ L_{42} &= L_{41}, & L_{43} &= \tfrac{3}{2}L_{41}, \end{aligned} \tag{7.6}$$

that leads to,

$$LC = \begin{bmatrix} L_{11} \\ L_{21} \\ L_{31} \\ L_{41} \end{bmatrix} \begin{bmatrix} 1 & 1 & \frac{3}{2} \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & -1 \end{bmatrix} \tag{7.7}$$

hence,

$$, C_{\text{new}} = \begin{bmatrix} 1 & 1 & \frac{3}{2} \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & -\frac{5}{2} \end{bmatrix} \tag{7.8}$$

We conducted the simulation with,

$$R = \begin{bmatrix} 1 & 0 \\ 0 & -\gamma_{min}^2 \end{bmatrix} \tag{7.9}$$

where $\gamma_{min}^2 = 0.3$ **to enforce string stability**. The eigenvalues of the observer were chosen to be placed at -2. The optimal control solution obtained by solving the ZSG-ARE (4.26) is,

$$P^* = \begin{bmatrix} 0.004 & 0.006 & 0.005 & -0.036 \\ 0.006 & 0.491 & 0.405 & -1.093 \\ 0.005 & 0.405 & 0.351 & -0.938 \\ -0.036 & -1.093 & -0.938 & 2.704 \end{bmatrix} \tag{7.10}$$

$$K^* = \begin{bmatrix} -0.038 & -1.296 & -1.113 & 3.173 \end{bmatrix} \tag{7.11}$$

$$H^* = \begin{bmatrix} -0.068 & -1.983 & -1.628 & 4.648 \end{bmatrix} \tag{7.12}$$

$$M_{u_1} = \begin{bmatrix} -8.5 & -64.0 & -115.5 & 0.0 \\ -3.1 & -19.2 & -29.9 & 0.0 \\ -9.5 & -42.8 & -56.3 & -0.5 \\ -5.0 & -24.5 & -33.5 & 1.0 \end{bmatrix} \tag{7.13}$$

$$M_{u_2} = \begin{bmatrix} 1.6 & 17.8 & 17.1 & 1.0 \\ 1.0 & 5.5 & 4.8 & 0.4 \\ 1.5 & 8.1 & 6.6 & 0.0 \\ 1.0 & 5.4 & 4.4 & 0.0 \end{bmatrix} \tag{7.14}$$

$$M_y = \begin{bmatrix} 0 & 0 & -8.5 & -38.5 \\ 0 & 0 & -3.1 & -10.0 \\ 1.0 & 1.5 & -6.0 & -17.8 \\ 0 & -1.0 & -6.0 & -12.5 \end{bmatrix} \tag{7.15}$$

Data sets were collected over 599 intervals, with two different white noises as the exploration signals (for u_1 and u_2). The initial controller for the learning process was 0.3 of the optimal gain, and the stop threshold is $\epsilon = 10^{-2}$.

Table 7.1 shows the values of the gain vector $\bar{K}_i$ in selected iterations (every three iterations and the final iteration) compared to the optimal gain $\bar{K}^*$. Initially, large deviations from $\bar{K}^*$ are observed in all components. As the iterations progress, the gain values approach the optimal solution. The convergence behavior is further quantified in Table 7.2, which reports the relative error of $\bar{K}_i$ from $\bar{K}^*$. A noticeable reduction in relative error occurs between iterations 4 and 7, with most components reaching errors below 5% by the final iteration.

Component	$\bar{K}^*$	$\bar{K}_1$	$\bar{K}_4$	$\bar{K}_7$	$\bar{K}_{10}$	$\bar{K}_{11}$
1	-0.9800	-6.1315	-1.3418	-1.1244	-1.1236	-1.1236
2	-2.8315	-20.1383	-4.5317	-2.9322	-2.9301	-2.9301
3	-0.5170	-14.3899	-2.5146	0.0963	0.0974	0.0974
4	3.7297	8.0460	3.6600	3.7380	3.7380	3.7380
5	0.1465	1.0221	0.2961	0.1918	0.1915	0.1916
6	0.3403	2.7847	0.8217	0.3060	0.3057	0.3057
7	-0.2977	0.5898	0.0776	-0.3839	-0.3841	-0.3841
8	-0.4915	-1.1687	-0.4466	-0.4963	-0.4963	-0.4963
9	-1.1128	-2.0190	-1.0920	-1.1049	-1.1049	-1.1049
10	-4.8425	-10.1841	-4.7675	-4.8659	-4.8659	-4.8659
11	-8.0482	-20.1670	-8.2668	-8.1669	-8.1664	-8.1664
12	-5.5162	-16.2184	-6.0842	-5.3242	-5.3238	-5.3238

Table 7.1: Comparison of gain vector $\bar{K}$ values at selected iterations (every 3 iterations and final) versus the optimal vector $\bar{K}^*$ under $\mathcal{H}_\infty$ control.

Component	Error($\bar{K}_1$)	Error($\bar{K}_4$)	Error($\bar{K}_7$)	Error($\bar{K}_{10}$)	Error($\bar{K}_{11}$)
1	525.65%	36.92%	14.73%	14.65%	14.65%
2	611.24%	60.05%	3.56%	3.48%	3.48%
3	2683.15%	386.35%	118.62%	118.83%	118.83%
4	115.73%	1.87%	0.22%	0.23%	0.23%
5	597.75%	102.17%	30.90%	30.77%	30.77%
6	718.19%	141.42%	10.08%	10.18%	10.18%
7	298.13%	126.05%	28.96%	29.02%	29.02%
8	137.75%	9.15%	0.97%	0.98%	0.98%
9	81.42%	1.87%	0.72%	0.71%	0.71%
10	110.31%	1.55%	0.48%	0.48%	0.48%
11	150.58%	2.72%	1.47%	1.47%	1.47%
12	194.02%	10.30%	3.48%	3.49%	3.49%

Table 7.2: Relative error (%) between $\bar{K}_i$ and the optimal gain vector $\bar{K}^*$ at selected iterations.

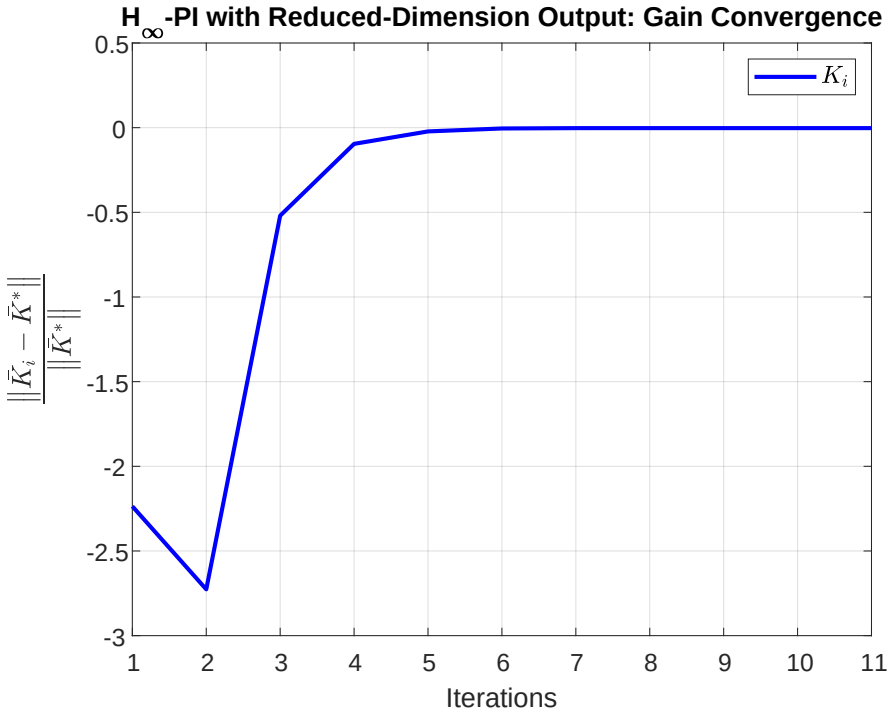

Figure 7.1: Convergence of the normalized parameters K_i towards the optimal gain K^*, under $\mathcal{H}_\infty$ control, in case of one HDV.

Figure 7.1 illustrates the relative norm error between the gain vector $\bar{K}_i$ at each iteration and the optimal gain $\bar{K}^*$, normalized by $\|\bar{K}^*\|$. The convergence behavior is evident, with a sharp reduction in error observed in the early iterations. Specifically, the error drops significantly between iterations 2 and 4, followed by a steady convergence toward zero from iteration 5 onward. After iteration 6, only marginal improvements are observed, indicating the numerical convergence of the gain vector.

7.1.2 Case 2: Two HDVs in the platoon

In this scenario, a mixed platoon composed of two HDVs followed by a CAV is considered, resulting in the Leader-HDV-HDV-CAV configuration.

Based on the formulation provided in Chapter 3, the state vector, consisting of spacing and velocity errors, is,

$$x = \begin{bmatrix} \Delta s_1 \\ e_{v,1} \\ \Delta s_2 \\ e_{v,2} \\ e_{s,3} \\ e_{v,3} \end{bmatrix} \tag{7.16}$$

and the model matrices are,

$$A = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 \\ f_{s_1} & -(f_{\Delta v_1} + f_{v_1}) & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & f_{\Delta v_2} & f_{s_2} & -(f_{\Delta v_2} + f_{v_2}) & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -t_h \\ 1 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 1 \\ f_{\Delta v_1} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{7.17}$$

The choice of different parameter values for the two HDVs reflects the variability in individual driving behaviors. Since human drivers exhibit diverse responses and decision-making patterns, it is essential to capture this variation through distinct model parameters for each vehicle. Therefore, we enlarged the second HDV parameters by 10%.

