Author:
Chen, Anthony Siming

Title:
Adaptive Optimal Control via Reinforcement Learning

Theory and Its Application to Automotive Engine Systems

Adaptive Optimal Control via Reinforcement Learning:

Theory and Its Application to Automotive Engine Systems

By

ANTHONY SIMING CHEN

Department of Mechanical Engineering

UNIVERSITY OF BRISTOL

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of DOCTOR OF PHILOSOPHY in the Faculty of Engineering.

DECEMBER 2021

Word count: 40,812

Abstract

This thesis presents a new adaptive optimal control framework for continuous-time nonlinear input-affine systems. The idea of combining adaptive control and optimal control has emerged recently due to the advancement in one class of machine learning: reinforcement learning. The topic is also known as the approximate/adaptive dynamic programming (ADP) which is often formulated in discrete time or as the Markov decision process (MDP). This work, for the first time, extends the idea of linear discrete-time Q-learning to a nonlinear continuous-time adaptive optimal control algorithm that runs without stepwise iterations. A particular focus of the research is on the automotive engine application with the objective of developing highly-integrated and complex propulsion technology of the future, accounting for sustainability of future transport technology, i.e. emission reduction and optimised energy and power use. Hence, the thesis comprises two parts:

The theoretical work is driven by the development of reinforcement learning and ADP, where a novel online Q-learning algorithm is proposed to approximately solve the optimal control problem in real time using a new adaptive critic neural network without the requirement of complete system knowledge. The finite-time convergence of the value function approximation is guaranteed by using a sliding-mode technique while the persistent excitation (PE) condition of the state trajectories can be verified directly in real time. Furthermore, the proposed Q-learning approach is extended to solve a nonlinear optimal observer design problem, where an observer Hamilton-Jacobi-Bellman (OHJB) equation is obtained. The closed-loop stability is rigorously proved via the Lyapunov analysis and numerical simulations demonstrate the effectiveness of the proposed methods.

The practical work investigates the control problems of a Wankel rotary engine, i.e. air-fuel ratio (AFR) control and idle speed control with the aim of emission reduction and efficiency improvement. An adaptive optimal controller is designed for the idle speed regulation. Two controllers: 1) nonlinear observer-based and 2) Q-learning-based are developed for the AFR. The control system development covers dynamics modelling, calibration, control design/simulation, implementation, and practical experiments. The proposed controllers are successfully applied and validated through a series of simulations and engine tests under different driving cycles.

ACKNOWLEDGEMENTS

I would like to thank my supervisors: Prof. Guido Herrmann, Prof. Chris Brace, and Prof. Stuart Burgess, for all their support and guidance throughout my doctoral research. Prof. Guido Herrmann offered me a great deal of academic insight about nonlinear control as well as patience and trust that helped me get through those difficult times. We share the same passion for theoretical research and work together on promoting practical control applications. I enjoy those moments when we were cracking equations on a white board, discussing ideas on a train or a bus, or brainstorming together in a zoom meeting with comfortable silence. His mentorship is unique and so valuable, for which I will always be grateful. Prof. Chris Brace created many collaboration opportunities and provided me a visiting position at the Institute of Advanced Automotive Propulsion Systems (IAAPS), University of Bath, through which I grew my professional networks and got to know many talented researchers and engineers from the automotive industry. Prof. Stuart Burgess has constantly provided me with encouragement and useful feedback. I will always remember those coffee chats at the table outside the Hawthorns cafe and I hope we can continue sharing our thoughts from time to time after I leave Bristol.

I would like to thank the IAAPS and the Department of Mechanical Engineering at the University of Bath for providing the laboratory facilities. I would like to thank my colleagues Matthew Turner, Dr. Giovanni Vorraro, Dr. Reza Islam, Prof. Jamie Turner, Kristina Burke, Shan Bradley-Cong at IAAPS and Nathan Bailey at the Advanced Innovative Engineering (AIE UK) Ltd. for their expertise and continuous support for our engine tests. Thanks also to Jon Mansfield, Mike Skinner, Patrick Simpson, and Oscar Frith-Macdonald at General Engine Management Systems (GEMS) Ltd. for their technical support with regard to hardware issues of engine control systems.

In addition, I would also thank Prof. Jing Na who led me to the door of academia and encouraged me to apply for the PhD scholarship.

Last but not least, I would like to thank my parents, Dr. Ye Chen and Wen Tan, and the family members who have cared and prayed for me. Special thanks to my beloved wife, Teresa Yu Bi, for always being there for me. Also my little dog, Sushi, for the companionship during the writing-up in the pandemic. May God bless you all.

AUTHOR'S DECLARATION

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: ..... DATE: .....

TABLE OF CONTENTS

List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

This thesis presents the development of adaptive optimal control theory via the reinforcement learning philosophy for general continuous-time nonlinear problems. The subject can easily become a highly mathematical discipline but its principles and tools are practically useful to solve real-world problems. In particular, a substantial amount of work was devoted to the application on automotive engine systems. This chapter provides the scope and motivations of the research, followed by the objectives, the methodology, and the thesis outline.

1.1 Research Scope and Motivations

In the field of systems and control, control engineering covers a wide range of subjects and disciplines, and it is valuable not just in its own right, but also in terms of its influence on other areas such as manufacturing, transportation, aerospace, communications, computers, biology, energy, and economics. The design of control systems is the core subject in control engineering studies. A control system should be able to regulate, manage or direct the behaviour of the objective system, i.e. the plant. To be

specific, the control system usually senses the plant's operation, computes the corrective action, and actuates the plant to deliver the desired performance. This closed loop of sensing, computation, and actuation forms the central concept of control: feedback. For feedback control, we care about three major aspects:

- Stability
- Robustness
- Performance

Stability is considered to be the most important property. For a linear system, the output should be bounded given a bounded input, which is known as bounded-input, bounded-output (BIBO) stability. For a nonlinear system, we often use input-to-state stability (ISS), which extends the notion of BIBO stability using Lyapunov stability. **Robustness** refers to the ability of the control system to withstand the uncertainty or the external disturbance (e.g. noise, model uncertainty or parameter variations), where we often look at gain and phase margins, and sensitivity analysis. Robust control and adaptive control are approaches that explicitly deal with uncertainty. **Performance** in this context means that the dynamics of the closed-loop system have additional desired behaviours such as fast responsiveness to changes and good disturbance attenuation. For example, a control engineer may specify the rise time, the settling time, and the overshoot for transient responses and the error and the accuracy for the steady-state responses.

In control engineering, these three properties are established via a variety of modelling and analysis techniques. The subject relies on and shares tools from physics (dynamics, modelling), computer science (information theory, algorithms, and software), and operations research (optimisation, game theory).

The era of Artificial Intelligence (AI): Machine Learning

Being in 2021, it is not possible to talk about the state of the art control engineering without mentioning artificial intelligence (AI). The name of AI was coined at the Dartmouth conference on artificial intelligence in 1956. Since then, the AI industry has experienced several booms to billions of dollars along with two difficult periods of "AI winter" in 1974-1980 and 1987-1993, respectively [5]. Beginning about 2012, the interest in AI and especially machine learning as its sub-field has led to a dramatic increase in funding and investment from research and corporate communities in the last decade. The success of machine learning comes with two major factors: 1) Data: Machine learning needs huge amount of data for training. Firms such as Google, Meta, Microsoft, and Amazon can access a massive amount of data thanks to the Internet and the cloud data storage. 2) Computing power: Rapid improvements in Central Processing Unit (CPUs) and Graphics Processing Unit (GPUs) have allowed neural networks to run at speeds which were not possible a decade ago. Control engineering tools and methods tend to have less computational complexity (low-capability micro-controllers) than some AI applications. The deployment of new AI approaches will contribute to the creation of more capable control systems and applications. This motivates the development in "Intelligent Control" by integrating AI techniques and control theory.

It would be convenient to summarise our research scope as "intelligent control". Nevertheless, the definition of it is somewhat ambiguous, where there are several competing concepts for the terminology such as "adaptive" control and "learning" control [6]. To understand the interdisciplinary subject, we use Fig. 1.1 to explain our research scope as the interaction of the fields of automatic control, operations research, and artificial intelligence. Each field can be extremely broad and there are notable synergies of the tools and techniques between these fields. For example, the AI techniques such as machine learning and artificial neural networks can help the decision-making process in operations research and the control system design in automatic control; the operations research methods such as dynamic programming can be used to compute the optimal policy for reinforcement learning in AI and the optimal controller in auto-

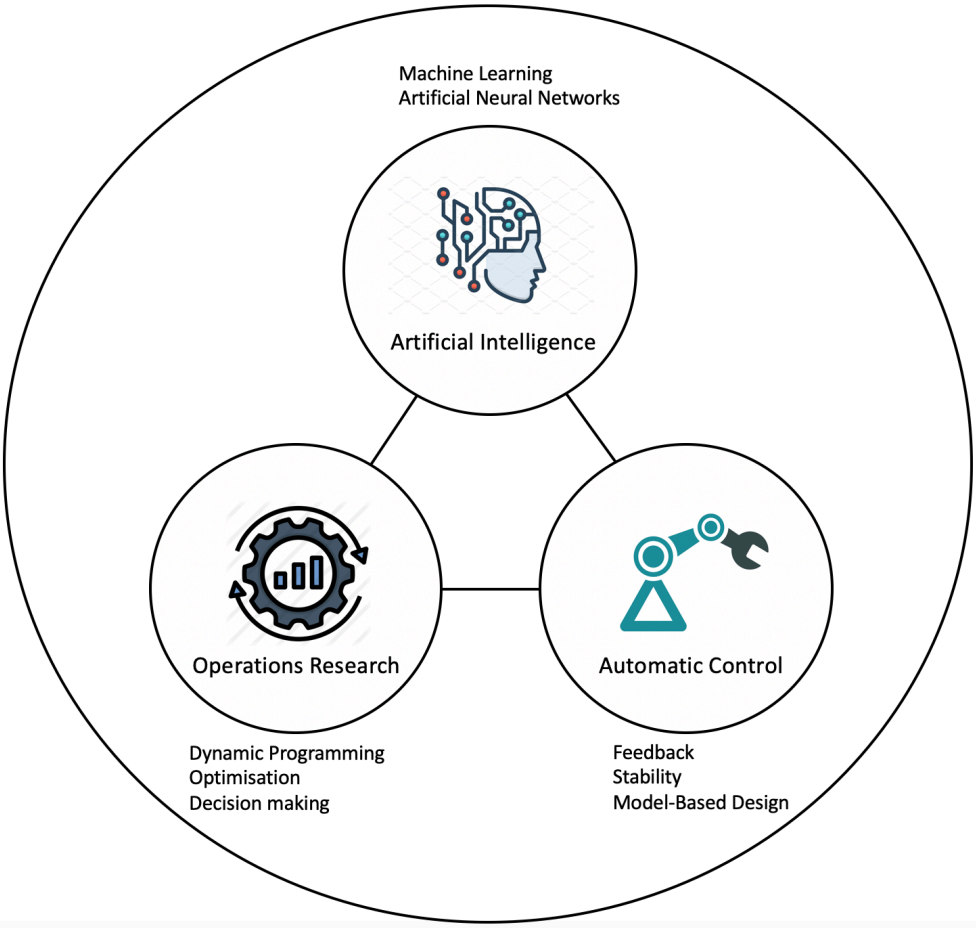

FIGURE 1.1. The interdisciplinary research scope as the interaction of the fields of automatic control, operations research, and artificial intelligence.

matic control; the automatic control principles such as feedback, stability, and model-based design can be particularly useful for scientific machine learning (SciML) and reinforcement learning in AI and decision analysis in operations research.

From "breadth" to "depth":

In terms of theoretical research, we pose the two following questions:

- *What is the underlying connection between control theory and machine learning?*
- *How does machine learning technology help control systems design?*

On the one hand, the use of control theory as a mathematical tool can help formulate and solve machine learning problems such as optimal parameter tuning and neural network training. On the other hand, the use of machine learning as a kernel method or a data-driven approach can numerically solve complex control theory problems that are intractable by analytical methods [7]. We look forward to a synthesis of data-centred machine learning and model-centred control theory in the foreseeable future as our research continues. In this thesis, we will show the elegant equivalence between adaptive optimal control, reinforcement learning, and approximate dynamic programming. This is illustrated in Fig. 1.2 in the context of Fig. 1.1.

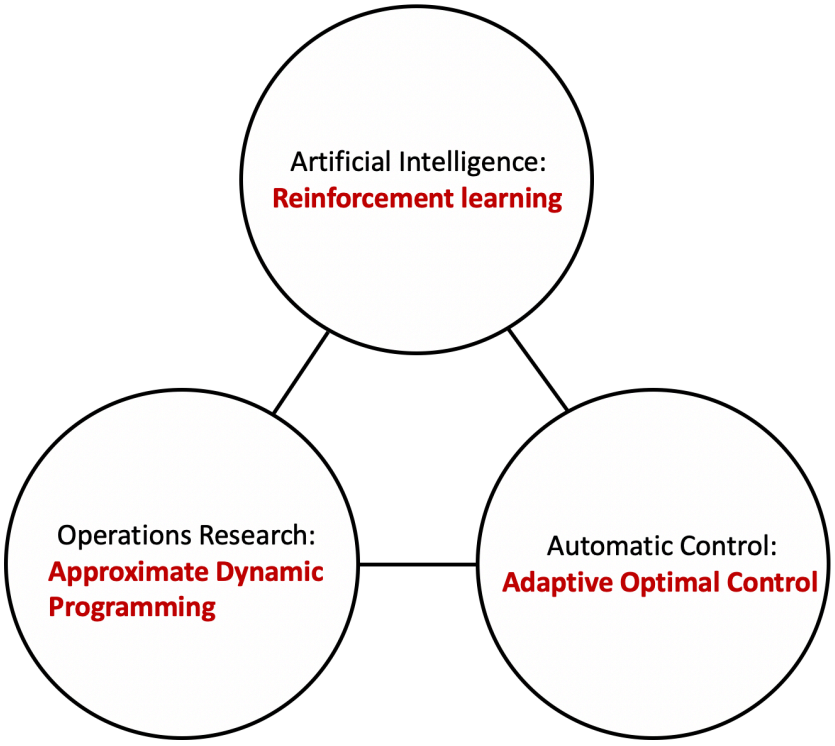

FIGURE 1.2. The equivalence between adaptive optimal control, reinforcement learning, and approximate dynamic programming.

For the practical application, we are interested in the development and implementation of new control systems for automotive powertrain, where the control problems

for internal combustion engines are especially addressed. Therefore, we can raise the two motivating questions:

- *What are the control problems and their general solutions in automotive engine systems?*
- *How do we improve the current engine control strategy?*

1.2 Objectives

In the context of control theory and engineering, we aim to study adaptive optimal control following the scope and motivations in Section 1.1. This subject is positioned as a distinctive intersection that connects control theory, operations research, and AI. We will further describe their underlying relationships in the literature review in Chapter 2. This Ph.D. research is expected to cover both theoretical and practical sides of the subject. We summarise the research objectives into three points:

1. To **formulate** a new adaptive optimal control scheme for continuous-time nonlinear systems in terms of novel control and observation techniques.
2. To **develop** new adaptive optimal control algorithms using reinforcement learning that bring benefits such as adaptation, optimality, model-free learning, etc.
3. To **design** and **implement** new control systems for automotive engines with the aim of technology improvement.

1.3 Research Process

For the theoretical research (*Objective 1 and 2*) on adaptive optimal control, the research process diagram is shown in Fig. 1.3. Beginning with literature review, we investigate not only the state of the art but also cover some of the development history

of the related research topics such as adaptive control, optimal control, robust control, intelligent control, convex optimisation, and reinforcement learning. This can help creating an inclusive framework and seeing a big picture as one can often advance a new theory by absorbing ideas from a seemingly different area, e.g. it has been shown that the optimal control problem can be solved by a critic-actor structure using adaptive control, which was originated in the reinforcement learning studies. After we find the limitation of current methods and the core idea to improve them, a hypothesis can be constructed using the idea, followed by detailed development and analysis. The proposed theory can then be validated using numerical examples and adjustment can be made to rectify the theory.

FIGURE 1.3. Theoretical research methodology diagram.

For control system development (*Objective 3*), we borrow the knowledge from systems engineering. The V-Model is one of the widely-used system development pro-

cesses in the automotive industry. Fig. 1.4 presents the V-Model for engine control systems development in the automotive industry. Our research develops the engine control system via an electronic control unit (ECU) and the plan should follow the industrial-standard development process that includes design, simulation, and implementation.

The V-Model (where V stands for verification and validation) splits the development process into two segments. The left arm of the V consists of the decomposition of the control requirement, analysis and design, software simulation and rapid control prototyping while the right arm concentrates on verification and validation activities such as code generation, calibration, hardware-in-the-loop (HiL) testing and in-vehicle testing. This systematic process is easy to manage and works very well for small projects such as the ECU development.

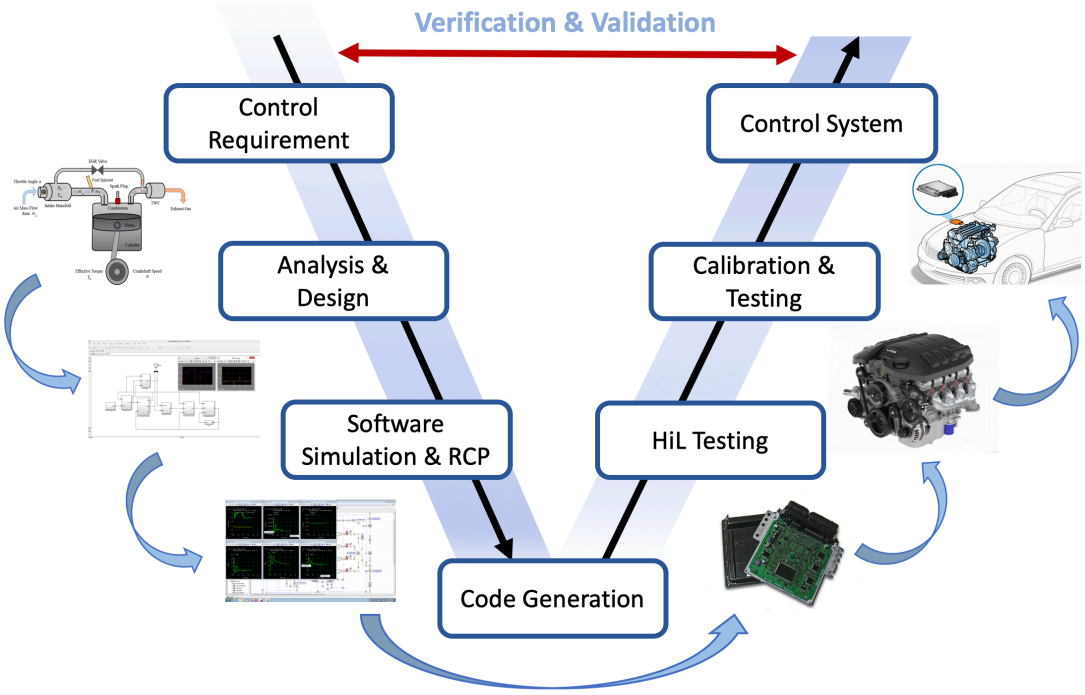

FIGURE 1.4. The V-Model for engine control systems development in the automotive industry.

1.4 Thesis Outline

The Ph.D. thesis presents both the theoretical (Chapter 3-4) and practical results (Chapter 5-8) of the research on adaptive optimal control via reinforcement learning and automotive engine control. Each chapter of 3-8 is supported by a (planned) journal or conference publication.

Chapter 1 provides the introduction of the Ph.D. research, which includes the research scope, motivations, objectives, the research process, and the thesis outline.

Chapter 2 describes the literature review on several important aspects throughout the Ph.D. studies, which covers the fundamental subjects such as modern control theory, reinforcement learning, and in-depth topics on adaptive optimal control and automotive engine control.

Chapter 3 establishes the adaptive optimal control theory which forms the core contribution of the theoretical research. Two online adaptive optimal control algorithms are proposed based on reinforcement learning for continuous-time nonlinear input-affine systems: 1) generalised policy iteration (GPI) and 2) Q-learning. The adaptive critic and actor are continuously and simultaneously updating each other without iterative steps or an initial stabilising policy. The two approaches can online approximate the value functional/Q-functional and are partially/completely model-free. The new adaptive design enables the online verification of the persistent excitation (PE) condition and guarantees the overall closed-loop stability and the finite-time convergence. A detailed mathematical analysis and numerical simulations are provided to show the effectiveness of the algorithms.

Chapter 4 extends the adaptive optimal control results to an optimal observer design problem. An online continuous-time Q-learning algorithm is proposed to solve the optimal observer design problem online while ensuring stability and optimality. We show that the optimal solution can be obtained by approximately solving an observer

Hamilton-Jacobi-Bellman (OHJB) equation. The Q-functional is approximated by an adaptive critic neural network that solves the Q-learning Bellman equation online. A case study on observer design for the Van der Pol oscillator is provided. Numerical simulations demonstrate the effectiveness of the proposed algorithm compared with the high-gain observer.

Chapter 5 investigates the engine dynamics modelling methodology that is particularly suitable for the theoretical studies and new control development for automotive engine systems. As a result, a set of control-oriented models are developed for a 225CS Wankel rotary engine produced by Advanced Innovative Engineering (AIE) UK Ltd. Through a synthesis approach that involves State Space (SS) principles and the artificial Neural Networks (NN), the Wankel engine models are derived by leveraging both first-principle knowledge and engine test data. By using either (or both) physical knowledge or test data, the developed models are able to describe the Wankel engine dynamics with acceptable accuracy. They are all control-oriented models that have less computational demand and should be able to run faster than the available CFD models due to their simplicity.

Chapter 6 develops an output feedback controller based on reinforcement learning for the idle speed regulation problem. The proposed controller is completely model-free and able to learn the optimal control solution online in finite time using only the measurable outputs. The regulation of idle speed can be formulated as an optimal control problem that minimises a pre-defined value function by actuating the throttle angle. Then, we incorporate the extended Kalman filter (EKF) as an optimal reduced-order state observer, which enables the online estimation of the unknown fuel puddle dynamics, to achieve an output feedback idle speed controller. The overall Lyapunov stability is proved and the simulation results of a benchmark engine demonstrate that the proposed controller can effectively regulate the idle speed to a set point under certain load disturbance.

Chapter 7 and 8 present a generic observer-based air-fuel ratio (AFR) control framework for automotive engine systems. Chapter 7 focuses on the nonlinear observer-based design while Chapter 8 develops an adaptive optimal AFR controller. The complex nonlinear air-filling dynamics are lumped together and estimated using novel observer techniques. A newly-proposed unknown input observer is compared with a "dirty" differentiation observer and then employed in the feedback AFR control design. An adaptive optimal AFR controller is developed to achieve both AFR regulation and adaptive-critic learning. Comparative simulations and practical experimental results compared to a benchmark PID controller show that the proposed control can speed up the transient response and regulate the AFR around the stoichiometric value.

Chapter 9 summarises the main contributions of the Ph.D studies and potential directions for future research.

1.5 List of Publications

1. **Chen, A.S.**, Herrmann, G., Na, J., Turner, M., Vorraro, G. and Brace, C., 2018, September. Nonlinear Observer-Based Air-Fuel Ratio Control for Port Fuel Injected Wankel Engines. In 2018 UKACC 12th International Conference on Control (CONTROL) (pp. 224-229). IEEE. (Published)

2. **Chen, A.S.** and Herrmann, G., 2019, December. Adaptive Optimal Control via Continuous-Time Q-Learning for Unknown Nonlinear Affine Systems. In 2019 IEEE 58th Conference on Decision and Control (CDC) (pp. 1007-1012). IEEE. (Published)

3. **Chen, A.S.**, Vorraro, G., Turner, M., Islam, R., Herrmann, G., Burgess, S., Brace, C., Turner, J. and Bailey, N., 2020. Control-oriented modelling of a Wankel rotary engine: a synthesis approach of state space and neural networks (No. 2020-01-0253). SAE Technical Paper. (Published)

4. **Chen, A.S.** and Herrmann, G., Adaptive optimal observer design for deterministic nonlinear systems: A continuous-time Q-learning algorithm. (to be submitted to a high-profile theoretical journal)

5. **Chen, A.S.**, Vorraro, G., Turner, M., Islam, R., Herrmann, G., Burgess, S., Brace, C., Turner, J. and Bailey, N., Air-fuel ratio estimation and control for a Wankel rotary engine with nonlinear observers synthesis: Design and experimental validation. (to be submitted to a high-profile practical journal)

6. **Chen, A.S.**, Herrmann, G., Burgess, S., Brace, C., Adaptive Optimal Air-fuel Ratio Control via Approximate Dynamic Programming for a Wankel rotary engine (to be submitted to a high-profile journal)

7. Islam, A., **Chen, A.S.**, Herrmann, G., A Q-learning approach for the two-player zero-sum game problem for completely unknown continuous-time nonlinear systems (to be submitted to a high-profile conference)

Other publications due to the work before PhD:

1. Harding, T., Rames, C., Teh, H.Y., Mill, T., Na, J., **Chen, A.S.**, and Herrmann, G., 2017. Engine Torque Estimation with Integrated Unknown Input Observer and Adaptive Parameter Estimator. IFAC-PapersOnLine, 50(1), pp.11058-11063. (Published)

2. **Chen, A.S.**, Na, J., Herrmann, G., Burke, R. and Brace, C., 2017, July. Adaptive air-fuel ratio control for spark ignition engines with time-varying parameter estimation. In 2017 9th International Conference on Modelling, Identification and Control (ICMIC) (pp. 1074-1079). IEEE. (Published)

3. Na, J., **Chen, A.S.**, Herrmann, G., Burke, R. and Brace, C., 2017. Vehicle engine torque estimation via unknown input observer and adaptive parameter estimation. IEEE Transactions on Vehicular Technology, 67(1), pp.409-422. (Published)

4. Na, J., **Chen, A.S.**, Huang, Y., Agarwal, A., Lewis, A., Herrmann, G., Burke, R. and Brace, C., 2019. Air-fuel ratio control of spark ignition engines with unknown system dynamics estimator: theory and experiments. IEEE Transactions on Control Systems Technology. (Published)

CHAPTER 2

LITERATURE REVIEW

This chapter provides an extensive literature review on several important aspects throughout the Ph.D. studies. We intend to cover the fundamental topics in control theory and reinforcement learning. We also look into the modelling and control of automotive engines as per theoretical applications. Some topics enclosed in this chapter are particularly useful to our theoretical development while the others will contribute in-principle knowledge or as an informative guide, e.g., the extension of Q-learning from the discrete-time algorithm space to the continuous-time control space will need the knowledge of both reinforcement learning and optimal control. One would readily understand the context and basic concepts of the current research by going through this literature review.

2.1 Modern Control Theory

The subject of control theory deals with the analysis and control design of dynamical systems. In an engineering context, classical control theory uses Laplace transform as a basic tool to model systems as transfer functions and design control systems using

tools such as the root locus method, Routh-Hurwitz stability analysis, frequency approaches including the Nyquist, the Bode, and Nichols methods, phase/gain margin and bandwidth. Nearly all classical methods were developed for linear time-invariant single-input single-output (SISO) systems. Modern control methods, however, use state-space approaches to deal with multiple-input multiple-output (MIMO) systems, because the availability of digital computers made it possible for time-domain analysis and synthesis of complex systems. More history and main methods of control theory can be found in many textbooks, e.g., [8–17]. The three important design philosophies of modern control theory are: adaptive control, optimal control, and robust control. This section will cover a broad range of significant topics that appeared in modern control theory. We focus on the state-of-the-art methodologies and cutting-edge techniques that are quickly advancing and have the potential for developing a robust, optimal, adaptive, practically-feasible control system.

2.1.1 Adaptive Control

The history of adaptive control began in the early 1950s for the design of autopilots for high-performance aircraft and has grown to be one of the richest in terms of algorithms, design techniques, analytical tools, and modifications. It can be summarised as a method using online parameter estimation for controlling linear/nonlinear systems with parametric uncertainty. Core material of analysis and design of nonlinear systems and control can be found in [18, 19].

The basic idea of adaptive control can be explained as follows. Consider the control error e with dynamics given by

$$\dot{e} = f(x) - u \tag{2.1}$$

where u is the control input and $f(x)$ is some unknown nonlinearity of structure

$$f(x) = W^{\mathsf{T}} \phi(x) \tag{2.2}$$

where W is the unknown parameter vector and $\phi(x)$ is a known basis set of regression vectors.

We define the estimate of the nonlinear function $\hat{f}(x) = \hat{W}^\top \phi(x)$ with $\hat{W}$ being a time-varying estimate of the unknown parameter W, which is, for example, updated by

$$\dot{\hat{W}} = \Gamma \phi(x) e^\top \tag{2.3}$$

with an adaptive gain matrix $\Gamma > 0$ to be tuned. This is often called the adaptation (update) law. We can then design the control law as

$$u = \hat{f}(x) + K_p e = \hat{W}^\top \phi(x) + K_p e \tag{2.4}$$

with control gain $K_p > 0$. The closed-loop system becomes

$$\dot{e} = W^\top \phi(x) - u = W^\top \phi(x) - \hat{W}^\top \phi(x) - K_p e = \tilde{W}^\top \phi(x) - K_p e \tag{2.5}$$

with the estimation error $\tilde{W} = W - \hat{W}$. The closed-loop system can be proved to be asymptotically stable via Lyapunov theory [18], i.e. the control error e goes to zero. Moreover, the parameter estimates converge to the actual unknown parameters if an additional *persistence of excitation* condition holds [19].

In general, the methods used for solving adaptive control problems can be broadly classified as

- **Direct adaptive control** in which the estimated parameters are directly used in the adaptive controller, e.g., model reference adaptive control (MRAC) that incorporates a reference model defining desired performance.
- **Indirect adaptive control** in which the estimated parameters are used to calculate the required controller parameters, e.g., model identification adaptive control (MIAC) that performs system identification while running.

There are other adaptive control techniques that might not simply be grouped into one of the above categories but are still practical and powerful, e.g.,

- Adaptive pole placement control [20]: full state feedback control primarily for linear time-invariant systems.
- Extremum-seeking control [21]: model-free real-time optimisation of some performance function using perturbation signals.
- Iterative learning control [22]: control input updated at each repetitive or iterative operation based on the observed error.

More material on the subject of adaptive control can be found in textbooks [23–29]. Many appealing concepts proposed in the notion of feedback systems were also labelled with the prefix of “self-tuning" or “self-adjusting" control [23]. Gain scheduling and other rudimentary model reference schemes were introduced to overcome the sensitivity of a fixed-gain controller to large variations of system parameters [24]. The early famous “MIT rule" of adaptive control that was based on online gradient search can be found in [25]. The input-output stability and the Popov hyperstablity theories were introduced in [26]. The later work on robust adaptive control with modifications was presented in [27]. The $\mathcal{L}_1$ adaptive control theory via the decoupling of adaptation from robustness was given in [28]. A good tutorial of adaptive control without sacrificing mathematical depth and rigour can be referred to [29] for a wide audience.

2.1.2 Optimal Control

Optimal control [30–40] deals with the control design for a given system such that a certain optimality criterion is achieved. In the main, the optimal control problem includes a cost functional V (to be minimised/maximised) that is a function of time, state, and control variables. The optimal control law can be derived by using Pontryagin’s minimum/maximum principle (a necessary condition); or by solving the Hamilton-Jacobi-Bellman (HJB) equation (a necessary and sufficient condition) if certain assumptions hold (e.g., the existence of second-order partial derivatives of V).

The name of the HJB equation was first used by Kalman in the 1960s based on the Hamilton-Jacobi equation in calculus of variations and Bellman's contribution on dynamic programming. It is noted that Pontryagin's maximum principle was being developed in the Soviet Union independently around the same time as Bellman's and Kalman's work on dynamic programming in the United States [38]. There are close connections between the two methodologies. Table 2.1 provides a review of optimal control theory textbooks.

The basic idea of optimal control can be explained by the example of a linear-quadratic regulator (LQR). Consider the continuous-time, linear time-invariant system described by

$$\dot{x} = Ax + Bu \tag{2.6}$$

where x is the measurable system state, u is the control input, and A, B are the plant and input matrices, respectively. Define the infinite horizon integral cost $V(x)$ that has the value associated with an admissible control policy $u = \mu(x)$ given by

$$V^{\mu}(x) := \int_t^{\infty} r(x(\tau), u(\tau))\mathrm{d}\tau \tag{2.7}$$

where the $r = x^{\mathsf{T}}Qx + u^{\mathsf{T}}Ru$ is the quadratic utility with the positive-definite matrices $Q = Q^{\mathsf{T}} \geq 0$, $R = R^{\mathsf{T}} > 0$. Thus, the optimal control problem is to find a control policy $u = \mu(x)$ that minimises the value function, i.e. $V^*(x(t)) \leqslant V^{\mu}(x(t)),\ \forall \mu$ and the optimal control satisfies

$$\mu^* = \arg\min_{\mu} V^{\mu}(x_0) \tag{2.8}$$

Then, the optimal value function can be determined as

$$V^*(x(t)) = \min_{\mu} \int_t^{\infty} r(x(\tau), u(\tau))\mathrm{d}\tau \tag{2.9}$$

If the value function V^{μ} is smooth, one can write an infinitesimal version of (2.7) using Leibniz's formula as

$$\begin{aligned} 0 &= r(x,u) + (\nabla V_x^{\mu})^{\mathsf{T}}(Ax + Bu); \\ V^{\mu}(0) &= 0 \end{aligned} \tag{2.10}$$

Table 2.1: Review of optimal control theory textbooks.