Substituting the values from Fig. 2.2 for a velocity of $20[\frac{m}{s}]$, we get: $f_{s1} = 0.05$, $f_{v1} = 0.08$,

$f_{\Delta v_1} = 0.34$, $f_{s_2} = 0.05 \cdot 1.1$, $f_{v_2} = 0.08 \cdot 1.1$, $f_{\Delta v_2} = 0.34 \cdot 1.1$, $t_h = 0.5[s]$, and the matrices,

$$A = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 \\ 0.08 & -0.48 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0.374 & 0.055 & -0.462 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -0.5 \\ 1 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 1 \\ 0.35 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{7.18}$$

To reduce the dimension of y, we apply the method of Chapter 6 with the following constraints,

$$\begin{aligned} L_{12} &= L_{11}, & L_{13} &= \frac{3}{2}L_{11}, \\ L_{22} &= L_{21}, & L_{23} &= \frac{3}{2}L_{21}, \\ L_{32} &= L_{31}, & L_{33} &= \frac{3}{2}L_{31}, \\ L_{42} &= L_{41}, & L_{43} &= \frac{3}{2}L_{41}. \\ L_{52} &= L_{51}, & L_{53} &= \frac{3}{2}L_{51}, \\ L_{62} &= L_{61}, & L_{63} &= \frac{3}{2}L_{61}. \end{aligned} \tag{7.19}$$

which leads to,

$$LC = \begin{bmatrix} L_{11} \\ L_{21} \\ L_{31} \\ L_{41} \\ L_{51} \\ L_{61} \end{bmatrix} \begin{bmatrix} 1 & 1 & \frac{3}{2} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \tag{7.20}$$

and therefore,

$$C_{\text{new}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & -2.5 \end{bmatrix} \tag{7.21}$$

We conducted the simulation with,

$$R = \begin{bmatrix} 1 & 0 \\ 0 & -\gamma_{min}^2 \end{bmatrix} \tag{7.22}$$

and chose $\gamma_{min}^2 = 0.2$ **to ensure weak string stability and optimal gains**. The eigenvalues of the observer are placed at -1. The optimal controller as obtained by solving the ZSG-ARE (4.26) is,

$$P^* = \begin{bmatrix} 0.002 & -0.001 & 0.002 & -0.008 & -0.003 & -0.004 \\ -0.001 & 0.046 & 0.006 & 0.054 & 0.027 & -0.118 \\ 0.002 & 0.006 & 0.002 & 0.001 & 0.001 & -0.020 \\ -0.008 & 0.054 & 0.001 & 0.443 & 0.366 & -0.968 \\ -0.003 & 0.027 & 0.001 & 0.366 & 0.320 & -0.841 \\ -0.004 & -0.118 & -0.020 & -0.968 & -0.841 & 2.406 \end{bmatrix} \tag{7.23}$$

$$K^* = \begin{bmatrix} -0.002 & -0.131 & -0.021 & -1.151 & -1.001 & 2.826 \end{bmatrix} \tag{7.24}$$

$$H^* = \begin{bmatrix} -0.043 & -0.413 & -0.094 & -0.303 & -0.169 & 1.189 \end{bmatrix} \tag{7.25}$$

$$M_{u_1} = \begin{bmatrix} 42.5 & 651.4 & 3749.8 & 9576.0 & 9128.2 & 0.0 \\ 9.2 & 139.8 & 796.6 & 2014.1 & 1901.2 & 0.0 \\ -143.8 & -1921.6 & -9927.3 & -23314.2 & -20833.7 & 0.0 \\ -11.8 & -132.4 & -586.7 & -1213.8 & -979.5 & 0.0 \\ -22.7 & -223.5 & -929.3 & -1867.9 & -1488.6 & -0.5 \\ -13.8 & -142.3 & -606.2 & -1231.9 & -985.7 & 1.0 \end{bmatrix} \tag{7.26}$$

$$M_{u_2} = \begin{bmatrix} 1.6 & -28.1 & -447.1 & -412.7 & 5.6 & 1.0 \\ 1.0 & -4.8 & -91.3 & -83.7 & 2.1 & 0.4 \\ 1.6 & 155.6 & 1126.9 & 973.4 & 0.4 & 0.0 \\ 1.0 & 14.6 & 59.9 & 46.4 & 0.1 & 0.0 \\ 1.5 & 21.9 & 89.8 & 69.4 & 0.0 & 0.0 \\ 1.0 & 14.6 & 59.8 & 46.2 & 0.0 & 0.0 \end{bmatrix} \tag{7.27}$$

$$M_y = \begin{bmatrix} 0.0 & 0.0 & 42.5 & 524.0 & 2177.7 & 3042.7 \\ 0.0 & 0.0 & 9.2 & 112.2 & 460.1 & 633.7 \\ 0.0 & 0.0 & -143.8 & -1490.3 & -5456.5 & -6944.6 \\ 0.0 & 0.0 & -11.8 & -97.0 & -295.8 & -326.5 \\ 1.0 & 3.5 & -15.2 & -150.4 & -453.2 & -494.9 \\ 0.0 & -1.0 & -16.8 & -106.9 & -305.5 & -330.6 \end{bmatrix} \tag{7.28}$$

Data sets were collected over 5600 time intervals, with two different white noises as the exploration signals (at u_1 and u_2). The learning process was initialized with the gain values $0.3H^*, 0.3K^*$, and the terminating threshold is $\epsilon = 10^{-1}$.

Component	$\bar{K}^*$	$\bar{K}_1$	$\bar{K}_4$	$\bar{K}_7$	$\bar{K}_9$
1	-1.0033	-4.5859	-1.0490	-0.9894	-0.9894
2	-5.8827	-25.5367	-6.2579	-5.6860	-5.6860
3	-13.4234	-49.9928	-14.5543	-12.3601	-12.3602
4	-13.3521	-29.9682	-14.4178	-10.7642	-10.7642
5	-2.6315	13.0168	-2.3365	-0.2550	-0.2551
6	3.3266	6.9281	3.3262	3.3266	3.3266
7	0.0068	0.0817	0.0030	0.0068	0.0068
8	-0.0069	-0.1529	0.0591	-0.0208	-0.0208
9	-0.2241	-2.8889	-0.1559	-0.3488	-0.3488
10	-0.6177	-4.2345	-0.6109	-0.7285	-0.7285
11	-0.4582	-1.8388	-0.4480	-0.4582	-0.4582
12	-0.0509	-0.2583	-0.0489	-0.0509	-0.0509
13	-1.0006	-1.8203	-0.9975	-1.0006	-1.0006
14	-6.3283	-12.3890	-6.3200	-6.3283	-6.3283
15	-16.9864	-36.2921	-17.0143	-16.9725	-16.9725
16	-24.8332	-56.9879	-25.0683	-24.6784	-24.6784
17	-20.4634	-44.8958	-20.8454	-19.8648	-19.8648
18	-7.8638	-10.1241	-7.7640	-7.0717	-7.0717

Table 7.3: Gain vector $\bar{K}_i$ at selected iterations (1, 4, 7, and 9) compared to the optimal gain $\bar{K}^*$ under $\mathcal{H}_\infty$ control for two HDVs.

Component	Error($\bar{K}_1$)	Error($\bar{K}_4$)	Error($\bar{K}_7$)	Error($\bar{K}_9$)
1	357.08%	4.55%	1.39%	1.39%
2	334.10%	6.38%	3.34%	3.34%
3	272.43%	8.43%	7.92%	7.92%
4	124.45%	7.98%	19.38%	19.38%
5	594.65%	11.21%	90.31%	90.31%
6	108.27%	0.01%	0.00%	0.00%
7	1108.92%	55.10%	0.01%	0.01%
8	2113.63%	955.98%	201.11%	201.10%
9	1189.23%	30.42%	55.67%	55.67%
10	585.51%	1.10%	17.94%	17.93%
11	301.27%	2.24%	0.00%	0.00%
12	407.09%	3.91%	0.01%	0.01%
13	81.93%	0.31%	0.00%	0.00%
14	95.77%	0.13%	0.00%	0.00%
15	113.65%	0.16%	0.08%	0.08%
16	129.48%	0.95%	0.62%	0.62%
17	119.40%	1.87%	2.93%	2.93%
18	28.74%	1.27%	10.07%	10.07%

Table 7.4: Relative error (%) between $\bar{K}_i$ and the optimal gain vector $\bar{K}^*$ at iterations 1, 4, 7, and 9.

Figure 7.2 illustrates the normalized relative error between the gain vector $\bar{K}_i$ and the optimal gain $\bar{K}^*$ over iterations. A clear convergence trend is observed, where the error magnitude decreases rapidly within the first few iterations and stabilizes from iteration 5 onward. This behavior is also reflected numerically in Table 7.4, which shows the relative error (%) for each gain component. In parallel, Table 7.3 presents the gain values in selected iterations (1, 4, 7, and 9) compared to $\bar{K}^*$, confirming the consistency of convergence across most components.

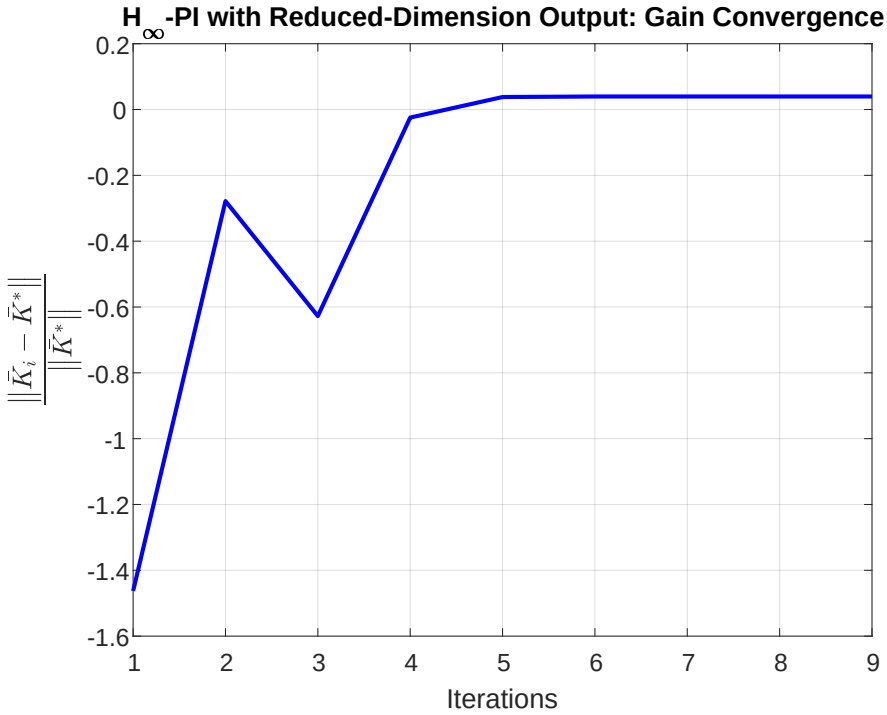

Figure 7.2: Convergence of the normalized parameters K_i towards the optimal gain K^*, under $\mathcal{H}_\infty$ control, in the case of two HDVs.

7.1.3 Case 3: Three HDVs in the Platoon

In this scenario, a mixed platoon composed of three HDVs followed by a CAV is considered, resulting in Leader-HDV-HDV–HDV-CAV configuration.

According to the formulation presented in Chapter 3, the state vector, which includes spacing

and velocity errors, is defined as follows,

$$x = \begin{bmatrix} \Delta s_1 \\ e_{v,1} \\ \Delta s_2 \\ e_{v,2} \\ \Delta s_3 \\ e_{v,3} \\ e_{s,4} \\ e_{v,4} \end{bmatrix} \tag{7.29}$$

with the model matrices,

$$A = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ f_{s_1} & -(f_{\Delta v_1} + f_{v_1}) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & f_{\Delta v_2} & f_{s_2} & -(f_{\Delta v_2} + f_{v_2}) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & f_{\Delta v_3} & f_{s_3} & -(f_{\Delta v_3} + f_{v_3}) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -t_h \\ 1 \end{bmatrix} \qquad B_2 = \begin{bmatrix} 1 \\ f_{\Delta v_1} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{7.30}$$

The choice of different parameter values for HDVs reflects the variability in individual driving behaviors. Therefore, we enlarge the second HDV parameters by 10% and reduce those of the

third HDV by 10%.