Book	Methodology	Time measure	Nonlinearity	Summary of findings
Lewis *et al.* 2012 [30]	Calculus of variations; Dynamic programming	Continuous; Discrete	Linear; Nonlinear	A comprehensive book that provides a complete picture of optimal control theory in both methods. Extensions include output feedback, reinforcement learning, and differential games.
Bryson and Ho 2018 [31]	Calculus of variations; Dynamic programming	Continuous; Discrete (Markov chain)	Linear; Nonlinear	Optimal control theory in both methods with emphasis on optimisation and variational methods. Important results about the sweep method, random processes, and optimal filtering.
Kirk 2012 [32]	Calculus of variations; Dynamic programming	Continuous	Linear; Nonlinear	A self-contained introductory book that covers both methods and their comparison. Extensions include numerical techniques for trajectory optimisation.
Kwakernaak and Sivan 1972 [33]	Calculus of variations	Continuous; Discrete	Linear	Optimal control methods in the context of modern linear control theory (including the stochastic aspects).
Bertsekas 2005 [34]	Dynamic programming	Continuous; Discrete	Linear; Nonlinear	A leading textbook on the far-ranging algorithmic methodology of dynamic programming. Extensions include neuro-dynamic programming/reinforcement learning.
Athans and Falb 2013 [35]	Calculus of variations	Continuous	Linear; Nonlinear	An introductory textbook on optimal control theory including Pontryagin's maximum principle.
Anderson and Moore 2007 [36]	Dynamic programming	Continuous	Linear	Linear optimal control theory from an engineering point of view. Important results on infinite-horizon linear quadratic regulation/tracking problems; the existence of matrix P; proof of optimality, etc.
Naidu 2002 [37]	Calculus of variations; Dynamic programming	Continuous; Discrete	Linear; Nonlinear	Concise results of optimal control theory with comparison and historical remarks of the two methods. Extensions include constrained optimal control.
Liberzon 2011 [38]	Calculus of variations; Dynamic programming	Continuous	Linear; Nonlinear	Variational methods on a very detailed mathematical level. Important proof for Pontryagin's maximum principle and its relationship to the HJB equation are included.
Vinter 2010 [39]	Calculus of variations; Dynamic programming	Continuous	Linear; Nonlinear	Heavy material on nonsmooth analysis and viscosity methods for optimal control. Necessary conditions for nonconvex problems.
Geering 2007 [40]	Calculus of variations; Dynamic programming	Continuous	Linear; Nonlinear	An introductory book on optimal control and differential games (H_∞ control). Extensions include non-scalar (matrix-valued) cost functionals.

where ∇V_x^{μ} denotes the gradient of the value function V^{μ} with respect to the state x. Equation (2.10) is essentially the Lyapunov equation, which is found to be equivalent to the continuous-time Bellman equation [41]. Define the Hamiltonian of the problem following (2.10) as

$$H(x, \mu(x), \nabla V_x^{\mu}) := r(x, \mu(x)) + (\nabla V_x^{\mu})^{\mathsf{T}}(Ax + B\mu(x)) \tag{2.11}$$

Using Pontryagin's minimum principle, the optimal value function $V^*(x)$ satisfies the continuous-time HJB equation

$$0 = \min_{\mu} H(x, \mu(x), \nabla V_x^*) \tag{2.12}$$

and the optimal control satisfies

$$\mu^* = \arg\min_{\mu} H(x, \mu(x), \nabla V_x^*);\ \forall\ x \tag{2.13}$$

By letting $\partial H(\cdot)/\partial \mu = 0$, the optimal control for system (2.6) can be obtained as

$$\mu^* = -\frac{1}{2} R^{-1} B^{\mathsf{T}} \nabla V_x^* \tag{2.14}$$

Inserting the optimal control (2.14) into (2.12), the formulation of the HJB equation can be written as

$$0 = x^{\mathsf{T}} Q x + (\nabla V_x^*)^{\mathsf{T}} A x - \frac{1}{4} (\nabla V_x^*)^{\mathsf{T}} B R^{-1} B^{\mathsf{T}} \nabla V_x^*; \quad V^*(0) = 0 \tag{2.15}$$

For the linear system (2.6), by definition (2.7) the optimal value function is quadratic in the state [30] so that

$$V^*(x) := x^{\mathsf{T}} P x;\ \forall\ x \tag{2.16}$$

with a matrix $P = P^{\mathsf{T}} > 0$ such that $V^*(x)$ is a positive definite and radially unbounded function. Substituting $\nabla V_x^* = 2Px$ into (2.14), the optimal control becomes

$$\mu^* = -R^{-1} B^{\mathsf{T}} P x = K x;\ \forall\ x \tag{2.17}$$

Inserting (2.17) into the Lyapunov equation (2.10) using $\partial V^*/\partial t = \dot{x}^\top P x + x^\top P \dot{x}$ gives the algebraic Riccati equation:

$$A^\top P + PA - PBR^{-1}B^\top P + Q = 0 \tag{2.18}$$

which is essentially the HJB equation, where a stabilising closed-loop controller (2.17) can be determined by solving for the matrix P. In summary, the above optimal control problem can be formulated as: given the continuous-time linear time-invariant system (2.6) associated with the infinite horizon integral cost (2.7) for a set of admissible control policies $u = \mu(x)$, find a $\mu(x)$ such that V^μ (2.7) is minimised. This is essentially solving the HJB equation (2.15) or, in this linear case, solving the algebraic Riccati equation (2.18).

The LQR above combined with Kalman filter (linear-quadratic state estimator) [42] is called linear-quadratic-Gaussian (LQG) control, which is one of the most fundamental optimal control problems. Another important variation of optimal control that has been widely used in industrial applications is the model predictive control (MPC) [43, 44]. MPC optimises in a receding time horizon (repeatedly optimising the current time slot while considering the future time slot), which differs from the LQR that optimises in a fixed time horizon (a single optimal solution for the whole time horizon).

Various applications of optimal control in automotive systems are given in [45, 46]. Relevant references on optimal control for engine applications can be found in [47–53] from a seminar attended at the IAAPS, University of Bath during PhD, which was delivered by Dr Benjamin Pla from Universitat Politècnica de València (UPV).

2.1.3 Robust Control

Robust control [54–57] deals explicitly with uncertainty in controller design, where the controller can effectively address the uncertain parameters or disturbances within some compact set. The early classical methods of Bode and others were satisfactorily

robust. The modern state-space methods, however, were sometimes found to lack robustness. Prime examples of modern robust control techniques include H_∞ loop shaping developed by McFarlane and Glover [54, 58] and sliding model control (SMC) [59, 60]. In contrast with an adaptive control policy, a robust control policy is static, i.e. the controller, instead of adapting to measurements of variations, is designed to work assuming that certain disturbances (H_∞ control) or variables (SMC) will be unknown but bounded.

The basic idea of robust control can be explained in a similar manner as for adaptive control with respect to the control error e. For error dynamics given as (2.1), we also know a fixed nominal value or estimate $\hat{f}(x)$ for unknown $f(x)$ such that the estimation error $\tilde{f}(x) = f(x) - \hat{f}(x)$ is bounded, i.e. $||\tilde{f}(x)|| \leq F(x)$ with $F(x)$ being a known upper bound (possibly nonlinear) function. Unlike the adaptive control that needs the linear structure $f(x) = W^\top \phi(x)$, robust control tends to assume less information (the bound of $f(x)$). A robust nonlinear sliding mode controller can be selected as

$$u = \hat{f}(x) + K_p e - \gamma \tag{2.19}$$

where γ is a robust control term given by

$$\gamma = \begin{cases} -e\frac{F(x)}{||e||}, ||e|| \geq \epsilon \\ \\ -e\frac{F(x)}{\epsilon}, ||e|| < \epsilon \end{cases} \tag{2.20}$$

where $\epsilon > 0$ is a small design parameter. The closed-loop system becomes

$$\dot{e} = f(x) - u = f(x) - (\hat{f}(x) + K_p e - \gamma) = \tilde{F}(x) - K_p e + \gamma \tag{2.21}$$

This robust controller is sometimes easier to implement than the adaptive controller because it does not include any additional dynamics, the closed-loop system is bounded stable with $||e||$ bounded with a magnitude near ϵ. The error does not go to zero but does stay small.

2.1.4 Intelligent Control

Apart from the three building blocks of modern control techniques, i.e. adaptive control; optimal control; robust control, recent years have witnessed significant advancement in intelligent control [61, 62] thanks to the increased computing power. Intelligent control is a collection of control techniques that use various artificial intelligence computing approaches such as neural networks [63], fuzzy logic [64], Bayesian probability, machine learning, evolutionary computation, etc. As the distinction begins to become meaningless in terms of mathematics, artificial intelligence and control theory tend to be more compatible with each other in this new era. We are particularly interested in marrying reinforcement learning with adaptive optimal control, which will be discussed in Section 3.1.1.

2.1.5 Robust Adaptive Control

The success of adaptive control in the 1970s was soon followed by controversies over practicality, where the control schemes were criticised as being easily unstable in the presence of a small disturbance [65]. This motivated many researchers to understand the mechanisms of instabilities and find ways to counteract them. By the mid 1980s, several new redesigns and modifications were proposed, which led to a more general framework known as robust adaptive control [27, 66]. For example, the dead-zone modification [67] was proposed to stop the adaptation process when the norm of the tracking error becomes smaller than the prescribed value. The σ-modification adds damping to the adaptive law, which does not require any prior information on the system disturbance upper bounds [68]. A drawback when applying the σ-modification is that adaptive parameters tend to return to the origin for small tracking errors, even for the persistent excitation condition. The e-modification [69] was developed to overcome this undesirable effect by replacing the constant damping gain with a term proportional to a linear combination of tracking errors. The projection operator [70] was introduced to enable the adaptive laws to achieve robustness with respect to

both parametric and nonparametric uncertainties that might exist in system dynamics, which can tolerate fast adaptation.

Different procedures for combining "adaptive" and "robust" techniques can be found in the work by Yao *et al.* [71–73] and by Herrmann *et al.* [74–76].

2.1.6 Convex Optimisation

Mathematical optimisation is universally useful in solving quantitative problems in almost all the disciplines. Optimisation problems are often referred to as programmes in the context of operations research. In control theory, subjects such as optimal control, MPC, and extremum seeking control can be viewed as the generalisations of mathematical optimisation. Here, we feel the need to include the discussion on convex optimisation in this section as it plays a prominent role in the development of modern control theory.

An important class of optimisation problems is convex optimisation [77], which studies the case when the objective function is convex (minimisation) or concave (maximisation) and the constraint set is convex. It includes well-known least-squares and linear programming problems and can be viewed as a particular case of nonlinear programming or as a generalisation of linear or convex quadratic programming. Convex optimisation is genuinely useful for automatic control. There are great advantages in recognising or formulating a control problem as a convex optimisation problem. A control problem can then be solved, very reliably and efficiently, using interior-point methods or other numerical methods. For example, our theoretical development on adaptive control will largely extend the conventional methods used from least-squares, gradient descent, to more sophisticated finite-time parameter estimation techniques.

Another important benefit is that some control problems can be reduced to standard convex or quasiconvex optimisation problems involving the linear matrix inequality

(LMI) [78], where Lyapunov or Riccati equations can be solved efficiently.

2.2 Reinforcement Learning

Artificial intelligence has advanced significantly in recent decades, owing in great part to breakthroughs in machine learning [79], particularly advances in reinforcement learning [1]. Although part of these advancements are due to the increased computer power available, new innovations in theory and algorithms have also been driving forces. Reinforcement learning is a computational paradigm that helps us to analyse, abstract, and automate goal-directed learning and decision making [80]. It differs from other computational approaches in that it focuses on learning from the agent's direct interactions with its environment, and it is regarded as the third machine learning paradigm after supervised learning [79] and unsupervised learning [81]. Reinforcement learning is the closest kind of machine learning to the kind of learning that humans and other animals perform, and many of the key algorithms of reinforcement learning were inspired by biological learning systems. It concerns the agent that takes control actions in an environment to optimise a cumulative reward. In contrast to supervised learning that replicates the decisions of human experts, reinforcement learning systems are trained from their own experience which may allow them to exceed human capabilities [82], e.g., AlphaGo (trained by reinforcement learning from self-play) achieved a 99.8% winning rate against other Go programs and became the first computer program that defeated the human European Go champion by 5 games to 0 [83].

We are interested in reinforcement learning because it substantively interacts with many engineering and scientific disciplines toward greater integration with optimisation, control theory, operations research, and other mathematical subjects. The reinforcement learning is also formulated as a set of approximate dynamic programming (ADP) or adaptive dynamic programming methods by Werbos [84]. ADP was developed originally for feedback control of discrete-time systems, but the idea was soon

naturally extended to continuous-time systems, which allows the design of adaptive controllers using a "critic-actor" structure that learns the optimal control solution in real time. ADP or reinforcement learning essentially bridges the gap in philosophy between adaptive control and optimal control [85].

2.2.1 Markov Decision Process

The problem of reinforcement learning is often formalised as the optimal control of incompletely-known Markov decision processes [1]. Markov decision processes [34] are a mathematically idealised framework of reinforcement learning, which define the interaction between a learning agent and its environment with respect to states, actions, and rewards, as shown in Fig. 2.1.

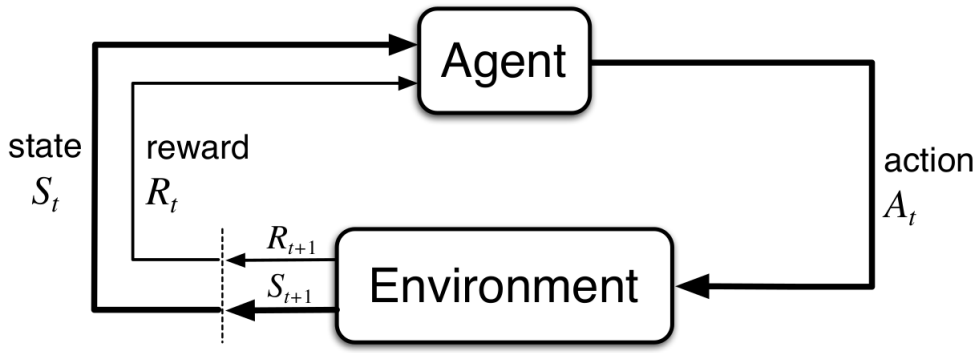

Figure 2.1: The agent-environment interaction in a Markov decision process [1].

To be specific, the learning agent and environment interact at each of a sequence of discrete time steps, $t = 0,\ 2,\ 3, ...$ such that

- the actions A_t are taken by the agent;
- the states S_t are for taking the actions;
- the rewards R_t are for evaluating the actions;
- the internal dynamics is completely known by the agent;

- the environment may not be completely known by the agent.

The objective of the agent is to maximise the amount of reward it receives over time. In a finite Markov decision process, the sets of states, actions, and rewards all have a finite number of elements, and the random variables R_t and S_t have well defined probability distributions dependent only on the preceding state and action, i.e.

$$p(s', r \mid s, a) = Pr\{S_t = s',\ R_t = r \mid S_{t-1} = s,\ A_{t-1} = a\} \tag{2.22}$$

where the probability p completely characterises the dynamics of the Markov decision process. That is, the probability of each possible value for S_t and R_t depends only on the immediately preceding state and action, S_{t-1} and R_{t-1}, and not at all on earlier states and actions. Thus, the state is said to have the Markov property that must include information about all aspects of the past agent-environment interaction that make a difference for the future [86].

Almost all reinforcement learning algorithms involve estimating value functions (function of states or of state-action pairs) that estimate how good it is for the agent to be in a given state or to perform a given action in a given state. Accordingly, value functions are defined in terms of specific sets of acting, i.e. policies. A policy $\pi(a, s)$ is a mapping from states to probabilities of selecting each possible action. A (discounted) return G_t is a function of future rewards that the agent looks to maximise (in expected value), i.e.

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{2.23}$$

where γ is the discount rate such that $0 \leq \gamma \leq 1$.

The value function of a state s under a policy π, denoted $v_\pi(s)$, is the expected return when starting in s and following π thereafter, which is defined as

$$v_\pi(s) = E_\pi\{G_t \mid S_t = s\} \tag{2.24}$$

where $E_\pi\{\cdot\}$ denotes the expected value of a random variable provided that the agent follows the policy π. It is a crucial characteristic of reinforcement learning and dynamic programming that value functions satisfy recursive relationships at successive time steps as follows:

$$\begin{aligned} v_\pi &= E_\pi\{G_t \mid S_t = s\} \\ &= E_\pi\{R_{t+1} + \gamma G_{t+1} \mid S_t = s\} \\ &= \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a)[r + \gamma v_\pi(s')] \end{aligned} \tag{2.25}$$

This is the Bellman equation for v_π and it expresses the relationship between the value of a state and the values of its successor states. The optimal value function is defined as

$$v_*(s) = \max_\pi v_\pi(s) \tag{2.26}$$

The Bellman equation for v_*, i.e. the Bellman optimality equation, expresses the fact that the value of a state under an optimal policy must equal the expected return for the best action from that state, which can be written as

$$v_*(s) = \max_a \sum_{s',r} p(s', r \mid s, a)[r + \gamma v_*(s')] \tag{2.27}$$

In principle, an optimal policy can be determined by solving the Bellman optimality equation for the optimal value functions.