Substituting the values from Fig. 2.2 for $20[\frac{m}{s}]$ we have: $f_{s_1} = 0.05$, $f_{v_1} = 0.08$, $f_{\Delta v_1} = 0.34$, $f_{s_2} = 0.05 \cdot 1.1$, $f_{v_2} = 0.08 \cdot 1.1$, $f_{\Delta v_2} = 0.34 \cdot 1.1$, $f_{s_3} = 0.05 \cdot 0.9$, $f_{v_3} = 0.08 \cdot 0.9$, $f_{\Delta v_3} = 0.34 \cdot 0.9$, $t_h = 0.5[s]$, and the model matrices are,

$$A = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.05 & -0.42 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0.374 & 0.055 & -0.462 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0.306 & 0.045 & -0.378 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -0.5 \\ 1 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 1 \\ 0.35 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{7.31}$$

To reduce the dimension of y, we apply the method of Chapter 6 with the following constraints,

$$\begin{aligned} L_{12} &= L_{11}, & L_{13} &= \frac{3}{2} L_{11} \\ L_{22} &= L_{21}, & L_{23} &= \frac{3}{2} L_{21} \\ L_{32} &= L_{31}, & L_{33} &= \frac{3}{2} L_{31} \\ L_{42} &= L_{41}, & L_{43} &= \frac{3}{2} L_{41} \\ L_{52} &= L_{51}, & L_{53} &= \frac{3}{2} L_{51} \\ L_{62} &= L_{61}, & L_{63} &= \frac{3}{2} L_{61} \\ L_{72} &= L_{71}, & L_{73} &= \frac{3}{2} L_{71} \\ L_{82} &= L_{81}, & L_{83} &= \frac{3}{2} L_{81} \end{aligned} \tag{7.32}$$

which leads to,

$$LC = \begin{bmatrix} L_{11} \\ L_{21} \\ L_{31} \\ L_{41} \\ L_{51} \\ L_{61} \\ L_{71} \\ L_{81} \end{bmatrix} \begin{bmatrix} 1 & 1 & \frac{3}{2} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \tag{7.33}$$

and therefore,

$$C_{\text{new}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & -\frac{5}{2} \end{bmatrix} \tag{7.34}$$

This choice (i.e., (7.32)) gives a higher weight to the velocity error with respect to the leader (i.e., y_3) in y_{new} (compared to the weight given to y_1 and y_2); by that, we prioritize string-stability. For the design we take,

$$R = \begin{bmatrix} 1 & 0 \\ 0 & -\gamma_{min}^2 \end{bmatrix} \tag{7.35}$$

with $\gamma_{min} = 0.2$ (note that $\gamma < 1$ is essential from string-stability). The eigenvalues of the observer were placed at -1. The optimal controller, obtained by solving the ZSG-ARE (4.26), is,

$$P^* = \begin{bmatrix} 0.001 & 0.000 & 0.001 & -0.004 & 0.001 & -0.007 & -0.002 & -0.001 \\ 0.000 & 0.024 & 0.003 & 0.016 & 0.004 & -0.004 & -0.006 & -0.012 \\ 0.001 & 0.003 & 0.001 & -0.001 & 0.001 & -0.006 & -0.002 & -0.002 \\ -0.004 & 0.016 & -0.001 & 0.032 & 0.003 & 0.052 & 0.023 & -0.094 \\ 0.001 & 0.004 & 0.001 & 0.003 & 0.001 & 0.002 & 0.001 & -0.016 \\ -0.007 & -0.004 & -0.006 & 0.052 & 0.002 & 0.460 & 0.368 & -0.994 \\ -0.002 & -0.006 & -0.002 & 0.023 & 0.001 & 0.368 & 0.318 & -0.841 \\ -0.001 & -0.012 & -0.002 & -0.094 & -0.016 & -0.994 & -0.841 & 2.396 \end{bmatrix} \tag{7.36}$$

$$K^* = \begin{bmatrix} 0.000 & -0.010 & -0.001 & -0.106 & -0.016 & -1.178 & -1.000 & 2.817 \end{bmatrix} \tag{7.37}$$

$$H^* = \begin{bmatrix} -0.033 & -0.190 & -0.053 & -0.049 & -0.053 & 0.199 & 0.095 & 0.119 \end{bmatrix} \tag{7.38}$$

$$M_{u_1} = \begin{bmatrix} 8.9 & 176.3 & 1446.2 & 6086.3 & 13131.9 & 11849.7 & 938.9 & 0.0 \\ 0.0 & -7.8 & -162.9 & -1352.3 & -5687.5 & -12101.5 & -10353.8 & 0.0 \\ 289.0 & 6708.9 & 66664.8 & 361196.7 & 1121369.5 & 1885041.8 & 1334892.7 & 0.0 \\ 55.1 & 1256.1 & 12273.2 & 65444.7 & 200096.9 & 331488.3 & 231533.5 & 0.0 \\ -505.4 & -10887.4 & -102066.4 & -528204.6 & -1580147.2 & -2575669.1 & -1777046.7 & 0.0 \\ -18.2 & -305.7 & -2325.8 & -10157.0 & -26479.8 & -38559.6 & -24225.2 & 0.0 \\ -32.3 & -493.5 & -3593.6 & -15410.1 & -39892.8 & -57939.4 & -36365.6 & -0.5 \\ -20.2 & -319.7 & -2367.7 & -10226.7 & -26548.5 & -38598.3 & -24234.4 & 1.0 \end{bmatrix} \tag{7.39}$$

$$M_{u_2} = \begin{bmatrix} 1.6 & 10.5 & -105.9 & -98.5 & 35.0 & 25.4 & 7.7 & 1.0 \\ 1.0 & 6.4 & 16.7 & 155.7 & 150.9 & 8.8 & 2.6 & 0.3 \\ 1.6 & -274.0 & -4768.8 & -21801.5 & -17306.1 & 2.5 & 0.3 & 0.0 \\ 1.0 & -50.3 & -887.7 & -3837.3 & -3000.1 & 0.9 & 0.1 & 0.0 \\ 1.6 & 521.8 & 7373.5 & 29902.9 & 23049.8 & 0.1 & 0.0 & 0.0 \\ 1.0 & 21.4 & 159.1 & 453.2 & 314.5 & 0.0 & 0.0 & 0.0 \\ 1.5 & 32.1 & 238.7 & 679.7 & 471.6 & 0.0 & 0.0 & 0.0 \\ 1.0 & 21.4 & 159.1 & 453.2 & 314.4 & 0.0 & 0.0 & 0.0 \end{bmatrix} \tag{7.40}$$

$$M_y = \begin{bmatrix} 0.0 & 0.0 & 8.9 & 149.7 & 997.0 & 3095.4 & 3845.6 & 313.0 \\ 0.0 & 0.0 & 0.0 & -7.9 & -139.2 & -934.7 & -2883.4 & -3451.3 \\ 0.0 & 0.0 & 289.0 & 5842.1 & 49138.5 & 213781.2 & 480025.8 & 444964.2 \\ 0.0 & 0.0 & 55.1 & 1090.8 & 9000.8 & 38442.2 & 84770.2 & 77177.8 \\ 0.0 & 0.0 & -505.4 & -9371.3 & -73952.6 & -306346.8 & -661106.7 & -592348.9 \\ 0.0 & 0.0 & -18.2 & -251.1 & -1572.5 & -5439.4 & -10161.5 & -8075.1 \\ 1.0 & 5.5 & -16.8 & -373.1 & -2376.2 & -8190.3 & -15265.7 & -12120.2 \\ 0.0 & -1.0 & -25.2 & -272.1 & -1607.5 & -5474.2 & -10181.8 & -8080.8 \end{bmatrix} \tag{7.41}$$

Data sets were collected over 6,700 time intervals, with two different white noise as the exploration signals. The controller was initialized with optimal gain values of $0.3K^*, 0.3H^*$, and the stop threshold was $\epsilon = 10^{-1}$. The learning process results are presented in Table 7.5, Table 7.6,

and Figure 7.3.

Component	$\bar{K}^*$	$\bar{K}_1$	$\bar{K}_4$	$\bar{K}_7$	$\bar{K}_{13}$
1	-1.00	-7.42	-0.87	-2.12	-1.51
2	-8.05	-21.14	3.80	-10.13	-10.64
3	-28.15	56.98	14.39	-33.32	-34.85
4	-54.97	281.72	28.15	-63.46	-64.44
5	-63.50	335.23	12.31	-82.01	-73.60
6	-40.36	114.95	-26.17	-69.98	-53.67
7	-8.52	157.27	-20.68	-22.49	-18.38
8	3.32	5.39	3.30	3.28	3.29
9	-0.00	-0.05	-0.29	0.01	0.02
10	-0.04	-6.57	-1.41	-0.24	-0.07
11	-0.14	-8.69	-2.40	-0.47	-0.33
12	-0.30	-6.38	-2.20	-0.53	-0.46
13	-0.38	-5.40	-1.67	-0.70	-0.50
14	-0.22	-1.45	-0.99	-0.48	-0.35
15	-0.04	-0.32	-0.26	-0.11	-0.07
16	-0.00	-0.03	-0.03	-0.01	-0.01
17	-1.00	1.54	-0.69	-0.60	-0.84
18	-8.32	0.93	-9.21	-7.88	-7.73
19	-30.58	-24.61	-30.19	-29.49	-29.49
20	-65.18	-62.35	-57.85	-63.95	-64.44
21	-88.36	-37.33	-67.62	-87.52	-88.74
22	-78.06	18.24	-54.01	-79.80	-79.08
23	-43.07	-25.34	-36.64	-50.75	-45.95
24	-12.02	37.94	-16.00	-16.53	-15.20

Table 7.5: Comparison of gain vector $\bar{K}$ values at iterations 1, 4, 7, and 13 compared to the optimal vector $\bar{K}^*$, under $\mathcal{H}_\infty$ control, in the case of three HDVs.

Component	Error($\bar{K}_1$)	Error($\bar{K}_4$)	Error($\bar{K}_7$)	Error($\bar{K}_{13}$)
1	642.83%	13.11%	112.82%	51.62%
2	162.48%	147.25%	25.80%	32.14%
3	302.44%	151.12%	18.36%	23.81%
4	612.48%	151.21%	15.44%	17.23%
5	627.89%	119.39%	29.14%	15.90%
6	384.79%	35.16%	73.37%	32.98%
7	1946.87%	142.84%	164.13%	115.85%
8	62.40%	0.53%	1.12%	0.81%
9	1192.79%	7503.59%	459.73%	540.75%
10	18139.27%	3816.62%	565.54%	104.49%
11	6040.31%	1593.94%	230.91%	136.09%
12	2038.58%	637.53%	78.40%	54.23%
13	1319.93%	340.33%	83.13%	32.43%
14	544.15%	341.52%	115.48%	55.62%
15	800.83%	635.15%	213.51%	99.07%
16	1073.32%	1028.66%	344.25%	154.69%
17	253.66%	30.64%	39.93%	15.57%
18	111.13%	10.79%	5.32%	7.03%
19	19.53%	1.28%	3.57%	3.59%
20	4.33%	11.24%	1.88%	1.13%
21	57.76%	23.47%	0.95%	0.43%
22	123.37%	30.81%	2.23%	1.31%
23	41.17%	14.92%	17.83%	6.69%
24	415.76%	33.11%	37.59%	26.52%

Table 7.6: Relative error (%) between $\bar{K}_i$ and the optimal gain vector $\bar{K}^*$ at iterations 1, 4, 7, and 13, in the case of three HDVs.