Similarly, value functions can be treated with respect to state-action pairs. One can define the value of taking action a in state s under a policy π, denoted $q_\pi(s,a)$, as the expected return starting from s, taking action a, and following policy π, i.e.

$$q_\pi(s,a) = E_\pi\{G_t \mid S_t = s,\ A_t = a\} \tag{2.28}$$

Accordingly, the optimal action-dependent value function can be written as

$$q_*(s,a) = \max_\pi q_\pi(s,a) \tag{2.29}$$

The Bellman optimality equation for q_* becomes

$$q_*(s,a) = \sum_{s',r} p(s',r \mid s,a)[r + \gamma \max_{a'} q_*(s',a')] \tag{2.30}$$

This action-dependent version of value functions is particularly useful to develop model-free algorithms, which will be exploited later in Monte Carlo methods, Q-learning, etc.

The following three sections go through three different types of approaches for solving finite Markov decision problems: dynamic programming, Monte Carlo methods, and temporal-difference learning. Dynamic programming approaches have been theoretically well developed, but they demand a thorough and accurate representation of the environment. Monte Carlo techniques are theoretically simple and do not require a model, but they are not well suited for step-by-step incremental computing. Temporal-difference approaches, on the other hand, do not require a model and are fully incremental, although they are more difficult to analyse.

2.2.2 Dynamic Programming

Dynamic programming was developed by Bellman [87] in the 1950s and has found a wide range of applications from engineering to economics. Remarkable literature of dynamic programming can be found such as Bertsekas [34, 88], Ross [89], and Puterman [90]. It covers a collection of algorithms that can be used to compute optimal policies given a perfect model of the environment as a Markov decision process. Classical dynamic programming is of limited utility in reinforcement learning due to its strict assumption of a perfect model and its great computational expense. However, the algorithms theoretically provide an essential foundation for the understanding of most reinforcement learning methods, which attempt to achieve the same effect but with less computation and without assuming a perfect model.

The first connection between dynamic programming and reinforcement learning was made by Minsky [91] where Samuel's checkers players show the possibility of apply-

ing dynamic programming to solve Samuel's backing-up process analytically. The advancement was then made by Werbos [84, 92–95] who proposed an approximating approach named as heuristic dynamic programming that emphasises gradient-descent methods. Watkins in his PhD thesis [86] explicitly connected the two by characterising a class of reinforcement learning methods as incremental dynamic programming.

Algorithm 1 Policy iteration algorithm for estimating $\pi \approx \pi_*$ [1]

Initialization
$V(s)$ and $\pi(s)$ arbitrarily for all nonterminal s
Policy Evaluation
Loop:
    $\Delta \leftarrow 0$
    Loop for each s:
        $v \leftarrow V(s)$
        $V(s) \leftarrow \sum_{s',r} p(s', r \mid s, \pi(s))[r + \gamma V(s')]$
        $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$ (a small positive number determining the accuracy of estimation)
Policy Improvement
policy-stable$\leftarrow$*true*
For each s:
    old-action$\leftarrow \pi(s)$
    $\pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s', r \mid s, a)[r + \gamma V(s')]$
    If *old-action*$\neq \pi(s)$, then *policy-stable*$\leftarrow$*false*
If *policy-stable*, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else go to Policy Evaluation

The key idea of both dynamic programming and reinforcement learning is the use of value functions to organise and structure the search for good policies. The two most popular dynamic programming methods, policy iteration and value iteration, can be used to reliably compute optimal policies and value functions for finite Markov decision processes given complete knowledge of the model.

2.2.2.1 Policy Iteration

Each policy iteration is guaranteed to represent a significant improvement over the one before it (unless it is already optimal). Because there are only a finite number of policies in a finite Markov decision process, it must converge to an optimal policy and optimal value function in a finite number of iterations (see Algorithm 1).

One disadvantage of policy iteration is that each iteration requires policy review, which can be a lengthy repetitive process involving numerous sweeps across the state set. One might shorten the policy evaluation step without sacrificing policy iteration's convergence guarantees, for example, by halting policy evaluation after only one sweep (one state update), which is known as value iteration.

2.2.2.2 Value Iteration

Value iteration can be simply obtained by turning the Bellman optimality equation (2.27) into an update rule. It successfully combines one sweep of policy assessment and one sweep of policy reform in each of its sweeps (see Algorithm 2). Interposing numerous policy assessment sweeps between each policy improvement sweep typically results in faster convergence.

Algorithm 2 Value iteration algorithm for estimating $\pi \approx \pi_*$ [1]

Initialisation
$V(s)$ arbitrarily for all nonterminal s except that $V(terminal) = 0$
Value Update
Loop:
    $\Delta \leftarrow 0$
    Loop for each s:
        $v \leftarrow V(s)$
        $V(s) \leftarrow \max_a \sum_{s',r} p(s', r \mid s, \pi(s))[r + \gamma V(s')]$
        $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$ (a small positive number determining the accuracy of estimation)
Output a deterministic policy, $\pi \approx \pi_*$, such that
    $\pi(s) = \arg\max_a \sum_{s',r} p(s', r \mid s, a)[r + \gamma V(s')]$

2.2.2.3 Generalised Policy Iteration

Generalised policy iteration (GPI) is the basic concept of allowing policy-evaluation and policy-improvement processes to interact, regardless of the granularity and other features of the two processes. GPI can be used to define almost all reinforcement learning approaches. As demonstrated in Fig. 2.2, all have distinct policies and value functions, with the policy continually improving in relation to the value function and the value function always being driven toward the policy's value function.

Only when the value function is compatible with the present policy, and only when the policy is *greedy* with regard to the current value function, does it stabilise. As a result, both processes stabilise only when a policy that is greedy in terms of its own evaluation function is discovered. This means that the Bellman optimality equation (2.27) holds true, implying that the policy and value function are both optimal.

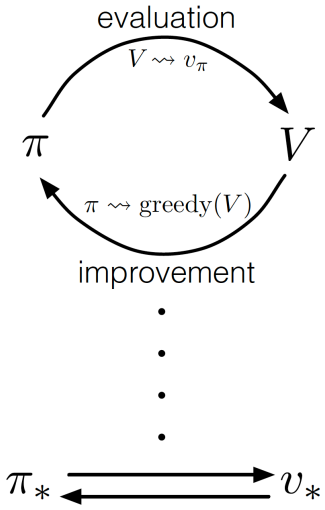

Figure 2.2: The interaction between policy evaluation and improvement processes in GPI [1].

"Bootstrapping" is a characteristic of dynamic programming approaches in which estimations of the value of states are updated based on estimates of the values of succeeding states. Many reinforcement learning approaches perform bootstrapping, even those that do not require, as dynamic programming does, a complete and accurate model of the environment.

2.2.3 Monte Carlo Methods

Monte Carlo methods are a broad class of algorithms that rely on repeated random sampling, of which the principle can be found in textbooks, e.g. [96–99]. The essential idea is to use randomness to solve problems that might be deterministic in general. In reinforcement learning, the Monte Carlo methods learn value functions and optimal

policies from experience in the form of sample episodes, which means the complete knowledge of the environment is not needed.

Given a model, it is sufficient to estimate the state-dependent value function to determine a policy. One simply looks ahead one step and chooses whichever action leads to the best combination of reward and next state, as in dynamic programming methods. Without a model, however, state-dependent value function alone is not sufficient. One must explicitly estimate the value of each action in order for the values to be useful in suggesting a policy, i.e. to estimate q_*.

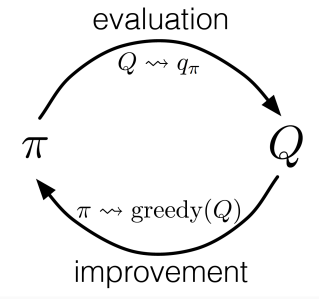

Figure 2.3: The interaction between policy evaluation and improvement processes in GPI (action-dependent version) [1].

The Monte Carlo approaches (see Algorithm 3), which follow the GPI framework, also include interactive processes of policy review and policy improvement. The distinction is that the techniques simply average several returns that begin in the state, rather than utilising a model to determine the value of each state. This average can become a decent approximation to the value because a state's value is the expected return. We're particularly interested in approximating action-dependent value functions in control techniques since they may be utilised to enhance policies without requiring an environment model. The GPI maintains an approximation policy as well as an approximate value function in this case. As demonstrated in Fig. 2.3, the value function is regularly changed to more closely match the value function for the present policy, and the policy is continually improved in relation to the current value function.

Algorithm 3 Monte Carlo algorithm for estimating $\pi \approx \pi_*$ [1]

Initialisation
$\pi(s)$, $Q(s)$ arbitrarily for all nonterminal s
$Returns(s,a) \leftarrow$empty list, for all s
Value and Policy Update
Loop forever (for each episode):
    Choose S_0, A_0 randomly such that all pairs have probability > 0
    Generate an episode from S_0, A_0, following π : $S_0, A_0, R_1, \cdots, S_{T-1}, A_{T-1}, R_T$
    $G \leftarrow 0$
    Loop for each step of episode, $t = T-1, T-2, \cdots, 0$:
        $G \leftarrow \Gamma G + R_{t+1}$
        Unless the pair S_t, A_t appears in $S_0, A_0, S_1, A_1, \cdots, S_{T-1}, A_{T-1}$:
            Append G to $Returns(S_t, A_t)$
            $Q(S_t, A_t) \leftarrow$ average($Returns(S_t, A_t)$)
            $\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$

Compared to dynamic programming, Monte Carlo methods can learn the optimal behaviour directly from the interaction with the environment and they require only experience (sample sequences of states, actions, and rewards) without a model. Another advantage is that they are less likely to be impacted by breaches of the Markov property since they do not bootstrap their value estimates using the value estimates of future states.

2.2.4 Temporal Difference Learning

Temporal difference learning is one of the central and novel ideas in reinforcement learning and can be regarded as a combination of dynamic programming and Monte Carlo ideas. It can learn directly from raw experience without a model as with Monte Carlo methods and it bootstraps as in dynamic programming. The idea has its early roots in animal learning psychology and artificial intelligence, most notably the work of Samuel [100] and Klopf [101] and then developed later, e.g., [86, 102–107].

One of the radical challenges in reinforcement learning is to balance exploitation and exploration. The *greedy* method is an example of exploitation which exploits the current knowledge of the values of actions. The exploration occurs when one selects

one of the nongreedy actions, which may produce the greater total reward in the long run. ε-greedy methods first used in Watkins [86] are often used to trade off exploitation and exploration by adjusting the small probability ε that all the actions are randomly selected with equal probability. The temporal difference methods can be classified into on-policy or off-policy approaches to address the sufficient exploration problems. On-policy approaches aim to improve or assess the policy that is used to make choices, whereas off-policy methods aim to enhance or evaluate a policy that is not used to create the data. Sarsa is an on-policy approach, whereas Q-learning is an off-policy method, as explained below.

2.2.4.1 Sarsa

As an on-policy method, Sarsa [105, 106] estimates $q_\pi(s,a)$ for the current behaviour policy π and for all states s and actions a. Every element of the quintuple of events, $(S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1})$, that make up a transition from one state-action pair to the next is used by its update rule (see Algorithm 4).

Algorithm 4 Sarsa (on-policy) algorithm for estimating $Q \approx q_*$ [1]

Initialisation
$Q(s,a)$ arbitrarily for all s,a except that $Q(terminal,\cdot) = 0$
Value and Policy Update
Loop for each episode:
    Choose A from S using policy derived from Q (e.g., ε-greedy)
    Loop for each step of episode:
        Take action A, observer R, S'
        Choose A' from S' using policy derived from Q (e.g., ε-greedy)
        $Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma Q(S',A') - Q(S,A)]$ with step size $\alpha \in [0,1]$
        $S \leftarrow S'$; $A \leftarrow A'$
until S is terminal

2.2.4.2 Q-Learning

As an off-policy method, Q-learning [86, 108] directly approximates the optimal action-dependent value function q_*, independent of the policy being followed (see Algorithm 5).

Algorithm 5 Q-learning (off-policy) algorithm for estimating $\pi \approx \pi_*$ [1]

Initialisation
$Q(s,a)$ arbitrarily for all s,a except that $Q(terminal,\cdot) = 0$; Step size $\alpha \in [0,1]$
Value and Policy Update
Loop for each episode:
    Initialise S
    Loop for each step of episode:
        Choose A from S using policy derived from Q (e.g., ε-greedy)
        Take action A, observer R, S'
        $Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma \max_a Q(S',a) - Q(S,A)]$
        $S \leftarrow S'$
until S is terminal

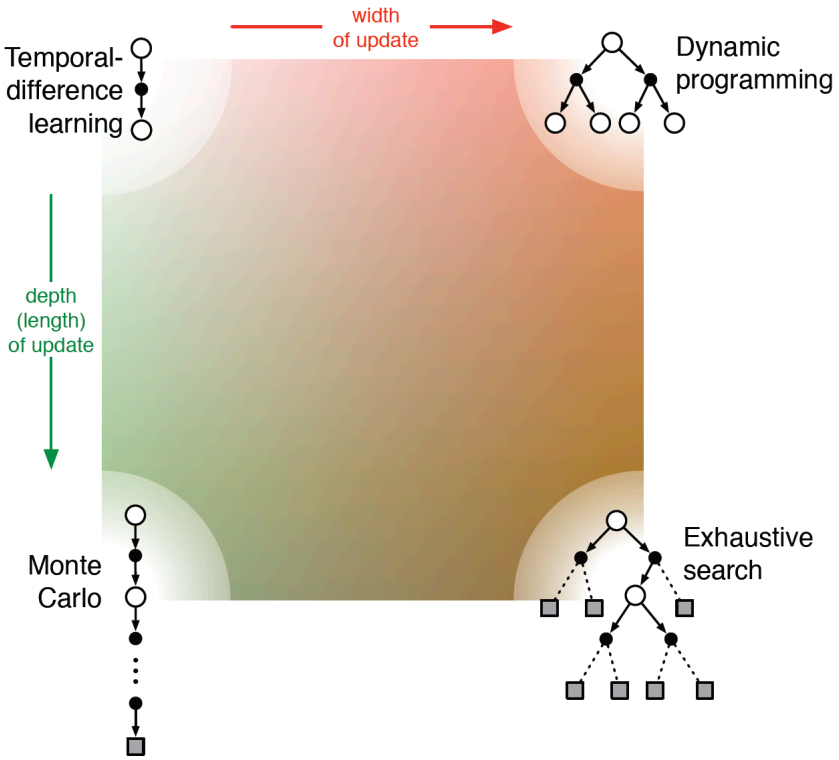

Figure 2.4: Reinforcement learning methods: the depth and width of the updates [1].

Almost all reinforcement learning techniques share three fundamental concepts: 1) they all try to estimate value functions; 2) they all work by backing up values along actual or potential state trajectories; and 3) they all use the general strategy of generalised policy iteration (GPI), which means they keep an approximate value function and an approximate policy and try to improve one on the other. The three concepts of value functions, storing up value updates, and GPI are strong organising principles that might be applied to any artificial or natural intelligence model [1].

The aforesaid reinforcement learning techniques are connected and compared in Fig. 2.4 with regard to two dimensions [1]. The horizontal dimension determines whether the updates are sample (based on a sample trajectory) or expected (based on a distribution of possible trajectories). Expected updates need a distribution model, whereas sample updates can be done with just a sample model or without any model at all (another dimension of variation). The depth of updates, or the degree of bootstrapping, is represented by the vertical dimension of Fig. 2.4. The three basic approaches for estimating values are dynamic programming, temporal difference, and Monte Carlo, which are located at three of the four corners of the space. The sample-update techniques, which range from one-step temporal-difference updates to full-return Monte Carlo updates, are located at the left edge of the space. A range of approaches based on n-step updates exists between these two extremes [1].

2.3 Automotive Engine Control

We look into the opportunities to develop and implement novel control systems for automotive powertrain as the practical application, where the control problems for internal combustion engines are particularly of our interest to the Ph.D. research.

2.3.1 Automotive Control Overview

Automotive control has become a driving factor in automotive innovation over the past decades with the advancement of electronics. The control problems for modern automotive systems can be grouped into two sides, i.e. powertrain control and vehicle control, with respect to powering and driving [109–112]. The powertrain acts as the core of a vehicle and comprises the main components that power the vehicle, which typically includes the engine, clutch, transmission, drive shafts, and wheels or other final drives such as propellers, continuous tracks, etc. The powertrain control problems have become more complex and challenging since the emergence of electric vehicles along with hybrid electric vehicles and fuel cell vehicles. In our research, we focus on powertrain control and particularly on internal combustion engines control. Although the invention of the combustion engine dates back to the 18th century, engine technology continues to advance and it is predicted that by 2050, at least 60 percent of light-duty vehicles will still use combustion engines, but often work with electric motors in hybrid systems and largely equipped with a turbocharger [113]. Many surveys provide comprehensive summaries of the control problems for powertrain systems [114–116] and engines [117, 118]. The control problems for the vehicle itself are extensive yet with the same main purpose of helping the driver to perform the task of keeping the vehicle on the road in a safe manner [109]. Vehicle control systems are often safety- or comfort-oriented, e.g. cruise and headway control, traction control, anti-lock brake system (ABS), vehicle stability control, and active suspensions [109, 110]. Moreover, it is necessary to mention that autonomous driving has been springing up and will also remain a dynamic and exciting control topic [119, 120].

2.3.2 Engine Dynamics Modelling

Engine dynamics modelling has been playing an important role in the engine development and optimisation process. The objective with engine dynamics modelling is twofold: 1) to *predict* the engine performance without practical tests; 2) to *deduce* the

performance of parameters that can be difficult to measure in tests . It is clearly advantageous if engine performance can be anticipated without having to go through the difficulty of first creating an engine, then instrumenting, testing, and finally analysing the data. However, most of the events that occur in an internal combustion engine are too complicated to be modelled from first principles only [121]. Engine modelling often relies heavily on experimental data and empirical correlations, e.g. lookup tables along with proper interpolation methods. In this thesis, we focus on the thermodynamics modelling for engine control design. The mechanical modelling of the engine (e.g. a finite element model for the engine structure) is not in the scope of our research. Yet we also include crankshaft dynamics considering its important interactions with other engine subsystems.

There have been many different methodologies for engine modelling [122]. Some are completely based on measurements of relevant engine outputs at varying ranges of control inputs, i.e. black-box models. On the other hand, some are based primarily on physics principles with a few relevant parameters that are experimentally calibrated, i.e. white-box models. If a suitable engine model is provided, accurate and fast engine simulations can allow for rapid incorporation of new control design. The control development process can be significantly shortened when using efficient computational tools. A high-fidelity model may reduce the hardware prototypes and development cost. Commercial software such as GT POWER and AVL BOOST offers platforms for modelling the engine dynamics as an one-dimensional computational fluid dynamics (CFD) model. The CFD models usually have high fidelity and require a large amount of computational time. In the case of real-time control development, a physics-based mean value engine model (MVEM) developed by Hendricks [123–126] is widely used with low-fidelity but fast running speed. Instead of cycle by cycle analysis, the MVEM presents the average response of multiple ignition cycles in the time domain, which contains nonlinear differential equations mixed with empirical static maps and often has a cumbersome structure.

One way to overcome the complexity and to provide a fast solution for control design is linear state space modelling. The state space model plays a central role in modern control theory [9] and has been commonly used as a framework for robust control, optimal control, etc. It can effectively deal with a multi-input, multi-output (MIMO) system such as engines. A linear state space model for engines was proposed in [127] for the application of linear quadratic control. The model was then generalised to control-oriented engine models in [128]. In the last decade, artificial neural networks have been seen as an attractive approach for dynamic system modelling and control. There are many studies on the application of neural networks on engine modelling, e.g. [129–134] therein. Neural networks can be regarded as a black-box system identification approach that is conceptually simple, and easy to use, and has excellent approximation properties.

2.3.3 Engine Management System

Control has always been a component of engine design, and it is one of the most difficult problems to solve [135]. For modern engines, engine control is generally achieved by the engine management system (EMS). The main objective of the EMS is to regulate the engine torque as required by the demand (e.g., the driver), and, at the same time, to meet the stringent requirements for emission, fuel consumption, power output, and safety [110]. The power output from an internal combustion engine is determined by the available torque (the clutch torque) and the engine speed. The clutch torque is produced from the indicated torque generated from the combustion process, reduced by the friction loss, and pumping loss, as well as the torque necessary to operate the auxiliary systems. The indicated torque by combustion is determined by three variables: 1) the air mass for combustion; 2) the fuel mass for combustion; and 3) the spark ignition timing [112]. The EMS usually consists of different sensors that measure the real-time engine performance and actuators that control the fuel injector, spark plug, throttle, etc [135].

The fundamental torque control module is coordinated with a number of basic modules within the EMS:

- Air-fuel ratio (AFR) control
- Idle speed control
- Ignition timing and knock control
- Electronic throttle control (ETC)

Some engines also have variable valve timing control, and exhaust gas recirculation (EGR) control.

Emission reduction and fuel economy improvement are two active areas in engine research. Rather than taking a holistic approach to engine control system development, the existing research endeavours focus on developing the specific control module. Among the engine control problems, AFR control is the most crucial and demanding topic and will be addressed primarily in this thesis. Fig. 2.5 shows the schematic of a typical port fuel injection spark ignition engine with EGR.

2.3.4 Air-Fuel Ratio Control

The common treatment for engine emissions is to convert pollutant exhaust CO, NO_x, into innocuous ones: N_2, H_2O, and CO_2, using three-way catalytic (TWC) converters. However, as shown in Fig. 2.6, the conversion efficiency of TWC is fairly sensitive to AFR, which is required to be regulated around the stoichiometric value (e.g. 14.7 for petrol) [136]. Moreover, combustion with a stoichiometric AFR is essential to achieving the optimal thermal efficiency and dynamic performance. Therefore, it is of great importance to design a well-performing AFR controller for engines so as to improve emissions, thermal efficiency and fuel economy. For most spark ignition engines in production, the widely-used control strategy is still PID control based on lookup tables, which could be difficult to meet the emission requirement in the presence of

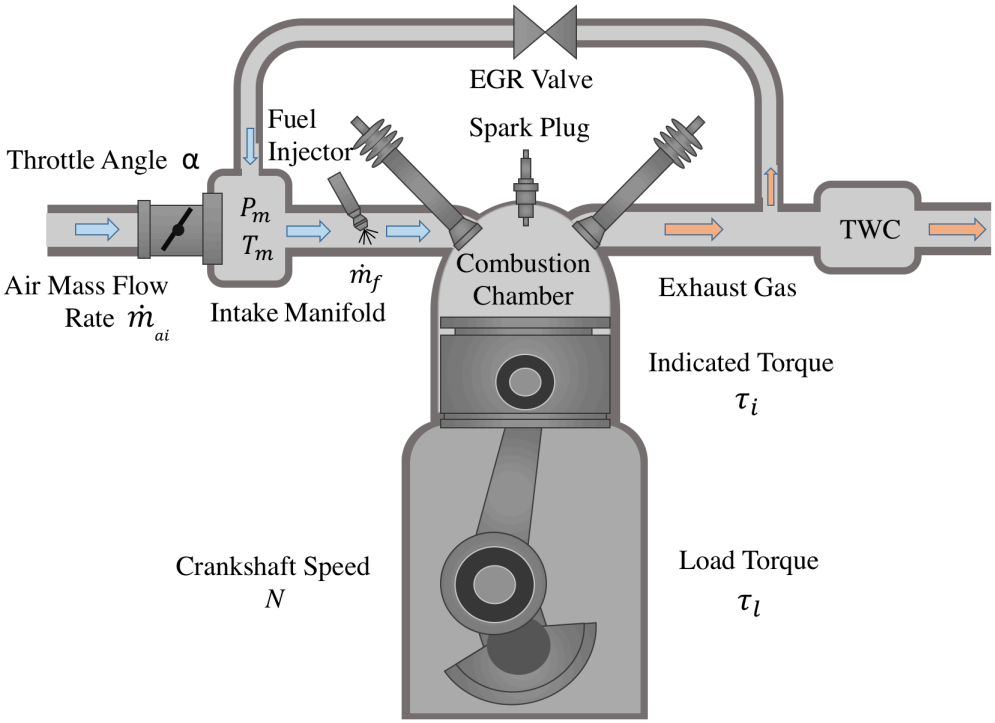

Figure 2.5: The schematic of a typical port fuel injection spark ignition engine with EGR.

complex dynamics and rapid-change operation scenarios. Practically, the compilation of the lookup tables also requires significant effort in engine calibration tests and is usually time-consuming [137].

This motivates the research on advanced AFR control design such as optimal control [138], robust control [139][140], adaptive control [136][141], and, more recently, observer-based control [137][142]. An optimal AFR controller was designed in [138] considering the cyclic variations of residual gas. However, it requires the knowledge of in-cylinder pressure, for which the sensor could often be expensive and not applicable for commercial engines. Then, robust techniques such as H_∞ control [139] and sliding mode control [140] were proposed to regulate the AFR in the presence of external disturbance. In order to deal with parameter uncertainties, adaptive approaches were

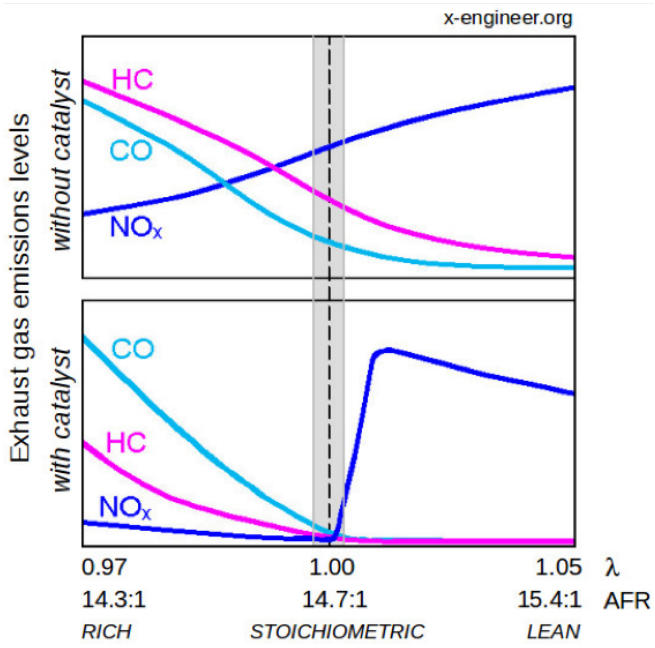

Figure 2.6: Emission levels with respect to AFR.

presented to address air-filling dynamics in [136] and time delay dynamics in [141]. However, the complexity of the adaptive controller limits their practical implementation. This prompts further work on AFR control using simple, easily implemented observers. In [142], a sliding mode AFR controller was proposed using observers to reduce chattering. Later on, various popular observer techniques were investigated in [137], which show great potential in application with design simplicity. However, the effect of fuel puddle dynamics was not specifically studied in [137]. There the two parameters: fuel puddle fraction and the time constant for the puddle evaporation, are assumed to be known for AFR control, which are, however, not measurable in practice. In this thesis, we will use novel nonlinear observer techniques to estimate the unknown dynamics and account for disturbances. We will also investigate the implementation of adaptive optimal control for the AFR regulation, where, to our best knowledge, Q-learning-based control is applied to engine systems for the first time.

2.4 Conclusions

The literature review addressed the fundamental topics in control theory: adaptive control, optimal control, robust control, intelligent control, and robust adaptive control and in reinforcement learning: MDP, dynamic programming, Monte Carlo Methods, and temporal difference learning. We will show the in-depth knowledge of adaptive optimal control and its “identical twin" adaptive/approximate dynamic programming (ADP) in the next chapter, where the generalised policy iteration and Q-learning is used to develop new adaptive optimal control. We have also presented a literature review of automotive engine control: dynamics modelling, engine management system, and air-fuel ratio control. This will be useful to understand our work on control-oriented modelling, idle speed control, and air-fuel ratio control in the later chapters.

CHAPTER 3

A New Approach to Adaptive Optimal Control*

In this chapter, we propose two novel adaptive optimal control algorithms for continuous-time nonlinear input-affine systems based on reinforcement learning: i) generalised policy iteration (GPI) and ii) Q-learning. As a result, the *a priori* knowledge of the system drift $f(x)$ is not needed via GPI, which gives us a partially model-free and online solution. We then for the first time extend the idea of Q-learning to the *nonlinear* continuous-time optimal control problem in a noniterative manner. This leads to a completely model-free method where neither the system drift $f(x)$ nor the input gain $g(x)$ is needed. For both methods, the adaptive critic and actor are continuously and simultaneously updating each other without iterative steps, which effectively avoids the hybrid structure and the need for an initial stabilising control policy. Moreover, finite-time convergence is guaranteed by using a sliding mode technique in the new adaptive approach, where the persistent excitation (PE) condition can be directly verified online. We also prove the overall Lyapunov stabil-

*The content of this chapter is adapted from the author's own work [143], where some materials have been re-used.

ity and demonstrate the effectiveness of the proposed algorithms using numerical examples.

3.1 Introduction

In the context of control theory, the idea of combining adaptive control [29] and optimal control [30] has emerged recently due to the advancement in reinforcement learning [1][41][144], which is also known as approximate/adaptive dynamic programming (ADP) [84]. A common framework for studying reinforcement learning is the Markov decision process (MDP), where the control process is often stochastic and formulated in discrete time. That follows the increasing need to formalise the method in a control perspective for deterministic continuous-time systems. Fig. 3.1 shows a closed-loop control system based on reinforcement learning, which is equivalent to the agent-environment interaction shown in Fig. 2.1 in a control perspective.

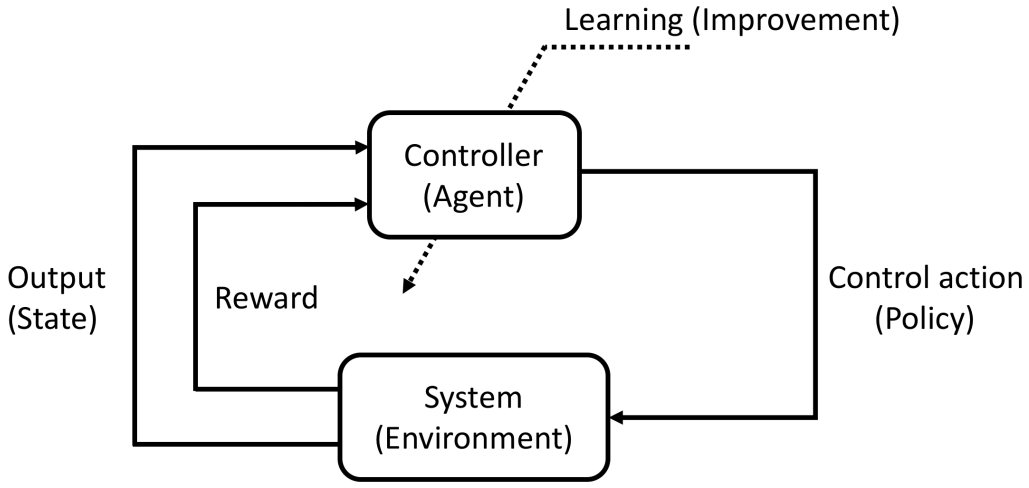

Figure 3.1: Control system based on reinforcement learning.

3.1.1 Related Work

Many surveys are available [144][145][146] addressing the development of reinforcement learning algorithms in both discrete and continuous time. The challenge now becomes directing these learning techniques towards practically feasible continuous-time adaptive optimal controllers. At the early stage of MDP studies, one class of ADP methods known as policy iteration (PI) [1] was developed, i.e. a two-step iteration: policy evaluation and policy improvement. This was equivalent to Newton's method proposed by Kleinman [147] by iteratively solving Lyapunov equations. The proposed PI was known to be computationally intensive and could be only implemented offline. Vrabie later developed an online PI algorithm for continuous-time linear systems [148] and then for nonlinear control-affine systems [149]. The method employed two neural networks in a critic/actor configuration and, more importantly, allowed the online adaptation of the controller to the optimal state feedback control without knowing the system drift dynamics. This forms the idea of integral reinforcement learning (IRL) [150] which generates a large family of algorithms. However, the controller was not in a standard form but was based on a hybrid structure with a continuous-time controller and a discrete-time sampling data learning structure.

To overcome the sequential updates of the critic and actor, Vamvoudakis [151] proposed an online synchronous PI algorithm by using an adaptive control approach, where the two neural networks were simultaneously tuned online. The PE condition was required to ensure the convergence of the adaptive algorithm. Unlike the IRL in [148][149], the synchronous PI approximated the solution of the HJB equation and required the complete knowledge of system dynamics. In order to deal with systems that are only partially known, Na *et al.* [152] suggested an identifier-critic structure for ADP that identifies the unknown nonlinear part of the system and, by using new adaptive techniques [75], the actor neural network is avoided while the PE condition can be easily verified in real time. Nevertheless, all the above ADP methods require complete or at least partial *a priori* knowledge of the system dynamics.

To lift the requirement of system knowledge, the idea of Q-learning is of particular interest since it is a model-free reinforcement learning algorithm. This leads to a different approach to address the ADP for unknown systems. Q-learning has been primarily developed for stochastic discrete-time systems, i.e., MDP. The very first Q-learning algorithm was initiated in a finite MDP context by Watkins in his PhD thesis [86] and has attracted much attention from the machine learning communities. Al-Tamimi implemented Q-learning [153] for the first time for a discrete-time linear H_∞ control problem of an F-16 aircraft autopilot in simulation. This work inspired many advancements in Q-learning implementation for discrete-time systems, e.g. linear quadratic-tracking (LQT) [154]) and output feedback control [155]. However, there is still a lack of research regarding Q-learning studies in a deterministic continuous-time case. In fact, Q-learning was first posed in continuous time as “advantage updating" [156]. It was specified in [157] that the Q-function can be seen as an extension of the Hamiltonian, which connects Q-learning with continuous-time control. A proper definition of a Q-function in continuous time is still disputed. It should be noted that the “Q-function" in [157][158] has a different meaning from that in reinforcement learning [159]. Researchers started to mimic the Q-learning from discrete-time systems to continuous-time systems, which resulted in stepwise iterative algorithms in [158] [160][161]. The consequence is again a hybrid structure like Vrabie’s IRL where the control signal is generated in discrete time while the system is in continuous time. An integral Q-learning algorithm [158] was derived from the singular perturbation of the control input; it solved the continuous-time linear-quadratic regulation (LQR) problem but required a stabilising (admissible) initial policy. The above limitations were later overcome by Vamvoudakis [162] via combining the idea of IRL [149] and synchronous PI [151] into Q-learning, where the algorithm employed two neural networks in a critic/actor configuration and was restricted to the LQR case. Indeed, all the solutions above [158]-[162] were limited to linear systems. In this chapter, we will show one of our theoretical results [143] which for the first time extends Q-learning to deterministic continuous-time nonlinear systems.