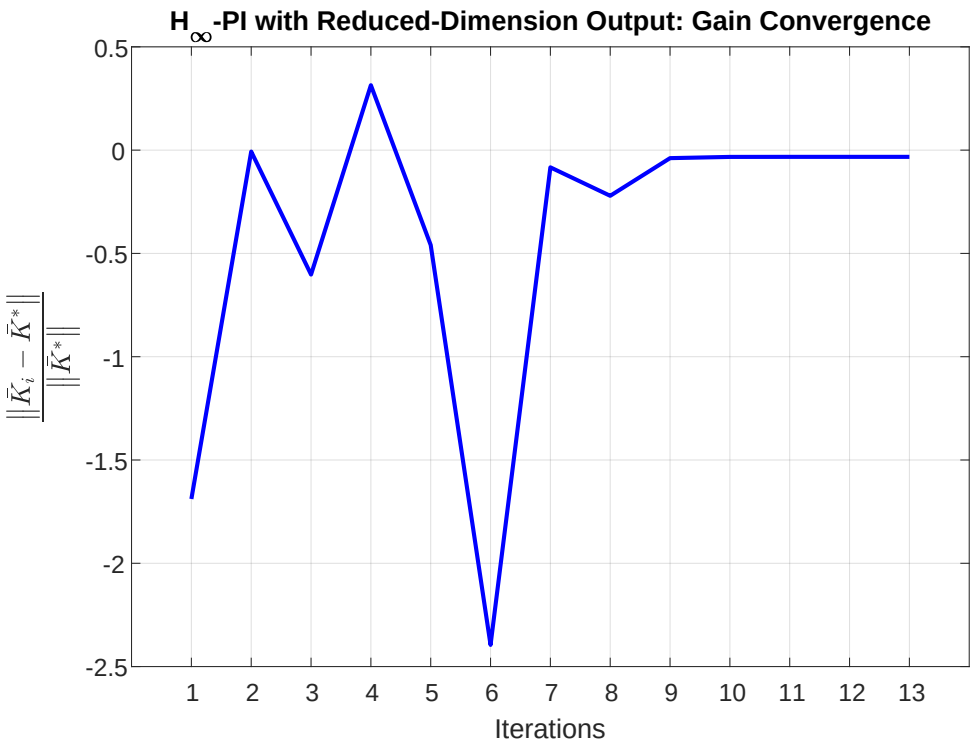

Figure 7.3: Convergence of the normalized parameters $\bar{K}_i$ towards the optimal gain $\bar{K}^*$, under $\mathcal{H}_\infty$ control, in case of three HDVs.

7.1.4 String Stability Analysis

As previously discussed in Section 4, our objective was to design a control input for a CAV in a mixed platoon such that, for any disturbance (injected by the leader), the L_2 norm of its influence is attenuated by the controlled CAV (at the back of the mixed platoon unit). This goal was addressed through the criterion presented in equation (4.24), which requires that $\gamma < 1$.

In practical terms, this ensures that the CAV responds to disturbances in a way that reduces

their "energy", thereby attenuating their effect as they propagate downstream. As a result, the system exhibits *weak string stability*, wherein disturbances introduced at the front of the platoon diminish rather than amplify. The term "weak" means that this property is allowed to skip HDVs, which are not controlled and inherently not string stable. On the other hand, it requires the controlled CAV at the back to compensate for the string instability of the HDVs in front of it, as the string stability is demanded between the leader and the CAV. By measuring disturbance attenuation through the L_2 norm ratio, the string stability criterion can be embedded in the $\mathcal{H}_\infty$ control design.

In the following section, we implement the controller presented in Section 7.1.3 within a simulation framework that introduces a bounded L_2 norm velocity disturbance at the head of the platoon.

Figure 7.4 illustrates the propagation of the velocity error throughout the platoon. The dashed black line indicates the injected disturbance, while the colored curves represent the velocity errors of HDVs and CAV.

In Figure 7.5, we present an analysis of the L_2 norm ratios between adjacent vehicles, as well as the relationship between the last CAV and the external disturbance injected by the leader.

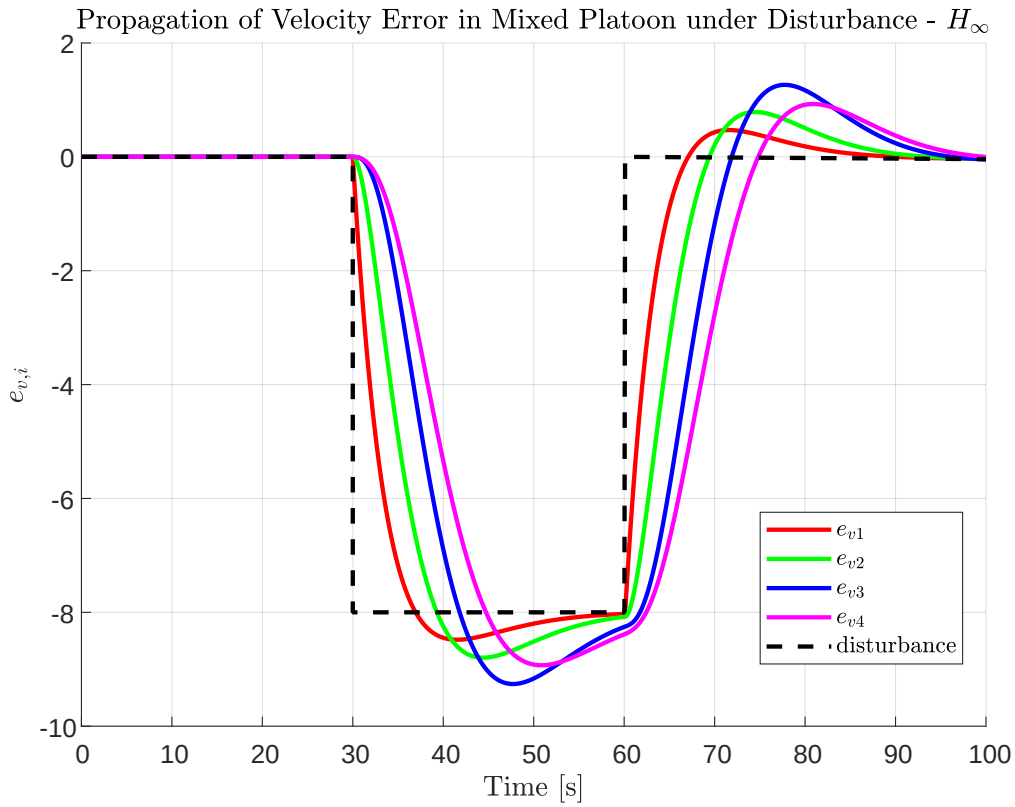

Figure 7.4: Velocity error response of the mixed platoon to a velocity disturbance applied to the lead vehicle, under $\mathcal{H}_\infty$ control. The dashed black line represents the injected disturbance, while the colored curves correspond to the velocity errors of HDVs and CAV.

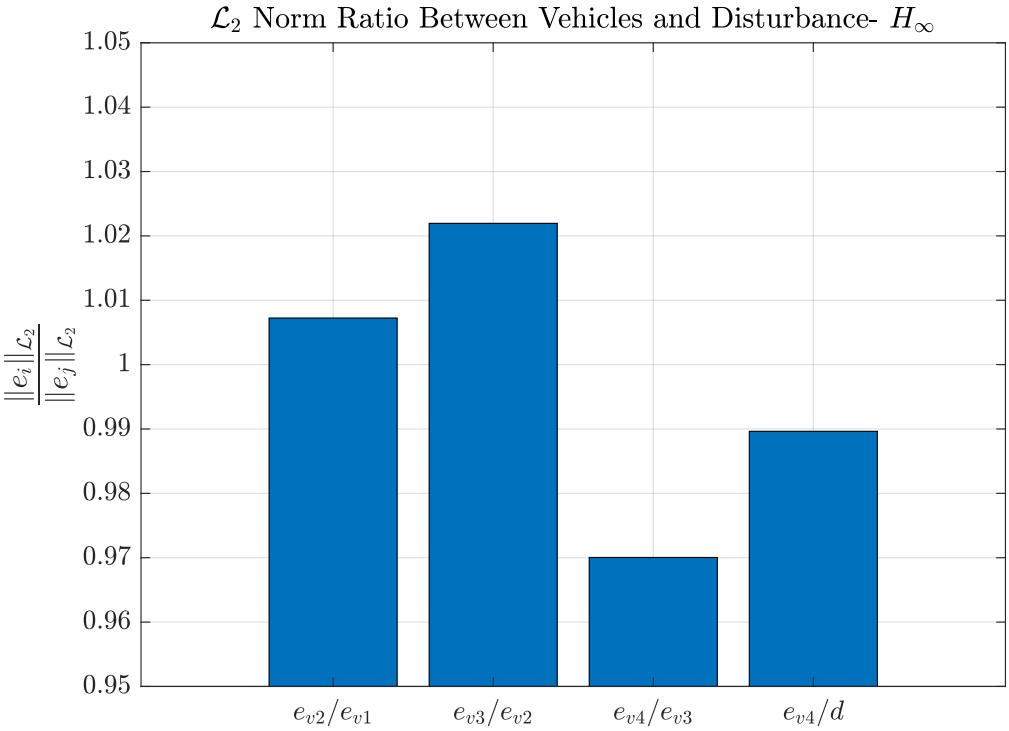

Figure 7.5: L_2-norm ratios of velocity errors under $\mathcal{H}_\infty$ control, including ratios between HDVs, between the CAV and its predecessor, and between the CAV and the external disturbance.

7.1.5 Comparison Between LQR and $\mathcal{H}_\infty$ Control Performance

In the previous sections, we designed a controller using the $\mathcal{H}_\infty$ methodology, employing the performance criterion given in (4.24) to ensure weak string stability. In this section, we aim to evaluate the effectiveness of this method by comparing it to the LQR approach.

Although the LQR method provides optimal gains in the sense of minimizing a quadratic cost

function, it does not explicitly incorporate performance demands related to disturbance attenuation. In contrast, the $\mathcal{H}_\infty$ framework is inherently designed to address worst-case disturbances through an energy-based criterion.

To highlight the differences between the two approaches, the design requirements were taken to be similar. A mixed platoon composed of three HDVs followed by a CAV is considered, resulting in the Leader-HDV-HDV-HDV-CAV configuration.