3.1.2 Contributions

This chapter proposes two new adaptive optimal control algorithms for continuous-time nonlinear input-affine systems. The main contributions are summarised as follows: i) To the best of our knowledge, for the first time, the idea of Q-learning is extended to the *nonlinear* continuous-time optimal control problem as an adaptive optimal controller in a noniterative manner, where an initial stabilising policy as in [149][158][160][163] is not required. ii) The two proposed methods: GPI and Q-learning, are partially and completely *model-free*, i.e., neither the *a priori* knowledge of system dynamics in [151] nor the additional identifier in [152] is needed. iii) The adaptive critic and actor are continuously and simultaneously updating each other without iterative steps, which effectively avoids the hybrid structure in [149] with a continuous-time actor and a discrete-time sampling-based critic. iv) The finite-time convergence is guaranteed by using a sliding mode technique [75] in the new adaptive approach, where the PE condition can be directly verified online. Moreover, the actor neural network in [151] is not necessary to prove the overall stability.

3.2 Preliminaries

This section presents a general formulation of the infinite-horizon nonlinear optimal control problem for continuous-time systems. Given the continuous-time nonlinear input-affine time-invariant system

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t), \quad x(0) = x_0 \tag{3.1}$$

where $x(t) \in \mathbb{R}^n$ is the measurable state vector, $u(t) \in \mathbb{R}^m$ is the control policy or input vector, and $f(x(t)) \in \mathbb{R}^n$, $g(x(t)) \in \mathbb{R}^{n\times m}$ are the system drift and the input gain functions, respectively. We define the value function $V^u(x) \in C^1$ as the infinite-horizon integral cost

$$V^u(x(t)) := \int_t^\infty r(x(\tau), u(\tau))\mathrm{d}\tau \tag{3.2}$$

where $r = S(x(t)) + u^\mathsf{T}(t)Ru(t)$ is the utility (also known as reward in reinforcement learning) with positive definite $S(x(t)) \in \mathbb{R}$ and $R = R^\mathsf{T} \in \mathbb{R}^{m\times m}$. For simplicity, we set R to be diagonal in this chapter without loss of generality.

Assumption 3.1 *It is assumed that $f(x) + g(x)u$ is Lipschitz continuous on a compact set $\Omega \in \mathbb{R}^n$ that contains the origin and the system* (3.1) *is stabilisable, i.e. the system state x is bounded for a stabilising control u.* ◇

The optimal control problem is to minimise the value function (3.2) by choosing the optimal stabilising control (or admissible policy) $u^*(t)$. The optimal value function $V^*(x)$ can be defined as

$$V^*(x(t)) := \min_u \int_t^\infty r(x(\tau), u(\tau))\mathrm{d}\tau \tag{3.3}$$

A general solution to the nonlinear optimal control problem can be formulated as a partial differential equation for the optimal value function $V^*(x)$. We define the Hamiltonian of the problem as

$$\mathcal{H}(x, u, \nabla V_x^u) := r(x, u) + (\nabla V_x^u)^\mathsf{T}(f(x) + g(x)u) \tag{3.4}$$

with the gradient vector $\nabla V_x^u = \partial V^u/\partial x \in \mathbb{R}^n$. The optimal value function $V^*(x)$ in (3.3) satisfies the *Hamilton-Jacobi-Bellman* (HJB) equation

$$0 = \min_u \mathcal{H}(x, u, \nabla V_x^*) \tag{3.5}$$

For unconstrained control u, the optimal control u^* can be found by setting $\partial\mathcal{H}(x, u, \nabla V_x^*)/\partial u = 0$ so that

$$u^* = -\frac{1}{2}R^{-1}g(x)^\mathsf{T}\nabla V_x^* \tag{3.6}$$

Inserting the optimal control (3.6) into (3.5) gives the HJB equation in terms of ∇V_x^* as

$$0 = S(x) + (\nabla V_x^*)^\mathsf{T} f(x) - \frac{1}{4}(\nabla V_x^*)^\mathsf{T} g(x)R^{-1}g(x)^\mathsf{T}\nabla V_x^* \tag{3.7}$$

The HJB equation (3.4) is generally difficult to solve due to its nonlinearity and the requisite for explicitly knowing the system drift dynamics $f(x)$ and input gain dynamics $g(x)$.

3.3 Generalised Policy Iteration

Policy iteration [1] is one of the reinforcement learning methods for finding the optimal value and optimal control. It iteratively performs *policy evaluation* and *policy improvement* until the optimal policy is reached. The method generates a family of algorithms (e.g. [149][151]) to solve the HJB equation online and forward in time. In this section, these two processes are concurrent since the critic and the actor are continuously and simultaneously updating each other. This method can be interpreted as an extreme version of the generalised policy iteration (GPI) [1]. It should be noted that the proposed method, although called GPI, does not use iteration but solves the HJB equation in a continuous manner.

For continuous-time systems, *policy evaluation* can be achieved by an adaptive critic based on a nonlinear Lyapunov equation (e.g. [151][152]), which can be derived by differentiating value function (3.2) via Leibniz's formula. Another approach is via the integral reinforcement learning (IRL) [150] Bellman equation

$$V^u(x(t-T)) = \int_{t-T}^{t} r(x(\tau), u(\tau))\mathrm{d}\tau + V^u(x(t)) \tag{3.8}$$

with a sample period $T > 0$. This is an analogue to the discrete-time Bellman equation in the integral form. Note that the system drift $f(x)$ and input gain $g(x)$ appearing in the Lyapunov equation are not involved here in the Bellman equation (3.8). For *policy improvement*, it is shown in [164] by successively solving (3.8) for the value function V^u, that the following control

$$u = -\frac{1}{2}R^{-1}g(x)^\top \nabla V_x^u \tag{3.9}$$

will uniformly converge to the optimal control u^* (3.6).

3.3.1 Adaptive Critic for Value Function Approximation

This section presents a new design of the adaptive critic for *policy evaluation*. We approximate the value function using a critic neural network such that

$$V^u(x) = w^\top \varphi(x) + \varepsilon(x) \tag{3.10}$$

where $\varphi(x):\mathbb{R}^n\to\mathbb{R}^N$ denotes the activation function vector with the number N of neurons in the hidden layer, $w\in\mathbb{R}^N$ is the weight vector and $\varepsilon(x)\in\mathbb{R}$ is the neural network approximation error. The activation functions are selected to provide a complete independent basis set so that $V(x)$ is uniformly approximated. According to the Weierstrass higher-order approximation theorem [164], within a compact set Ω, the error $\varepsilon(x)$ and its derivative $\nabla\varepsilon_x$ are bounded for a fixed N and $\varepsilon(x)\to 0$, $\nabla\varepsilon_x\to 0$ as the number of neurons $N\to\infty$.

We use the Bellman approach to update the critic. Inserting the value function approximation (3.10) into the Bellman equation (3.8) gives

$$\underbrace{\int_{t-T}^{t} r(x(\tau),u(\tau))\mathrm{d}\tau}_{\rho(x,u)}+\underbrace{w^{\mathsf{T}}\varphi(x(t))-w^{\mathsf{T}}\varphi(x(t-T))}_{w^{\mathsf{T}}\Delta\varphi(t)}=-\varepsilon_B \tag{3.11}$$

with the integral reinforcement $\rho(x,u)$, the difference $\Delta\varphi(t)=\varphi(x(t))-\varphi(x(t-T))$, and the Bellman equation residual error $\varepsilon_B=\varepsilon(x(t))-\varepsilon(x(t-T))$ being bounded for bounded $\varepsilon(x)$ within the compact set Ω (In practical applications, one can almost always find a compact set that is sufficiently large to analyse the problem). In order to construct an adaptive law that can estimate the weight of the value function approximation with guaranteed convergence, we introduce a set of auxiliary variables $P_1\in\mathbb{R}^{N\times N}$ and $Q_1\in\mathbb{R}^N$ by low-pass filtering the variables in (3.11) as

$$\begin{cases}\dot{P}_1=-\ell P_1+\Delta\varphi(t)\Delta\varphi(t)^{\mathsf{T}},\ \ P_1(0)=0\\ \dot{Q}_1=-\ell Q_1+\Delta\varphi(t)\rho(x,u),\ \ Q_1(0)=0\end{cases} \tag{3.12}$$

with a filter parameter $\ell>0$. The forgetting factor ℓ providing an exponential leakage effectively avoids the unbounded explosion of $P_1(t)$, $Q_1(t)$ and guarantees stability [75]. Their solutions can be found by solving (3.12) as

$$\begin{cases}P_1(t)=\int_0^t e^{-\ell(t-\tau)}\Delta\varphi(\tau)\Delta\varphi^{\mathsf{T}}(\tau)\mathrm{d}\tau\\ Q_1(t)=\int_0^t e^{-\ell(t-\tau)}\Delta\varphi(\tau)\rho(\tau)\mathrm{d}\tau\end{cases} \tag{3.13}$$

Definition 3.1 (Persistent Excitation (PE) [19]) The signal $\Delta\varphi(t)$ is said to be persistently excited over the time interval $[t-T,\ t]$ if there exists a strictly positive constant

$\sigma_1 > 0$ such that

$$\int_{t-T}^{t} \Delta\varphi(\tau)\Delta\varphi(\tau)^\top d\tau \geq \sigma_1 I, \quad \forall t > 0 \tag{3.14}$$

The PE condition [19] is widely required in adaptive control to guarantee parameter convergence. The adaptation will stop when either the weights converge or when there is no excitation, whichever happens first. P_1 and Q_1 lose the information over time due to the exponential leakage from ℓ. If ℓ is selected to be very big, there would not be enough knowledge for the controller to learn the optimal solution, i.e. the weights converge before reaching the true value. Therefore, persistent excitation condition is needed to ensure good weight estimation.

Lemma 3.1 [75] *If the signal $\Delta\varphi(t)$ is persistently excited for all $t > 0$, the auxiliary variable P_1 defined in* (3.12) *is positive definite, i.e. $P_1 \succ 0$ and the minimum eigenvalue $\lambda_{min}(P_1) > \sigma_1 > 0$, $\forall t > 0$ for some positive constant σ_1.* ⋄

Proof. The detailed proof follows from [75]. □

The adaptive critic neural network can be written as

$$\hat{V}(x) = \hat{w}^\top \varphi(x) \tag{3.15}$$

where $\hat{w}$ and $\hat{V}(x)$ denote the current estimates of w and $V^u(x)$, respectively.

Now we design the adaptation law using a sliding mode technique to update $\hat{w}$ such that

$$\dot{\hat{w}} = -\Gamma_1 P_1 \frac{M_1}{||M_1||} \tag{3.16}$$

where $M_1 \in \mathbb{R}^N$ is defined as $M_1 = P_1\hat{w} + Q_1$ and $\Gamma_1 \succ 0$ is a diagonal adaptive learning gain to be tuned. The rate of convergence of the parameter estimation is proportional to the adaptive gain Γ_1. Increasing the value of Γ_1 will speed up the convergence. A large Γ_1, however, may make the differential equation (the adaptive law) stiff and, therefore, more difficult to solve numerically [27].

Lemma 3.2 *Given the adaptation law* (3.16)*, if the system state* $x(t)$ *is bounded for a stabilising control and* $u(t)$*,* $\Delta\varphi(t)$ *and the system states* $x(t)$ *are persistently excited, one can formulate for the estimation error of weight* $\tilde{w} = w - \hat{w}$ *that*

a) If there is no neural network approximation error, i.e. $\varepsilon(x) = 0$*, the error* $\tilde{w}$ *will converge to zero in finite time* $t_1 > 0$*.*

b) If $\varepsilon(x) \neq 0$*, the error* $\tilde{w}$ *will converge to a compact set in finite time* $t_1 > 0$*.* ◇

Proof. We first examine the boundedness in terms of M_1. From (3.13), with states $x(t)$, $x(t-T)$ being bounded, the matrix P_1 is upper bounded for some positive $\delta_{P_1} > 0$ such that $\lambda_{\max}(P_1) \leq \delta_{P_1}$. Inserting ρ in (3.11) into (3.13) gives $Q_1 = -P_1 w + \Lambda_1$ with $\Lambda_1(t) = \int_0^t e^{-\ell(t-\tau)} \Delta\varphi(\tau)\varepsilon_B(\tau)\mathrm{d}\tau$ being bounded by some constant $\delta_1 > 0$ as the Bellman equation residual error ε_B is bounded. Then, M_1 can be written as

$$M_1 = -P_1\tilde{w} + \Lambda_1 \tag{3.17}$$

Since $\Delta\varphi(t)$ is persistently excited, from *Lemma 3.1* we know P_1 is symmetric positive definite so it is invertible. Then, we have $P_1^{-1}M_1 = -\tilde{w} + P_1^{-1}\Lambda_1$. Here $P_1^{-1}M_1$ can be used to design a proper Lyapunov function as it contains the estimation error $\tilde{w}$ and Λ_1. We differentiate $P_1^{-1}M_1$ as

$$\frac{\partial}{\partial t}(P_1^{-1}M_1) = -\dot{\tilde{w}} + \frac{\partial P_1^{-1}}{\partial t}\Lambda_1 + P_1^{-1}\dot{\Lambda}_1 = \dot{\hat{w}} + \bar{\Lambda}_1 \tag{3.18}$$

with $\bar{\Lambda}_1 = -P_1^{-1}\dot{P}_1 P_1^{-1}\Lambda_1 + P_1^{-1}\dot{\Lambda}_1$ being bounded for bounded Λ_1, i.e., $||\bar{\Lambda}_1|| \leq \bar{\delta}_1$ holds for a constant $\bar{\delta}_1 > 0$. Note that P_1^{-1} is bounded since $\lambda_{\min}(P_1) > \sigma_1$ and $\lambda_{\max}(P_1) < \delta_{P_1}$, so the lower and upper bounds of P_1^{-1} can be found as $\lambda_{\min}(P_1^{-1}) > \delta_{P_1}$ and $\lambda_{\max}(P_1^{-1}) < 1/\sigma_1$. Thus, one can easily find two class $\mathcal{K}$ functions [165] of M_1 that serve as the lower and upper bounds of the following time-varying Lyapunov function

$$\mathcal{L}_1 = \frac{L_1}{2}(P_1^{-1}M_1)^\top \Gamma_1^{-1} P_1^{-1} M_1 \tag{3.19}$$

with a positive constant $L_1 > 0$. Its time derivative can be determined as

$$\begin{aligned}\dot{\mathcal{L}}_1 &= L_1 M_1^\top P_1^{-1} \Gamma_1^{-1} (\dot{\hat{w}} + \dot{\bar{\Lambda}}_1) \\ &= L_1 M_1^\top P_1^{-1} \Gamma_1^{-1} (-\Gamma_1 P_1 \frac{M_1}{||M_1||} + \dot{\bar{\Lambda}}_1) \\ &\leq -\alpha_1 \sqrt{\mathcal{L}_1}\end{aligned} \tag{3.20}$$

where $\alpha_1 = (\sigma_1 - L_1 \bar{\delta}_1 \lambda_{\max}(\Gamma_1^{-1}))\sqrt{2/\lambda_{\max}(\Gamma_1^{-1})}$ is a positive constant for a properly chosen L_1 with $0 < L_1 < \sigma_1/(\lambda_{\max}(\Gamma_1^{-1})\bar{\delta}_1)$. According to [166], it can be found that $\mathcal{L}_1 = 0$ and $M_1 = 0$ so that

a) In the case of $\varepsilon(x) = 0$, we can obtain $\varepsilon_B = 0$, and $\Lambda_1 = \bar{\Lambda}_1 = 0$, which implies that $\tilde{w} = 0$ and $M_1 = 0$ so that $\tilde{w}$ will converge to zero in finite time $t_1 = 2\sqrt{\mathcal{L}_1(0)}/\alpha_1 > 0$ where $\alpha_1 = \sigma_1\sqrt{2/\lambda_{\max}(\Gamma_1^{-1})}$.

b) In the case of $\varepsilon(x) \neq 0$, i.e., $\varepsilon_B \neq 0$, $M_1 = 0$ in finite time implies that $\dot{\tilde{w}} = P_1^{-1}\dot{\Lambda}_1$, and $||\tilde{w}|| \leq \delta_1/\sigma_1$ after finite time t_1. □

Remark 3.1 *From Lemma 3.1, the PE condition can be verified online by checking the minimum eigenvalue of P_1. For implementation, the PE condition can be retained by reinitiating the state or adding sufficient exploration noise to the control as in [151][163].* ◇

Remark 3.2 *The adaptation law* (3.16) *with the sliding mode term $M_1/||M_1||$ can lead to finite-time convergence of the weight $\hat{w}$ without causing a severe chattering phenomenon [75] due to the integration action.* ◇

3.3.2 Adaptive Optimal Control via GPI

Now we design an actor for *policy improvement*. By inspection of (3.9), one can determine the optimal control directly using the adaptive critic (3.15) if the weight $\hat{w}$ converges to the actual unknown weight w which solves the Bellman equation (3.8). The control law (actor) will be

$$u = -\frac{1}{2} R^{-1} g(x)^\top \nabla \Phi^\top \hat{w} \tag{3.21}$$

Now we summarise the first result of this chapter as follows:

Theorem 3.1 *Given the continuous-time nonlinear affine system* (3.1) *with the infinite-horizon value function* (3.2)*, the adaptive critic neural network* (3.15) *with the adaptation law* (3.16) *and the actor* (3.21) *form an adaptive optimal control so that:*

a) In the absence of a neural network approximation error, the adaptive critic weight estimation error $\tilde{w}$ will converge to zero and the actor u will converge to its optimal control solution u^* in finite time $t_1 > 0$.

b) In the presence of a neural network approximation error, the adaptive critic weight estimation error $\tilde{w}$ will converge to a compact set and the actor u will converge to a small bounded set around its optimal control solution u^* in finite time $t_1 > 0$.