The state space model, considered for the LQR design, is the following (note that this is the same model used for the $\mathcal{H}_\infty$ design with three-HDVs but without the disturbance input, therefore, $u = u_1$ and $B = B_1$). The state vector is,

$$x = \begin{bmatrix} \Delta s_1 \\ e_{v,1} \\ \Delta s_2 \\ e_{v,2} \\ \Delta s_3 \\ e_{v,3} \\ e_{s,4} \\ e_{v,4} \end{bmatrix} \tag{7.42}$$

and the model matrices,

$$A = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ f_{s_1} & -(f_{\Delta v_1} + f_{v_1}) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & f_{\Delta v_2} & f_{s_2} & -(f_{\Delta v_2} + f_{v_2}) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & f_{\Delta v_3} & f_{s_3} & -(f_{\Delta v_3} + f_{v_3}) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -t_h \\ 1 \end{bmatrix} \tag{7.43}$$

Substituting the values as in (7.31), we get,

$$A = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.05 & -0.42 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0.374 & 0.055 & -0.462 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0.306 & 0.045 & -0.378 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -0.5 \\ 1 \end{bmatrix} \tag{7.44}$$

Applying constraints on L, similar to (7.32),

$$\begin{aligned} L_{12} &= L_{11}, & L_{13} &= \frac{3}{2}L_{11} \\ L_{22} &= L_{21}, & L_{23} &= \frac{3}{2}L_{21} \\ L_{32} &= L_{31}, & L_{33} &= \frac{3}{2}L_{31} \\ L_{42} &= L_{41}, & L_{43} &= \frac{3}{2}L_{41} \\ L_{52} &= L_{51}, & L_{53} &= \frac{3}{2}L_{51} \\ L_{62} &= L_{61}, & L_{63} &= \frac{3}{2}L_{61} \\ L_{72} &= L_{71}, & L_{73} &= \frac{3}{2}L_{71} \\ L_{82} &= L_{81}, & L_{83} &= \frac{3}{2}L_{81} \end{aligned}$$

it leads to,

$$LC = \begin{bmatrix} L_{11} \\ L_{21} \\ L_{31} \\ L_{41} \\ L_{51} \\ L_{61} \\ L_{71} \\ L_{81} \end{bmatrix} \begin{bmatrix} 1 & 1 & \frac{3}{2} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

and therefore,

$$C_{\text{new}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & -\frac{5}{2} \end{bmatrix}$$

This simulation is conducted with $Q_y = R = 1$. The eigenvalues of the observer are placed at -1. The optimal controller as obtained by solving the ARE (4.12) is,

$$P^* = \begin{bmatrix} 0.001 & 0.000 & 0.001 & -0.002 & 0.000 & -0.004 & -0.001 & -0.000 \\ 0.000 & 0.016 & 0.002 & 0.013 & 0.004 & -0.003 & -0.005 & -0.012 \\ 0.001 & 0.002 & 0.001 & 0.000 & 0.001 & -0.004 & -0.002 & -0.002 \\ -0.002 & 0.013 & 0.000 & 0.026 & 0.004 & 0.046 & 0.022 & -0.094 \\ 0.000 & 0.004 & 0.001 & 0.004 & 0.001 & 0.003 & 0.002 & -0.016 \\ -0.004 & -0.003 & -0.004 & 0.046 & 0.003 & 0.448 & 0.365 & -0.994 \\ -0.001 & -0.005 & -0.002 & 0.022 & 0.002 & 0.365 & 0.317 & -0.842 \\ -0.000 & -0.012 & -0.002 & -0.094 & -0.016 & -0.994 & -0.842 & 2.396 \end{bmatrix} \tag{7.45}$$

$$K^* = \begin{bmatrix} 0.000 & -0.010 & -0.001 & -0.105 & -0.016 & -1.177 & -1.000 & 2.817 \end{bmatrix} \tag{7.46}$$

$$M_u = \begin{bmatrix} 8.9 & 176.3 & 1446.2 & 6086.3 & 13131.9 & 11849.7 & 938.9 & 0.0 \\ 0.0 & -7.8 & -162.9 & -1352.3 & -5687.5 & -12101.5 & -10353.8 & 0.0 \\ 289.0 & 6708.9 & 66664.8 & 361196.7 & 1121369.5 & 1885041.8 & 1334892.7 & 0.0 \\ 55.1 & 1256.1 & 12273.2 & 65444.7 & 200096.9 & 331488.3 & 231533.5 & 0.0 \\ -505.4 & -10887.4 & -102066.4 & -528204.6 & -1580147.2 & -2575669.1 & -1777046.7 & 0.0 \\ -18.2 & -305.7 & -2325.8 & -10157.0 & -26479.8 & -38559.6 & -24225.2 & 0.0 \\ -32.3 & -493.5 & -3593.6 & -15410.1 & -39892.8 & -57939.4 & -36365.6 & -0.5 \\ -20.2 & -319.7 & -2367.7 & -10226.7 & -26548.5 & -38598.3 & -24234.4 & 1.0 \end{bmatrix} \tag{7.47}$$

$$M_y = \begin{bmatrix} 0.0 & -0.0 & 8.9 & 149.7 & 997.0 & 3095.4 & 3845.6 & 313.0 \\ 0.0 & 0.0 & 0.0 & -7.9 & -139.2 & -934.7 & -2883.4 & -3451.3 \\ 0.0 & -0.0 & 289.0 & 5842.1 & 49138.5 & 213781.2 & 480025.8 & 444964.2 \\ 0.0 & -0.0 & 55.1 & 1090.8 & 9000.8 & 38442.2 & 84770.2 & 77177.8 \\ 0.0 & 0.0 & -505.4 & -9371.3 & -73952.6 & -306346.8 & -661106.7 & -592348.9 \\ 0.0 & 0.0 & -18.2 & -251.1 & -1572.5 & -5439.4 & -10161.5 & -8075.1 \\ 1.0 & 5.5 & -16.8 & -373.1 & -2376.2 & -8190.3 & -15265.7 & -12120.2 \\ 0.0 & -1.0 & -25.2 & -272.1 & -1607.5 & -5474.2 & -10181.8 & -8080.8 \end{bmatrix} \tag{7.48}$$

Since the LQR control design problem is formulated for a single-player system (i.e., only $u(t)$), and in this case, the excitation of the CAV at the tail of the platoon is insufficient for the control learning process (as the HDVs are uncontrollable through $u(t)$). To address this challenge, we propose augmenting the system with an additional excitation input $E\tilde{v}_0$ term, which allows injecting a disturbance into the platoon. A conceptually similar approach was adopted in [10], where external excitation was incorporated into the state feedback learning equation. Then, the model for the LQR design is,

$$\dot{x} = Ax + Bu + E\tilde{v}_0. \tag{7.49}$$

where,

$$E = \begin{bmatrix} 1 \\ 0.35 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{7.50}$$

Notice that E is equal to B_2 in (7.31).

When running the algorithm for data collection, $\tilde{v}_0$ is a random white noise, and while simulating to check the controller performance, it is a disturbance input, as seen in Figure 7.7. The

controller was initialized with $0.3K^*$, and the terminating threshold is $\epsilon = 10^{-3}$.

For this design, data sets were collected over 1900 time intervals.

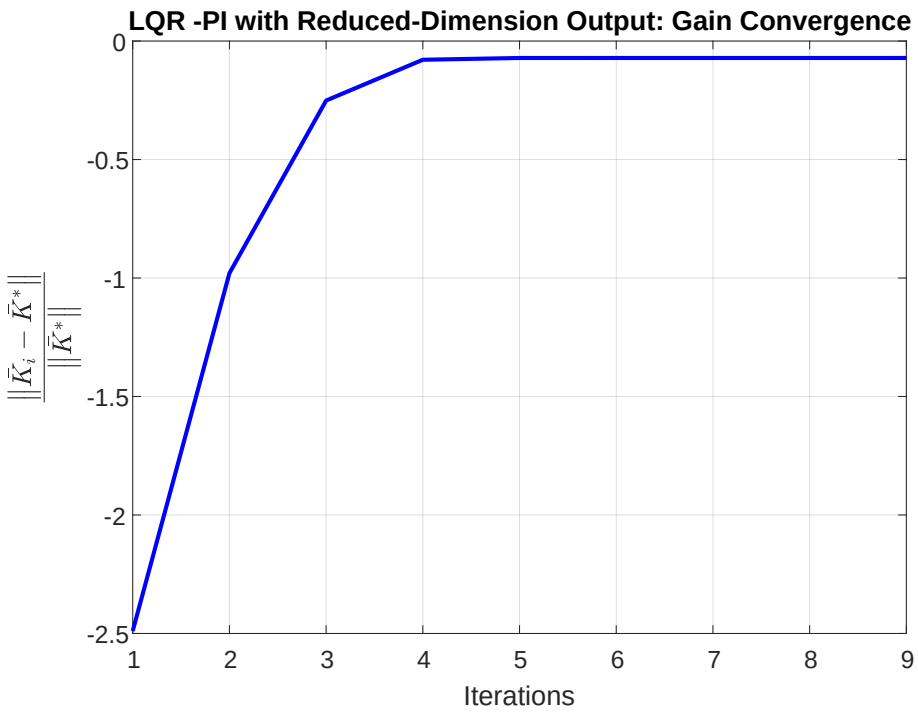

Figure 7.6: Convergence of the normalized gain parameters $\bar{K}_i$ toward the optimal gain $\bar{K}^*$ under LQR control.

Component	$\bar{K}^*$	$\bar{K}_1$	$\bar{K}_4$	$\bar{K}_7$	$\bar{K}_{\text{final}}$
1	-1.0020	-4.4780	-1.0060	-0.9990	-0.9990
2	-8.0648	-35.9328	-8.2174	-8.1366	-8.1366
3	-27.9541	-132.279	-29.2606	-28.9194	-28.9194
4	-53.1828	-258.913	-58.0972	-57.3435	-57.3435
5	-57.1006	-303.308	-68.8420	-67.7998	-67.7998
6	-29.2888	-183.885	-45.7229	-44.8820	-44.8820
7	-0.8097	-78.9065	-16.7416	-16.3371	-16.3371
8	3.3166	6.8943	3.3240	3.3218	3.3218
9	-1.0000	-1.7566	-0.9967	-0.9967	-0.9967
10	-8.3166	-16.0147	-8.3191	-8.3167	-8.3167
11	-30.5851	-61.6190	-30.5799	-30.5625	-30.5625
12	-65.1749	-143.970	-65.3681	-65.2852	-65.2852
13	-88.1604	-211.852	-89.0159	-88.8199	-88.8199
14	-76.8653	-208.234	-79.5985	-79.2984	-79.2984
15	-40.2356	-115.686	-44.0130	-43.7603	-43.7603
16	-9.4476	-45.3034	-14.7672	-14.6267	-14.6267

Table 7.7: Comparison of gain vector $\bar{K}$ values at selected iterations and the optimal vector $\bar{K}^*$ obtained using the LQR controller.

The convergence trend of the reduced-dimension model-free output feedback using the LQR controller is illustrated in Figure 7.6 and summarized in Table 7.7. These results highlight the evolution of the gain vector across iterations and demonstrate the stability and effectiveness of the proposed model-free control scheme.

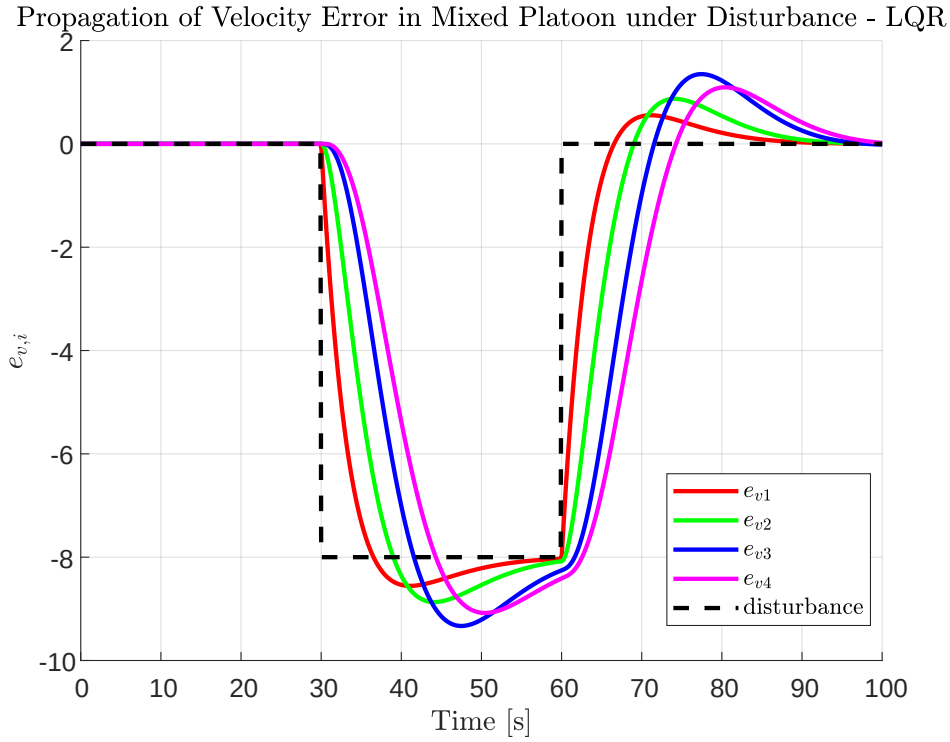

Figure 7.7: Velocity error response of the mixed platoon to a velocity disturbance applied to the lead vehicle, under LQR control. The dashed black line represents the injected disturbance, while the colored curves correspond to the velocity errors of HDVs and CAV.

Figure 7.7 illustrates the velocity error response of a mixed platoon to a velocity disturbance applied to the lead vehicle, under LQR control. The dashed black line indicates the injected disturbance, while the colored curves represent the velocity errors of the HDVs and CAV. As shown, velocity errors increase along the HDVs, and a shift in error behavior is observed at the CAV toward the end of the platoon.

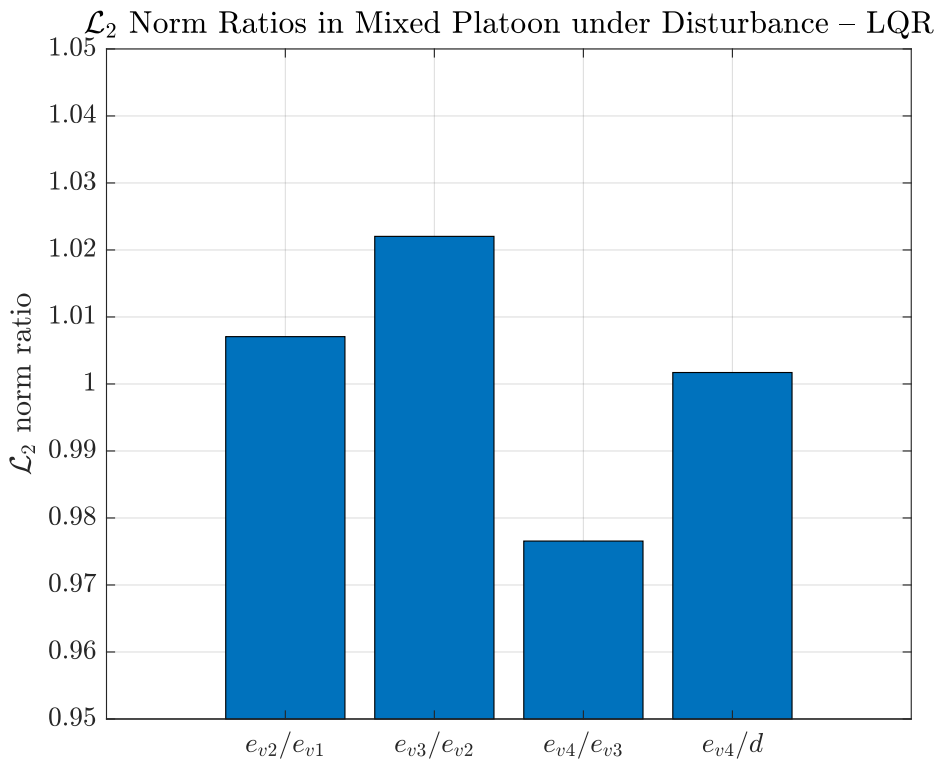

Figure 7.8: L_2-norm ratios of velocity errors under LQR control, including ratios between HDVs, between the CAV and its predecessor, and between the CAV and the external disturbance.

Figure 7.8 illustrates the L_2-norm ratios of velocity errors across the platoon. Specifically, the figure presents the ratio between each HDV and its predecessor, as well as the ratio between the CAV and both its immediate HDV predecessor and the external disturbance applied to the lead vehicle.

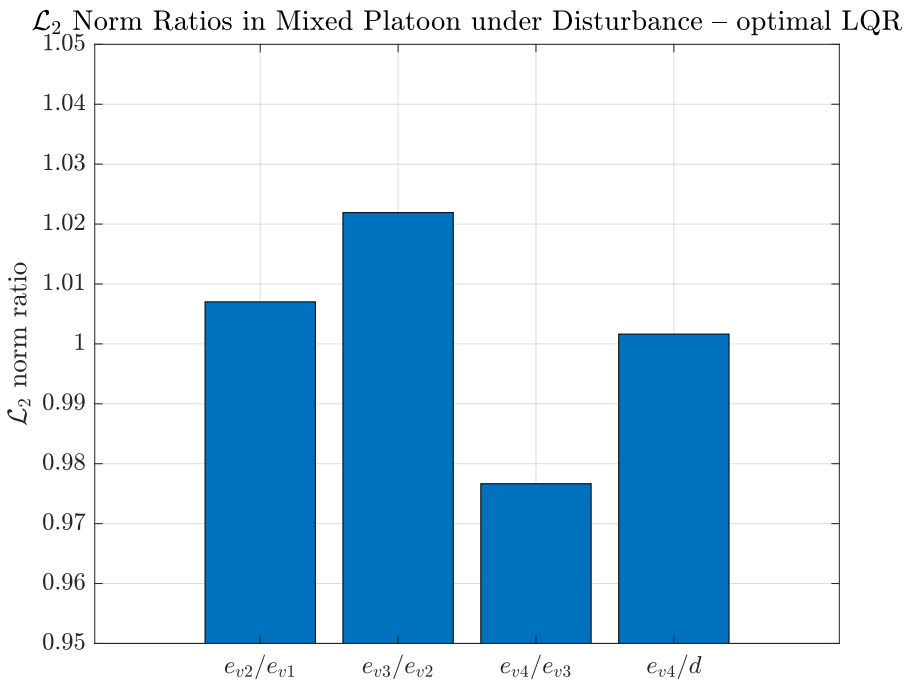

Figure 7.9: L_2-norm ratios of velocity errors under optimal LQR control, including ratios between HDVs, between the CAV and its predecessor, and between the CAV and the external disturbance.

The gain values used in the previous simulations were suboptimal, due to the use of output feedback for learning (see Table 7.7). Therefore, it is reasonable to assume that this suboptimality influences the obtained performance. To examine this, we computed the optimal gain vector based on the solution of the ARE (4.12). Using these optimal LQR gains, the resulting performance is presented in Figure 7.9.

7.1.6 Discussion of the results

1. The results summarized in Table 7.1 and Table 7.2 demonstrate the convergence characteristics of the $\mathcal{H}_\infty$ policy iteration (PI) algorithm. While the initial iterations exhibit substantial deviations from the optimal gain $\bar{K}^*$, with relative errors exceeding 500% in some components, a rapid reduction in error is observed as the iterations progress. By the seventh iteration, most components achieve a relative error of less than 5%, as shown in Table 7.2. Furthermore, the marginal difference between iterations 10 and 11, evident from the near-identical gain values, indicates that the algorithm reaches near-optimal performance within a limited number of steps. These results highlight the stability, efficiency, and fast convergence of the proposed reduced-dimension $\mathcal{H}_\infty$ PI method. The convergence trend shown in Figure 7.1 highlights the effectiveness of the $\mathcal{H}_\infty$ PI algorithm in computing near-optimal gains in a small number of iterations. The steep drop in relative error during the first few steps reflects the rapid adjustment of control gains in the optimal direction. From iteration 5 onward, the error approaches zero, suggesting that the algorithm has entered a refinement phase with diminishing changes. This behavior confirms both the numerical stability and the efficiency of the algorithm, especially in scenarios where fast convergence is desirable.

2. The results shown in Figure 7.2 and Tables 7.3–7.4 demonstrate the efficiency of the $\mathcal{H}_\infty$ PI algorithm in the two-HDV scenario. The initial sharp decline in error emphasizes the ability of the algorithm to quickly align control gains with the dynamics of the system. As the error stabilizes, further refinement of the gain vector is primarily limited by numerical sensitivity and model uncertainty. In particular, components with smaller magnitudes in $\bar{K}^*$ exhibit slower convergence, a behavior consistent with lower observability or increased sensitivity to noise. These insights suggest that, while the bulk of convergence is achieved quickly, achieving exact alignment requires additional iterations that may have diminishing practical returns. From an application standpoint, the method shows strong potential for rapid and robust adaptation in mixed-traffic platoons.

3. Despite using sufficiently large data sets and applying various exploration noises with differing frequencies and amplitudes, the reduced-dimension, model-free output feedback

algorithm did not yield a system of fully linearly independent equations. Interestingly, even when the rank (see (5.98) did not reach the needed 344 linearly independent equations, the algorithm still converged and produced satisfactory results. These findings suggest a potential limitation when scaling the algorithm to large systems with increased uncertainty, such as those involving multiple HDVs. In such scenarios, the presence of numerous HDVs introduces significant variability in behavior, making it more difficult to ensure complete excitation and full-rank conditions.

Table 7.5 presents the values of the gain vector $\bar{K}_i$ at iterations 1, 4, 7, and 13, compared to the optimal gain $\bar{K}^*$. A significant improvement in the gain estimates is observed between the initial iterations, particularly between iterations 1 and 7. By iteration 13, the majority of the components have converged very close to their optimal values. This convergence trend is quantitatively supported by Table 7.6, which shows the relative error percentages. It can be seen that after iteration 13, most relative errors are reduced to a few percent or less, except for some components with initially very large discrepancies. Figure 7.3 further illustrates the overall convergence behavior, plotting the normalized gain error in iterations. Despite some oscillations in the early iterations, the learning process stabilizes quickly, demonstrating the robustness and efficiency of the proposed $\mathcal{H}_\infty$ PI method in the more complex case of three HDVs.

4. Figure 7.4 illustrates the velocity error profiles of the vehicles in the mixed platoon under an external disturbance applied to the leading vehicle. It can be seen that the disturbance is amplified in the first three vehicles, all of which are HDVs. This pattern is consistent with the limited disturbance rejection capabilities typical of human drivers. However, a notable attenuation is observed in the fourth vehicle, which is a CAV.

   Figure 7.5 shows the corresponding L_2 norms of the velocity error, comparing the magnitude of the external disturbance with the response of the CAV. As the L_2 norm ratio is found to be less than one, this confirms that the system satisfies the condition for the stability of *weak string stability.* This holds true even though the optimal gain values were not perfectly attained due to limitations in the rank condition discussed earlier. These results underscore the crucial role of the CAV in attenuating upstream disturbances and enhancing the overall stability of the mixed platoon.

5. In Section 7.1.5, we compared the performance of the $\mathcal{H}_\infty$ controller with that of the LQR. Although LQR offers optimality with respect to a predefined cost function, it does not inherently account for disturbance attenuation (that is, ensuring $\gamma < 1$).