Proof. We design the Lyapunov function following a similar procedure as in [150][152]

$$\mathcal{L}_2 = \mathcal{L}_1 + L_2 V^* + \frac{L_3}{2} \Lambda_1^\top \Lambda_1 \tag{3.22}$$

with positive constants L_2 and L_3. We investigate the Lyapunov function $\mathcal{L}_2$ in a compact set $\tilde{\Omega} \in \mathbb{R}^N \times \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^N$ in tuple (M_1, x, u, Λ_1) that contains the origin and $\tilde{\Omega} \subset \Omega$. Ω in *Assumption 3.1* and $\tilde{\Omega}$ are chosen to be sufficiently large but of fixed size. Any initial value of (M_1, x, u, Λ_1) is assumed to be within the interior $\tilde{\Omega}$. Thus, for any initial trajectory, the state x and the control u remain bounded for at least finite time $t \in [0, T_1]$. Within (3.22), differentiating the term $L_2 V^*(x)$ will involve $\dot{V}^* = (\nabla V_x^*)^\top \dot{x}$. Note that the HJB equation (3.5) can be written as

$$0 = r(x, u) + (\nabla V_x^*)^\top (f(x) + g(x)u) \tag{3.23}$$

Considering a Young's inequality $ab \leq \frac{\eta_1}{2} a^2 + \frac{1}{2\eta_1} b^2$ with constant $\eta_1 > 0$, using (3.19)

(3.22)(3.23), the derivative of $\mathcal{L}_2$ can be derived as

$$\begin{aligned}\dot{\mathcal{L}}_2 &= \dot{\mathcal{L}}_1 + L_2(\nabla V_x^*)^\top(f+gu) + L_3\Lambda_1^\top\dot{\Lambda}_1 \\ &= L_1 M_1^\top P_1^{-1}\Gamma_1^{-1}(\dot{\hat{w}} + \dot{\bar{\Lambda}}_1) + L_2(-r(x,u)) \\ &\quad + L_3\Lambda_1^\top(-\ell\Lambda_1 + \Delta\varphi\varepsilon_B) \\ &\leq -\alpha_1'||M_1|| - \alpha_2 S(x) - \alpha_3||u||^2 - \alpha_4||\Lambda_1||^2 + \beta_1\end{aligned} \tag{3.24}$$

where $\alpha_1' = 1 - L_1\bar{\delta}_1\lambda_{\max}(\Gamma_1^{-1})/\sigma_1$, $\alpha_2 = L_2$, $\alpha_3 = L_2\lambda_{\min}(R)$, $\alpha_4 = L_3\ell - L_3\eta_1/2$ are positive constants for properly chosen L_1, L_2, L_3, η_1 with $0 < L_1 < \sigma_1/(\lambda_{\max}(\Gamma_1^{-1})\bar{\delta}_1)$, $L_2 > 0$, $L_3 > 0$, $0 < \eta_1 < 2\ell$, respectively; $\beta_1 = L_3||\Delta\varphi\varepsilon_B||/(2\eta_1)$ addresses the effect of the neural network approximation error. Thus, the first four terms in the last inequality of (3.24) form a negative definite function in $\tilde{\Omega}$ so that the set of ultimate boundedness Ω_u exists and it depends on the size of β_1, i.e. a smaller size of β_1 will decrease the size of Ω_u. Assuming that N has been chosen large enough, this implies β_1 to be sufficiently small so that $\Omega_u \subset \tilde{\Omega}$. Hence, it is impossible for any trajectory to leave $\tilde{\Omega}$, i.e. it is an invariant set, i.e. the states $x(t)$ remain bounded and subsequently also the functions of $x(t)$: approximation error $\varepsilon(x)$, $\varphi(x)$ are bounded functions over a compact set. This also implies that

a) In the case of no neural network approximation error, $\varepsilon = 0$ and $\varepsilon_B = 0$. Then we have $\beta_1 = 0$ and

$$\dot{\mathcal{L}}_2 \leq -\alpha_1'||M_1|| - \alpha_2 S(x) - \alpha_3||u||^2 - \alpha_4||\Lambda_1||^2 \leq 0 \tag{3.25}$$

According to Lyapunov's theorem and *Lemma 3.2*, $\mathcal{L}_2$ and $\tilde{w}$ will converge to zero, and based on (3.9)(3.10)(3.21), the difference of the actor to the optimal control

$$||u^* - u|| \leq \frac{1}{2}||R^{-1}g(x)^\top\nabla\varphi||\,||\tilde{w}|| \tag{3.26}$$

will also converge to zero in finite time t_1, i.e. the actor $\hat{u}$ will converge to its optimal solution u^*.

b) In the case of $\varepsilon \neq 0$, then $\varepsilon_B \neq 0$, $\beta_1 \neq 0$. From Lyapunov's theorem and *Lemma 3.2*, $\mathcal{L}_2$ and $\tilde{w}$ are uniformly ultimately bounded. The difference of the actor to the optimal

control

$$||u^* - u|| \leq \frac{1}{2}||R^{-1}g(x)^\top \nabla\varphi||||\tilde{w}|| + \frac{1}{2}||R^{-1}g(x)^\top||||\nabla\varepsilon|| \tag{3.27}$$

bounded after finite time t_1. It depends on the weight error $\tilde{w}$ and the approximation error $\nabla\varepsilon$. It follows the actor u will converge to a small bounded set around its optimal solution u^*. □

Remark 3.3 *The proposed GPI (Theorem 3.1) is a partially model-free algorithm that can approximately solve the continuous-time nonlinear optimal control problem online without the a priori knowledge of system drift $f(x)$. Hence, the identifier of the dual approximation structure in [152] can be further removed. Moreover, since the finite-time convergence of the critic weight is guaranteed, the actor neural network in [151] is not needed. The adaptive critic and the actor are continuously and simultaneously updating each other, which effectively avoids the hybrid structure as in [149] and does not require a stabilising initial control policy as in [149][163].* ◇

3.4 Nonlinear Q-Learning

It is widely shown that policy iteration [150][151][152], including our proposed GPI algorithm, still requires the *a priori* knowledge of the input gain $g(x)$. In this section, we extend the idea of Q-learning to continuous-time nonlinear systems in the form of adaptive optimal control, which leads to a completely model-free algorithm, i.e. neither the knowledge of $f(x)$ nor $g(x)$ is needed.

3.4.1 Parameterisation of Nonlinear Q-function

The core basis of Q-learning is to create an action-dependent value function $Q(x,u)$: $\mathbb{R}^{n+m} \to \mathbb{R}$ such that $Q^*(x,u^*) = V^*(x)$. For the continuous-time nonlinear input-affine system (3.1), the Q-function can be explicitly defined by adding the Hamiltonian (3.4)

onto the optimal value (3.3) as

$$\begin{aligned} Q(x,u) :&= V^*(x) + \mathcal{H}(x,u,\nabla V_x^*) \\ &= \underbrace{V^*(x) + S(x) + (\nabla V_x^*)^\top f(x)}_{F_{xx}(x)} + \\ &\underbrace{(\nabla V_x^*)^\top g(x) u}_{F_{xu}(x,u)} + \underbrace{u^\top R u}_{F_{uu}(u)} \end{aligned} \tag{3.28}$$

where $F_{xx}(x)$, $F_{xu}(x,u)$, and $F_{uu}(u)$ are the lumped terms that can be approximated respectively via neural networks.

Lemma 3.3 *The Q-function defined in* (3.28) *is positive definite with the optimisation scheme* $Q^*(x,u^*) = \min_u Q(x,u)$. *The optimal Q-function* $Q^*(x,u^*)$ *has the same optimal value* $V^*(x)$ (3.3) *as for the value function* $V^u(x)$ (3.2)*, i.e.* $Q^*(x,u^*) = V^*(x)$ *when applying the optimal control* u^*. ◇

Proof. From its definition (3.28), Q-function is the sum of the optimal value $V^*(x)$ and the Hamiltonian $\mathcal{H}(x,u,\nabla V_x^*)$, where $V^*(x)$ is positive definite. The HJB equation (3.5) implies that the minimisation of the Hamiltonian with respect to u yields the optimal solution. Hence, $Q^*(x,u^*) = \min_u Q(x,u)$. Inserting the HJB equation (3.5) with the optimal control u^* gives $\mathcal{H}(x,u^*,\nabla V_x^*) = 0$. Then we have $Q^*(x,u^*) = V^*(x)$. □

3.4.2 Adaptive Critic for Q-function Approximation

For the nonlinear affine system (3.1) with the Q-function (3.28), we approximate the Q-function using a critic neural network by

$$Q(x,u) = W^\top \Phi(x,u) + \varepsilon_Q(x,u) \tag{3.29}$$

where $\Phi(x,u) : \mathbb{R}^{n+m} \to \mathbb{R}^{N'}$ denotes the activation function vector with the number N' of neurons in the hidden layer, $W \in \mathbb{R}^{N'}$ is the weight vector, $\varepsilon_Q(x,u)$ is the neural network approximation error and $W^\top \Phi(x,u)$ can be explicitly expressed according to the three components $F_{xx}(x)$, $F_{xu}(x,u)$, and $F_{uu}(u)$ in (3.28) as

$$W^\top \Phi(x,u) = \begin{bmatrix} W_{xx}^\top & W_{xu}^\top & W_{uu}^\top \end{bmatrix} \begin{bmatrix} \Phi_{xx}(x) \\ vec(\Phi_{xu}(x) \otimes u) \\ \Phi_{uu}(u) \end{bmatrix}$$

$$= \begin{bmatrix} W_{xx} \\ --- \\ W_{xu1} \\ W_{xu2} \\ \vdots \\ W_{xum} \\ --- \\ W_{uu1} \\ W_{uu2} \\ \vdots \\ W_{uum} \end{bmatrix}^\top \begin{bmatrix} \Phi_{xx}(x) \\ ---- \\ \Phi_{xu}(x)u_1 \\ \Phi_{xu}(x)u_2 \\ \vdots \\ \Phi_{xu}(x)u_m \\ ---- \\ u_1^2 \\ u_2^2 \\ \vdots \\ u_m^2 \end{bmatrix} \tag{3.30}$$

where $\otimes$ denotes the Kronecker product and $vec(\cdot)$ is the vectorisation function which stacks the columns of a matrix together. For $\Phi_{xx} \in \mathbb{R}^{N_{xx}}$, $\Phi_{xu} \in \mathbb{R}^{N_{xu}}$ and $\Phi_{uu} \in \mathbb{R}^m$, the regressor $\Phi(x,u)$ is selected to provide a complete independent basis such that $Q(x,u)$ is uniformly bounded with $N' = N_{xx} + m(N_{xu}+1)$. Recall from the Weierstrass higher-order approximation theorem [164], the approximation error $\varepsilon_Q(x,u)$ is bounded for a fixed N' within a compact set Ω and as the number of neurons $N_{xx} \to \infty$ and $N_{xu} \to \infty$, i.e., $N' \to \infty$, we have $\varepsilon_Q(x,u) \to 0$.

Remark 3.4 *By the definition of the Q-function* (3.28)*, the terms* $W_{xx}^\top \Phi_{xx}(x)$, $W_{xu}^\top vec(\Phi_{xu}(x) \otimes u)$, $W_{uu}^\top \Phi_{uu}$ *in* (3.30) *account for the lumped functions* $F_{xx}(x)$, $F_{xu}(x,u)$, $F_{uu}(u)$ *in* (3.28)*, where* $F_{xu}(x,u)$ *is a linear function of* u *and* $F_{uu}(u)$ *is a quadratic function of* u*. Note that the regressors used here are radial activation functions that are different to the ridge activation functions in conventional neural networks. A polynomial regressor can easily represent each lumped function due to its simplicity.* $\diamond$

One needs to derive the Bellman equation in terms of the Q-function to update the

critic. By Bellman's principle of optimality [41], we have the following optimality equation

$$V^*(x(t-T)) = \int_{t-T}^{t} r(x(\tau), u(\tau))\mathrm{d}\tau + V^*(x(t)) \tag{3.31}$$

The result from *Lemma 3.3* showed that $Q^*(x,u^*) = V^*(x)$, which means we can rewrite (3.31) in terms of $Q^*(x,u^*)$ as

$$\begin{aligned} &\overbrace{-\int_{t-T}^{t} r(x,u^*)\mathrm{d}\tau}^{-\rho(x,u^*)} = Q^*(x(t),u^*(t)) - Q^*(x(t-T),u^*(t-T)) \\ &= \underbrace{W^\top \Phi(x(t),u^*(t)) - W^\top \Phi(x(t-T),u^*(t-T))}_{W^\top \Delta\Phi(x,u^*)} + \varepsilon_{BQ}(x,u^*) \end{aligned} \tag{3.32}$$

with the integral reinforcement $\rho(x,u)$, the difference $\Delta\Phi(t) = \Phi(x(t),u^*(t)) - \Phi(x(t-T),u^*(t-T))$, and the Bellman equation residual error $\varepsilon_{BQ} = \varepsilon_Q(x(t),u^*(t)) - \varepsilon_Q(x(t-T),u^*(t-T))$ being bounded for bounded $\varepsilon_Q(x,u)$. Define two auxiliary variables $P_2 \in \mathbb{R}^{N'\times N'}$ and $Q_2 \in \mathbb{R}^{N'}$ by low-pass filtering the variables in (3.32) as

$$\begin{cases} \dot{P}_2 = -\ell P_2 + \Delta\Phi(t)\Delta\Phi(t)^\top, \;\; P_2(0) = 0 \\ \dot{Q}_2 = -\ell Q_2 + \Delta\Phi(t)\rho(x,u), \;\; Q_2(0) = 0 \end{cases} \tag{3.33}$$

with a filter parameter $\ell > 0$.

The adaptive critic neural network can be written as

$$\hat{Q}(x,u) = \hat{W}^\top \Phi(x,u) \tag{3.34}$$

where $\hat{W}$ and $\hat{Q}(x,u)$ denote the current estimates of W and $Q(x,u)$, respectively.

Now we design the adaptation law using the sliding mode technique to update $\hat{W}$ such that

$$\dot{\hat{W}} = -\Gamma_2 P_2 \frac{M_2}{||M_2||} \tag{3.35}$$

where $M_2 \in \mathbb{R}^{N'}$ is defined as $M_2 = P_2\hat{W} + Q_2$ and $\Gamma_2 \succ 0$ is a diagonal adaptive learning gain to be tuned.