   To enable a fair comparison and to ensure adequate excitation of the platoon dynamics, a modified state-space formulation was used for the LQR scenario. In particular, applying the input only through the last CAV was insufficient to excite all the system modes. To overcome this limitation, we adopted the modeling strategy proposed in [10], in which an input of artificial disturbance is added to the system. This input was used solely to excite the dynamics and was not considered in the controller design, thereby preserving the classical LQR synthesis structure.

6. In this section, the results of the model-free output-feedback algorithm under LQR-based control were evaluated using the suggested reduced-dimension output-feedback technique. Table 7.7 presents the values of the gain vector K_i in several iterations, compared to the optimal gain vector K^*, which was calculated analytically by traditional LQR synthesis. It can be observed that, after only four iterations, the majority of the gain components converge closely to their optimal values.

   This convergence trend is further illustrated in Figure 7.6, where the normalized error between K_i and K^* decreases sharply during initial iterations and then flattens as the gain vector approaches its final form. Rapid drop in error highlights the stability and efficiency of the learning process under LQR-based conditions.

   The experimental conditions used here were kept consistent with those used for the $\mathcal{H}_\infty$ controller, including the structure of the reduced-dimension output matrix C_{new} and the learning parameters. This consistency demonstrates that the algorithm achieves robust and efficient convergence across different control frameworks, reinforcing its suitability for data-driven control in various settings.

7. Figure 7.7 illustrates the propagation of velocity error in a mixed platoon under LQR control, following a disturbance injected into the lead vehicle. As seen, the error is amplified through HDVs, an expected outcome given the absence of coordination mechanisms in human-driven vehicles. Nevertheless, the velocity error is not amplified in the last vehicle

(a CAV), and a degree of attenuation is observed. This behavior initially suggests that the CAV contributes to disturbance damping within the platoon.

However, further insight is gained from Figure 7.8, which shows the L_2 norm ratios of velocity errors between consecutive vehicles and between the CAV and the external disturbance. Although the ratio e_{v_4}/e_{v_3} is less than one, indicating the local attenuation by the CAV relative to its HDV predecessor, the ratio e_{v_4}/d is greater than one. This means that, globally, the disturbance propagated from the lead vehicle to the CAV was not attenuated below its original magnitude. Consequently, although the CAV helps limit the local amplification of errors, the system as a whole does not satisfy the condition for *weak string stability* based on the L_2 norm criterion. This result highlights the limitations of LQR in ensuring disturbance attenuation in mixed platoons without explicitly incorporating a disturbance attenuation criterion.

8. To address the concern that the absence of *weak string instability* observed under the LQR control may be attributed to inaccuracies in the estimated gain vector K, a complementary test was performed using the exact optimal LQR gains K^*. In this scenario, a velocity disturbance was applied to the lead vehicle, and the L_2 norm ratios were calculated to evaluate the propagation of the disturbance throughout the platoon. As illustrated in Figure 7.9, the ratio between the disturbance and the velocity error of the last vehicle (CAV) remains greater than one. This confirms that even with precise optimal gains K^*, *weak string stability* is not achieved.

Chapter 8

Conclusions

This work presented a model-free control framework for mixed platoons using a reduced-dimension output-feedback approach. The proposed method demonstrated the ability to learn effective optimal control policies from input-output data without requiring the full state measurement and the model of the system. Through extensive simulations, we showed that the algorithm converges rapidly under both $\mathcal{H}_\infty$ and LQR control objectives, accurately reproducing the optimal gains under reduced computational complexity.

Before presenting the main simulation results, the reduced-dimension technique was preliminarily evaluated on two representative systems using standard LQR control strategies. The findings consistently indicated that reducing the dimensionality of the system output for the learning process led to a notable reduction in computational effort without compromising the quality of convergence or control performance. Furthermore, it was observed that the reduced dimension approach effectively enriches the system identification process (compared to the case of fewer measurements) by introducing more informative data. Consequently, the learning algorithm achieved convergence in fewer iterations. These results suggest that the method can efficiently extract relevant information even from partially observed systems, making it a promising tool for large-scale applications, such as long-vehicle platoons.

Importantly, analysis of L_2 norm propagation confirmed that only the $\mathcal{H}_\infty$ based approach consistently attenuates disturbances across the platoon, achieving *weak string stability* even in the presence of multiple human-driven vehicles (HDVs). In contrast, the LQR controller, despite being optimal with respect to its cost function, failed to suppress disturbances when

evaluated through L_2 performance metrics.

Future Work. While the proposed algorithm successfully converged to near-optimal gain values in mixed platoon scenarios, certain limitations remain unresolved. Notably, in cases involving a large number of HDVs, the learning process was hindered by rank deficiency in the resulting system of equations, indicating linear dependence and incomplete mode excitation. This constraint limited the size and complexity of the platoon models that could be effectively handled. Future studies should investigate the origin of these dependencies and propose methods to ensure full-rank systems, which would enable the extension of the framework to larger heterogeneous platoons.

Once the rank deficiency issue is resolved, the framework could be extended with a method to estimate the number of HDVs between adjacent CAVs, as proposed in [3]. This would make it possible to accommodate significantly larger platoons, including many HDVs, and thus improve the scalability of the proposed control strategy.

In addition, preliminary experiments were conducted using standard LQR controllers to evaluate the effect of the reduced-dimension technique on two representative systems. These tests showed that for the case of multiple measurements, reducing the dimension of y (to a scalar) slightly accelerated convergence (compared to the case of a single measurement), although the improvement was not statistically significant. However, in contrast to PI, learning-based algorithms such as Value Iteration (VI) typically require a substantially more significant number of iterations to converge. As such, even modest improvements in convergence speed can lead to significant computational savings. This observation suggests that applying dimensionality reduction techniques in learning-based control may be particularly beneficial, especially in large-scale or partially observed systems. Further research is needed to quantify these effects and explore their integration into VI-based optimization frameworks.

Bibliography

[1] Shladover, Steven E., Su, Dongyan, and Lu, Xiao-Yun. Impacts of cooperative adaptive cruise control on freeway traffic flow. *Transportation Research Record: Journal of the Transportation Research Board*, 2324:63–70, 2012. doi: 10.3141/2324-08.

[2] Jiang, Yu and Jiang, Zhong-Ping. *Robust Adaptive Dynamic Programming*. John Wiley & Sons, Hoboken, New Jersey, 2017. ISBN 9781119132646.

[3] Orki, Omer Cohen. *Control of Mixed Platoons Consisting of Automated and Manual Vehicles*. Ph.d. thesis, Ben-Gurion University of the Negev, Beer-Sheva, Israel, March 2021.

[4] Orki, Omer and Arogeti, Shai. Control of mixed platoons consist of automated and manual vehicles. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 1–6. IEEE, 2019. doi: 10.1109/ITSC.2019.8917001. URL `https://ieeexplore.ieee.org/document/8917001`. Authorized licensed use limited to: Ben-Gurion University of the Negev.

[5] Shladover, Steven E., Desoer, Charles A., Hedrick, J. Karl, Tomizuka, Masayoshi, Walrand, Jean, Zhang, Wei-Bin, McMahon, Donn H., Peng, Huei, Sheikholeslam, Shahab, and McKeown, Nick. Automatic vehicle control developments in the path program. *IEEE Transactions on Vehicular Technology*, 40(1):114–130, 1991. doi: 10.1109/25.69966.

[6] Vahidi, Ardalan and Sciarretta, Antonio. Energy saving potentials of connected and automated vehicles. *Transportation Research Part C: Emerging Technologies*, 95:822–843, 2018. doi: 10.1016/j.trc.2018.09.001.

[7] Zhang, Yuqin, Xu, Zhihang, Wang, Zijian, Yao, Xinpeng, and Xu, Zhigang. Impacts of communication delay on vehicle platoon string stability and its compensation strategy:

A review. *Journal of Traffic and Transportation Engineering (English Edition)*, 10(4): 508–529, 2023. doi: 10.1016/j.jtte.2023.04.004.

[8] Monteil, J., Bouroche, M., and Leith, D. J. $\mathcal{L}_2$ and $\mathcal{L}_\infty$ stability analysis of heterogeneous traffic with application to parameter optimisation for the control of automated vehicles. Unpublished.

[9] Ge, Jin I. and Orosz, Gábor. Optimal control of connected vehicle systems with communication delay and driver reaction time. *IEEE Transactions on Intelligent Transportation Systems*, 18(8):2056–2067, 2017. doi: 10.1109/TITS.2016.2633164.

[10] Liu, Tong, Cui, Leilei, Pang, Bo, and Jiang, Zhong-Ping. Learning-based control of multiple connected vehicles in the mixed traffic by adaptive dynamic programming. *IFAC-PapersOnLine*, 54(14):370–375, 2021. doi: 10.1016/j.ifacol.2021.10.382.

[11] Rizvi, Syed Ali Asad and Lin, Zongli. Reinforcement learning-based linear quadratic regulation of continuous-time systems using dynamic output feedback. *IEEE Transactions on Cybernetics*, 50(11):4670–4679, 2020. doi: 10.1109/TCYB.2018.2886735.

[12] Rizvi, Syed Ali Asad and Lin, Zongli. Output feedback adaptive dynamic programming for linear differential zero-sum games. *Automatica*, 122:109272, 2020. doi: 10.1016/j.automatica.2020.109272.

[13] Ahmed, Hafiz Usman, Huang, Ying, Lu, Pan, and Bridgelall, Raj. Technology developments and impacts of connected and autonomous vehicles: An overview. *Smart Cities*, 5(1):382–404, 2022. doi: 10.3390/smartcities5010022. URL `https://www.mdpi.com/2624-6511/5/1/22`.

[14] National Highway Traffic Safety Administration (NHTSA). Automated vehicles for safety—the evolution of automated safety technologies. `https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety`, 2022. Accessed: 14 March 2022.

[15] Gunter, George, Gloudemans, Derek, Stern, Raphael E., McQuade, Sean, Bhadani, Rahul, Bunting, Matt, Delle Monache, Maria Laura, Lysecky, Roman, Seibold, Benjamin, Sprinkle, Jonathan, Piccoli, Benedetto, and Work, Daniel B. Are commercially implemented

adaptive cruise control systems string stable? *IEEE Transactions on Intelligent Vehicles*, 2020. URL `https://arxiv.org/abs/1905.02108`. Available as arXiv preprint: arXiv:1905.02108.

[16] Shladover, Steven E. Connected and automated vehicle systems: Introduction and overview. *Journal of Intelligent Transportation Systems*, 22(3):190–200, 2018. doi: 10.1080/15472450.2017.1336053. URL `https://doi.org/10.1080/15472450.2017.1336053`.

[17] Milanés, Vicente, Shladover, Steven E., Spring, John, Nowakowski, Christopher, Kawazoe, Hiroshi, and Nakamura, Masahide. Cooperative adaptive cruise control in real traffic situations. *IEEE Transactions on Intelligent Transportation Systems*, 15(1):296–305, 2014. doi: 10.1109/TITS.2013.2278494.

[18] van Arem, Bart, van Driel, Cornelie J. G., and Visser, Ruben. The impact of cooperative adaptive cruise control on traffic-flow characteristics. *IEEE Transactions on Intelligent Transportation Systems*, 7(4):429–436, 2006. doi: 10.1109/TITS.2006.884615.

[19] Gong, Siyuan and Du, Lili. Cooperative platoon control for a mixed traffic flow including human drive vehicles and connected and autonomous vehicles. *Transportation Research Part B: Methodological*, 116:25–61, 2018. doi: 10.1016/j.trb.2018.07.005. URL `https://doi.org/10.1016/j.trb.2018.07.005`.

[20] Minsky, Marvin L. *Theory of Neural-Analog Reinforcement Systems and Its Application to the Brain Model Problem*. Ph.d. thesis, Princeton University, 1954.

[21] Sutton, Richard S., Barto, Andrew G., and Williams, Ronald J. Reinforcement learning is direct adaptive optimal control. *IEEE Control Systems Magazine*, 12(2):19–22, 1992. doi: 10.1109/37.126844.

[22] Lewis, Frank L. and Vrabie, Draguna. Reinforcement learning and adaptive dynamic programming for feedback control. *IEEE Circuits and Systems Magazine*, 9(3):32–50, 2009. doi: 10.1109/MCAS.2009.933854.

[23] Vrabie, Draguna, Vamvoudakis, Kyriakos G., and Lewis, Frank L. *Optimal Adaptive*

Control and Differential Games by Reinforcement Learning Principles. Institution of Engineering and Technology (IET), London, 2013. ISBN 9781849196503.

[24] Barto, Andrew G., Sutton, Richard S., and Anderson, Charles W. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, 13(5):834–846, 1983. doi: 10.1109/TSMC.1983.6313077.

[25] Sutton, Richard S. Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44, 1988. doi: 10.1007/BF00115009.

[26] Bellman, Richard E. *Dynamic Programming.* Princeton University Press, Princeton, NJ, 1957.

[27] Lewis, Frank L. and Syrmos, Vassilis L. *Optimal Control.* Wiley, New York, NY, USA, 1995.

[28] Kleinman, David L. On an iterative technique for riccati equation computations. *IEEE Transactions on Automatic Control*, 13(1):114–115, 1968. doi: 10.1109/TAC.1968.1098797.

[29] Bradtke, Steven J., Ydstie, B. Erik, and Barto, Andrew G. Adaptive linear quadratic control using policy iteration. In *Proceedings of the American Control Conference*, pages 3475–3479, Baltimore, MD, USA, 1993. IEEE.

[30] Jiang, Yu and Jiang, Zhong-Ping. Robust approximate dynamic programming and global stabilization with nonlinear dynamic uncertainties. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pages 115–120, Orlando, FL, USA, 2011. IEEE.

[31] Jiang, Yu and Jiang, Zhong-Ping. Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics. *Automatica*, 48(11):2699–2704, 2012. doi: 10.1016/j.automatica.2012.06.096.

[32] Jiang, Yu and Jiang, Zhong-Ping. Value iteration and adaptive optimal control for linear continuous-time systems. In *2015 IEEE 7th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mecha-*

tronics (RAM), pages 54–58, Angkor Wat, Cambodia, 2015. IEEE. doi: 10.1109/CIS-RAM.2015.7405597.

[33] Fu, Yue, Fu, Jun, and Chai, Tianyou. Robust adaptive dynamic programming of two-player zero-sum games for continuous-time linear systems. *IEEE Transactions on Neural Networks and Learning Systems*, 26(12):3314–3325, 2015. doi: 10.1109/TNNLS.2015.2461452.

[34] Li, Hongliang, Liu, Derong, and Wang, Ding. Integral reinforcement learning for linear continuous-time zero-sum games with completely unknown dynamics. *IEEE Transactions on Automation Science and Engineering*, 11(3):706–714, 2014. doi: 10.1109/TASE.2014.2300532.

[35] Lewis, Frank L. and Vamvoudakis, Kyriakos G. Reinforcement learning for partially observable dynamic processes: Adaptive dynamic programming using measured output data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(1):14–25, 2011. doi: 10.1109/TSMCB.2010.2043839.

[36] Postoyan, Romain, Buşoniu, Lucian, Nešić, Dragan, and Daafouz, Jamal. Stability analysis of discrete-time infinite-horizon optimal control with discounted cost. *IEEE Transactions on Automatic Control*, 62(6):2736–2751, 2017. doi: 10.1109/TAC.2016.2616644.

[37] Rizvi, Syed Ali Asad and Lin, Zongli. Output feedback reinforcement q-learning control for the discrete-time linear quadratic regulator problem. In *Proceedings of the 56th IEEE Conference on Decision and Control (CDC)*, pages 1311–1316, Melbourne, Australia, 2017. IEEE. doi: 10.1109/CDC.2017.8263794.

[38] Rizvi, Syed Ali Asad and Lin, Zongli. Output feedback q-learning for discrete-time linear zero-sum games with application to the h-infinity control. *Automatica*, 95:213–221, Sep 2018. doi: 10.1016/j.automatica.2018.05.031.

[39] Treiber, Martin and Kesting, Arne. Evidence of convective instability in congested traffic flow: A systematic empirical and theoretical investigation. *Transportation Research Part B: Methodological*, 45(9):1362–1377, November 2011. doi: 10.1016/j.trb.2011.05.006.

[40] Treiber, Martin and Kesting, Arne. *Traffic Flow Dynamics: Data, Models and Simulation.*

Springer, Berlin, Heidelberg, 2013. ISBN 978-3-642-32460-4. doi: 10.1007/978-3-642-32460-4. URL https://doi.org/10.1007/978-3-642-32460-4.

[41] Wilson, R.E. and Ward, J.A. Car-following models: fifty years of linear stability analysis–a mathematical perspective. *Transportation Planning and Technology*, 34(1):3–18, 2011. doi: 10.1080/03081060.2011.530826. URL https://doi.org/10.1080/03081060.2011.530826.

[42] Li, S. E., Zheng, Y., Li, K., and Wang, J. An overview of vehicular platoon control under the four-component framework. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 286–291, Seoul, Korea, June 2015.

[43] Ailon, A. and Arogeti, S. String stability of a group of nonholonomic mobile robots whose models incorporate kinematic and dynamic equations of motion. In *Proceedings of the IEEE Chinese Control and Decision Conference (CCDC)*, pages 3062–3068, Chongqing, China, May 2017.

[44] Ploeg, J., Shukla, D. P., van de Wouw, N., and Nijmeijer, H. Controller synthesis for string stability of vehicle platoons. *IEEE Transactions on Intelligent Transportation Systems*, 15: 854–865, April 2014. doi: 10.1109/TITS.2013.2290153.

[45] Sheikholeslam, S. and Desoer, C. A. Longitudinal control of a platoon of vehicles. In *Proceedings of the 1990 American Control Conference*, pages 291–296, San Diego, CA, USA, June 1990.

[46] Klinge, S. and Middleton, R. H. Time headway requirements for string stability of homogeneous linear unidirectionally connected systems. In *Proceedings of the IEEE Conference on Decision and Control (CDC)*, Shanghai, China, December 2009.

[47] Knobloch, H. W., Isidori, A., and Flikcerzi, D. *Topics in Control Theory.* Springer-Verlag, Berlin, 1993.

[48] Zames, George. Feedback and optimal sensitivity: model reference transformations, multiplicative seminorms and approximate inverses. *IEEE Transactions on Automatic Control*, 26(2):301–320, 1981.

[49] van der Schaft, Arjan J. l_2-gain analysis of nonlinear systems and nonlinear state feedback h_∞ control. *IEEE Transactions on Automatic Control*, 37(6):770–784, 1992.

[50] Lanzon, Alexander, Feng, Yantao, Anderson, Brian D. O., and Rotkowitz, Michael. Computing the positive stabilizing solution to algebraic riccati equations with an indefinite quadratic term via a recursive method. *IEEE Transactions on Automatic Control*, 53(10): 2280–2291, 2008. doi: 10.1109/TAC.2008.2006108.

בקרה מבוססת פלט ללא ידע מוקדם של המודל, עם יישומים בשיירות כלי רכב מעורבות

אלירן אלבז

בהנחיית פרופ' שי ארוגטי

עבודת מחקר לתואר מוסמך להנדסה

אוניברסיטת בן־גוריון בנגב 2025

תקציר

מחקר זה מתמקד בפיתוח ושיפור של שיטות בקרה לשיירות מעורבות המורכבות מכלי רכב אוטונומיים וכלי רכב עם נהג אנושי. מטרת המחקר העיקרית היא להבטיח יציבות שרשרת חלשה וביצועים דינמיים נאותים, גם בתנאים של חוסר ודאות וחוסר ידע מלא על המערכת.

במסגרת המחקר מוצעת מסגרת בקרה חדשנית המבוססת על משוב־יציאה לשיירות מעורבות, תוך שימוש בשיטת תכנות דינמי אדפטיבי. שיטה זו מאפשרת ללמוד חוקי בקרה אופטימליים ישירות מנתוני קלט־פלט, ללא צורך בידע מלא על דינמיקת המערכת. האלגוריתם שופר על ידי הפחתה משמעותית במספר פרמטרי הבקרה שיש לאמוד, דבר שהוביל לצמצום דרישות החישוב והפך את הגישה לישימה גם במערכות בקנה מידה גדול, כגון שיירות ארוכות של כלי רכב.

תכונה מרכזית של המסגרת המוצעת היא היכולת להבטיח יציבות שרשרת חלשה גם בהיעדר היכולת לבקר את הנהגים האנושיים במערכת. תכונה זו אינה נובעת ישירות מהאלגוריתם הלומד, אלא מגישה המבוססת על בקרה מסוג H_∞, שבה קריטריון יציבות השיירה ממומש בתהליך התכן.

תוצאות הסימולציה מדגימות את יכולת האלגוריתם ללמוד חוקי בקרה יציבים ואופטימליים תוך השגת הביצועים הנדרשים בשיירה המעורבת. בפרט, מוצגת יכולת המערכת לפצות על חוסר היציבות שמאפיין נהגים אנושיים המשתתפים בשיירה.

מחקר זה תורם לקידום מערכות תחבורה חכמות ובטוחות, במיוחד בסביבות תנועה הטרוגניות. בנוסף, המסגרת המוצעת תומכת בצמצום התפשטות העומסים לאורך השיירה, ובכך משפרת את יציבות התחבורה ויעילותה הכוללת.

אוניברסיטת בן־גוריון בנגב
הפקולטה להנדסה
המחלקה להנדסת מכונות

בקרה מבוססת פלט ללא ידע מוקדם של המודל, עם יישומים בשיירות כלי רכב מעורבות

חיבור זה מהווה חלק מהדרישות לקבלת התואר מוסמך להנדסה (M.Sc)

בהנחיית פרופ' שי ארוגטי

חתימת הסטודנט: _____ תאריך: _____

חתימת המנחה: _____ תאריך: _____

חתימת יו"ר ועדת הלימודים לתארים מתקדמים: _____ תאריך: _____

מרץ 2025

אוניברסיטת בן־גוריון בנגב
הפקולטה להנדסה
המחלקה להנדסת מכונות

בקרה מבוססת פלט ללא ידע מוקדם של המודל, עם יישומים בשיירות כלי רכב מעורבות

חיבור זה מהווה חלק מהדרישות לקבלת התואר מוסמך להנדסה (M.Sc)

אלירן אלבז

בהנחיית פרופ' שי ארוגטי

מרץ 2025