Lemma 3.4 *Given the adaptation law* (3.35)*, if the system state* $x(t)$ *is bounded for a stabilising control and* $u(t)$*,* $\Delta\Phi(t)$ *and the system states* $x(t)$ *are persistently excited, one can formulate for the estimation error of weight* $\tilde{W} = W - \hat{W}$ *that*

a) if there is no neural network approximation error, i.e. $\varepsilon_Q(x,u) = 0$*, the error* $\tilde{W}$ *will converge to zero in finite time* $t_2 > 0$*.*

b) if $\varepsilon_Q(x,u) \neq 0$*, the error* $\tilde{W}$ *will converge to a compact set in finite time* $t_2 > 0$*.* ◇

Proof. The proof follows similarly from *Lemma 3.2*. It can be obtained that $M_2 = -P_2\tilde{W} + \Lambda_2$ with Λ_2 defining the effect of approximation error ε_{BQ} and $\bar{\Lambda}_2 = -P_2^{-1}\dot{P}_2 P_2^{-1}\Lambda_1 + P_2^{-1}\dot{\Lambda}_2$ being bounded for bounded Λ_1, i.e., $||\Lambda_2|| \leq \delta_2$, $||\bar{\Lambda}_2|| \leq \bar{\delta}_2$ for constants $\delta_2 > 0$, $\bar{\delta}_2 > 0$. We can design a time-varying Lyapunov function

$$\mathcal{L}_3 = \frac{L_4}{2}(P_2^{-1}M_2)^\top \Gamma_2^{-1} P_2^{-1} M_2 \tag{3.36}$$

with a postive constant $L_4 > 0$ so that its time derivative $\dot{\mathcal{L}}_3 \leq -\alpha_5\sqrt{\mathcal{L}_3}$ holds for a positve constant α_5 if $0 < L_4 < \sigma_2/(\lambda_{\max}(\Gamma_2^{-1})\bar{\delta}_2)$. The convergence time t_2 is finite with $t_2 = 2\sqrt{\mathcal{L}_4(0)}/\alpha_2$. □

3.4.3 Adaptive Optimal Control via Q-learning

We reconstruct the optimal control u^* from (3.6) based on the parameterisation of $Q(x,u)$ (3.28) such that

$$u^* = -\frac{1}{2}R^{-1}W_{xu}^\top \Phi_{xu}(x) + \varepsilon_{Qu} \tag{3.37}$$

where ε_{Qu} is a bounded approximation error due to ε_Q, $W_{xu}^\top \Phi_{xu}(x)$ accounts for the term $g(x)^\top \nabla V_x^*$. One can determine the optimal control directly using the adaptive critic (3.34) if the weight $\hat{W}$ converges to the actual weight W. The control law (actor) will be

$$u = -\frac{1}{2}R^{-1}\hat{W}_{xu}^\top \Phi_{xu}(x) \tag{3.38}$$

We summarise the main result as

Theorem 3.2 *Given the continuous-time nonlinear affine system* (3.1) *with the infinite-horizon value function* (3.2) *and Q-function defined in* (3.28)*, the adaptive critic neural network* (3.34) *with the adaptation law* (3.35) *and the actor* (3.38) *form an adaptive optimal control so that:*

a) in the absense of a neural network approximation error, the adaptive critic weight estimation error $\tilde{W}$ *will converge to zero and the actor* u *will converge to its optimal control solution* u^* *in finite time* $t_2 > 0$*.*

b) in the presence of a neural network approximation error, the adaptive critic weight estimation error $\tilde{W}$ *will converge to a compact set and the actor* u *will converge to a small bounded set around its optimal control solution* u^* *in finite time* $t_2 > 0$*.*

Proof. We design the Lyapunov function following a similar procedure in [150] as

$$\mathcal{L}_4 = \mathcal{L}_3 + L_5 Q^*(x,u) + \frac{L_6}{2}\Lambda_2^\top \Lambda_2 \tag{3.39}$$

with positive constants L_5 and L_6. We investigate the Lyapunov function $\mathcal{L}_4$ in a compact set $\tilde{\Omega}^Q \in \mathbb{R}^{N'} \times \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{N'}$ in tuple (M_2, x, u, Λ_2) that contains the origin and $\tilde{\Omega}^Q \subset \Omega$. Ω in *Assumption 3.1* and $\tilde{\Omega}^Q$ are chosen to be sufficiently large but of fixed size. Any initial value of (M_2, x, u, Λ_2) is assumed to be within the interior $\tilde{\Omega}^Q$. Thus, for any initial trajectory, the state x and the control u remain bounded for at least finite time $t \in [0, T_2]$. From (3.28), differentiating the term $L_5 Q^*(x,u)$ in (3.39) will involve $\dot{Q}^*(x,u) = \dot{V}^* + \dot{\mathcal{H}}(x,u,\nabla V_x^*)$. Since the Lagrange multiplier $\lambda = \nabla V_x^*$, differentiating the Hamiltonian gives

$$\dot{\mathcal{H}}(x,u,\nabla V_x^*) = \partial\mathcal{H}/\partial t + (\nabla\mathcal{H}_u)^\top \dot{u} + (\nabla\mathcal{H}_x + \dot{\lambda})^\top \dot{x} \tag{3.40}$$

According to Lagrange's theory (pp. 114-115 [30]), from the costate equation and stationarity condition, the derivative of the Lagrange multiplier λ satisfies $\dot{\lambda} = -\nabla\mathcal{H}_x$ and $\nabla\mathcal{H}_u = 0$. For a time-invariant system (3.1) and value function (3.2), the Hamiltonian

$\mathcal{H}(x,u,\nabla V_x^*)$ is not an explicit function of t, i.e. $\dot{\mathcal{H}} = \partial\mathcal{H}/\partial t = 0$. Considering a Young's inequality $ab \le \frac{\eta_2}{2}a^2 + \frac{1}{2\eta_2}b^2$ with constant $\eta_2 > 0$, using (3.23)(3.28)(3.36)(3.39)(3.40), the derivative of $\mathcal{L}_4$ can be derived as

$$\begin{aligned}
\dot{\mathcal{L}}_4 &= \dot{\mathcal{L}}_3 + L_5[(\nabla V_x^*)^\top(f+gu) + \dot{\mathcal{H}}] + L_6\Lambda_2^\top\dot{\Lambda}_2 \\
&= L_4 M_2^\top P_2^{-1}\Gamma_2^{-1}(\dot{w} + \bar{\Lambda}_2) + L_5(-r(x,u)) \\
&\quad + L_6\Lambda_2^\top(-\ell\Lambda_2 + \Delta\Phi\varepsilon_{BQ}) \\
&\le -(1 - L_4\bar{\delta}_2\lambda_{\max}(\Gamma_2^{-1})/\sigma_2)||M_2|| - L_5 S(x) \\
&\quad - L_5\lambda_{\min}(R)||u||^2 - (L_6\ell - \frac{L_6\eta_2}{2})||\Lambda_2||^2 \\
&\quad + \frac{L_6}{2\eta_2}||\Delta\Phi\varepsilon_{BQ}|| \\
&\le -\alpha_5'||M_1|| - \alpha_6 S(x) - \alpha_7||u||^2 - \alpha_8||\Lambda_2||^2 + \beta_2
\end{aligned} \tag{3.41}$$

where $\alpha_5' = 1 - L_4\bar{\delta}_2\lambda_{\max}(\Gamma_2^{-1})/\sigma_2$, $\alpha_6 = L_5$, $\alpha_7 = L_5\lambda_{\min}(R)$, $\alpha_8 = L_6\ell - L_6\eta_2/2$ are positive constants for properly chosen L_4, L_5, L_6, η_2 with $0 < L_4 < \sigma_2/(\lambda_{\max}(\Gamma_2^{-1})\bar{\delta}_2)$, $L_5 > 0$, $L_6 > 0$, $0 < \eta_2 < 2\ell$, respectively; $\beta_2 = L_6||\Delta\Phi\varepsilon_{BQ}||/(2\eta_2)$ is the lumped error. Thus, the first four terms in the last inequality of (3.41) form a negative definite function in $\tilde{\Omega}^Q$ so that the set of ultimate boundedness Ω_u^Q exists and it depends on the size of β_2, i.e. a smaller value of β_2 will decrease the size of Ω_u^Q. Assuming that N' has been chosen large enough, it is possible to obtain β_2 to be sufficiently small so that $\Omega_u^Q \subset \tilde{\Omega}^Q$. Hence, it is impossible for any trajectory to leave $\tilde{\Omega}^Q$, i.e. it is an invariant set, i.e. the states $x(t)$ remain bounded and subsequently also the functions of $x(t)$ and $u(t)$: approximation error $\varepsilon_Q(x,u)$, $\Phi(x,u)$ are bounded functions over a compact set. This also implies that

a) In the case of no neural network approximation error, $\varepsilon_Q = 0$, $\varepsilon_{BQ} = 0$, and $\varepsilon_{Qu} = 0$. Then we have $\beta_2 = 0$ and

$$\dot{\mathcal{L}}_4 \le -\alpha_5'||M_2|| - \alpha_6 S(x) - \alpha_7||u||^2 - \alpha_8||\Lambda_2||^2 \le 0 \tag{3.42}$$

According to Lyapunov's theorem and *Lemma 3.4*, $\mathcal{L}_4$ and $\tilde{W}$ will converge to zero, and based on (6.13)(6.22)(6.23), the difference of the actor to the optimal control

$$||u^* - u|| \le \frac{1}{2}||\Phi_{xu}(x)||\ ||diag(\tilde{W}_{uu})^{-1}\tilde{W}_{xu}|| \tag{3.43}$$

will also converge to zero in finite time t_2, i.e,. the actor $\hat{u}$ will converge to its optimal solution u^*.

b) In the case of $\varepsilon_Q \neq 0$, then $\varepsilon_{BQ} \neq 0$, $\beta_2 \neq 0$. From Lyapunov's theorem and *Lemma 3.2*, $\mathcal{L}_4$ and $\tilde{W}$ are uniformly ultimately bounded. The difference of the actor to the optimal control

$$||u^* - u|| \leq \frac{1}{2}||\Phi_{xu}(x)||\ ||diag(\tilde{W}_{uu})^{-1}\tilde{W}_{xu}|| + ||\varepsilon_{Qu}|| \tag{3.44}$$

bounded after finite time t_2 and it depends on the weight error $\tilde{W}$ and the approximation error ε_{Qu}. It follows the actor u will converge to a small bounded set around its optimal solution u^*. □

Remark 3.5 *Compared to the GPI method (Theorem 3.1), the proposed Q-learning algorithm (Theorem 3.2) further relaxes the requirement for the a priori knowledge of $g(x)$, which is a completely model-free approach solving the continuous-time nonlinear optimal control problem online. It does not restrict Q-learning to linear cases as in [158]-[162] and the actor neural network in [162] is not needed due to the finite-time convergence of the adaptive critic. Unlike other iterative model-free algorithms [160][163], the method does not require an initial stabilising control policy.* ◇

3.5 Numerical Examples

In order to demonstrate the effectiveness of our theoretical result, we consider a numerical example [151] for a continuous-time nonlinear affine system (3.1) with $x = [x_1\ x_2]^\mathsf{T} \in \mathbb{R}^2$, $u \in \mathbb{R}$, and

$$f(x) = \begin{bmatrix} -x_1 + x_2 \\ -0.5x_1 - 0.5x_2(1 - (\cos(2x_1) + 2)^2) \end{bmatrix} \tag{3.45}$$

$$g(x) = \begin{bmatrix} 0 \\ \cos(2x_1) + 2 \end{bmatrix} \tag{3.46}$$

We define the infinite horizon value function $V^u(x)$ to be minimized with $Q(x) = x_1^2 + x_2^2$ and $R = 1$. Using the converse procedure [167], the optimal value function is $V^* = \frac{1}{2}x_1^2 + x_2^2$ and the optimal control is $u^* = -(\cos(2x_1) + 2)x_2$.

3.5.1 Adaptive Optimal Control via GPI (Theorem 3.1)

We implement the GPI algorithm as in *Theorem 3.1*. The activation function $\varphi(x)$ of the adaptive critic neural network (3.15) is selected as $\varphi(x) = [x_1^2 \;\; x_1x_2 \;\; x_2^2]^\mathsf{T}$ with the number of neurons $N = 3$. We initialize the state $x(0) = [1 \;\; 1]^\mathsf{T}$ and the weight $\hat{w}(0) = [0.1 \;\; 0.1 \;\; 0.1]^\mathsf{T}$. The tuning parameters are properly chosen as follows. The sample period $T = 2$s, the filter parameter $\ell = 1$, the adaptive learning gain $\Gamma_1 = I$. The system state trajectory is presented in Fig. 3.2 with the exploration noise removed after 100s. Fig. 3.3 presents the weight convergence of the adaptive critic (3.15). The PE condition is ensured by adding onto the control input a small exploration noise that can satisfy the state to remain PE until the weights converge. It is reasonable to add the noise onto input instead of the states because the input can be adjusted by the controller and the states are practically not capable of being manipulated. A typical type of exploration noise is a set of sinusoidal signals with different frequencies. Empirically, the number of different frequencies should be at least the number of weights to be estimated in order to get good convergence. The result shows the neural network weight $\hat{w}$ converges to $w = [0.49 \;\; 0.01 \;\; 1.02]^\mathsf{T}$, which is close to the optimal value $w = [0.5 \;\; 0 \;\; 1]^\mathsf{T}$.

3.5.2 Adaptive Optimal Control via Q-learning (Theorem 3.2)

We implement the Q-learning algorithm as in *Theorem 3.2*. The activation function $\Phi(x,u)$ of the adaptive critic neural network (3.15) is selected as $\Phi(x,u) = [x_1^2 \;\; x_1x_2 \;\; x_2^2 \;\; x_1u \;\; x_2u \;\; x_1x_2u \;\; x_1^2u \;\; x_2^2u \;\; x_1^2x_2u \;\; x_1x_2^2u \; x_1^4x_2u \;\; x_1x_2^4u \;\; u^2]^\mathsf{T}$ with the number of neurons $N' = 13$. We initialise the state $x(0) = [1 \;\; 1]^\mathsf{T}$ and the weight $\hat{W}(0) = [0.5 \;\; 0.5 \;\; 0.5 \;\; 0.5 \;\; 0.5 \;\; 0.5 \;\; 0.5 \;\; 0.5 \;\; 0.5 \;\; 0.5 \;\; 0.5 \;\; 0.5 \;\; 1]^\mathsf{T}$. The tuning parameters are chosen as the sample period is $T = 2$s, the filter parameter is $\ell = 1$, the adaptive learning

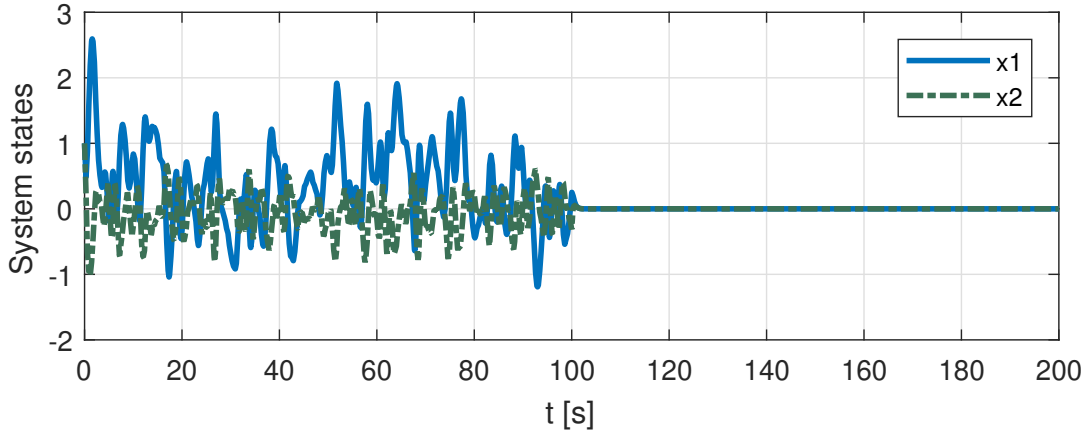

Figure 3.2: System trajectory with exploration noise with GPI.

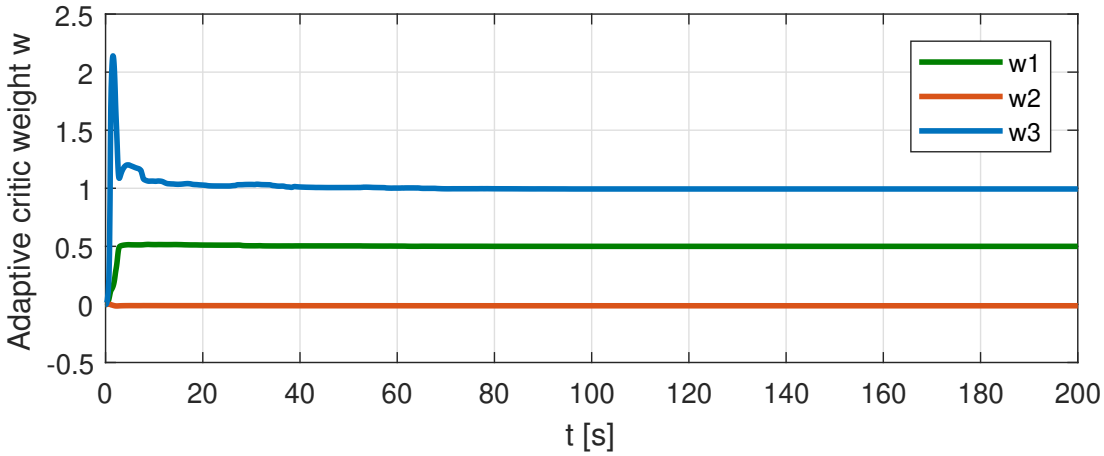

Figure 3.3: The weight convergence of the adaptive critic (3.15) in GPI.

gain is $\Gamma_2 = I$. Fig. 3.4 presents the system state trajectory with exploration noise removed after 100s. Fig. 3.5 shows the neural network weight $\hat{W}$ converges around 80s before removing the noise. Using the Taylor series for $\cos(2x_1)$, the optimal value $u^* = -(\cos(2x_1) + 2)x_2 \approx -\frac{1}{2}(6x_2 - 4x_1^2 x_2)$ for small x_1, i.e. $W_5 \approx 6$, and $W_9 \approx -4$. One can verify the optimal weight convergence by checking the value of $\hat{W}_5$, and $\hat{W}_9$. After 80s, the critic weights converge to the values of $\hat{W}_5 = 5.76$, and $\hat{W}_9 = -3.64$, which are close to the optimal values.

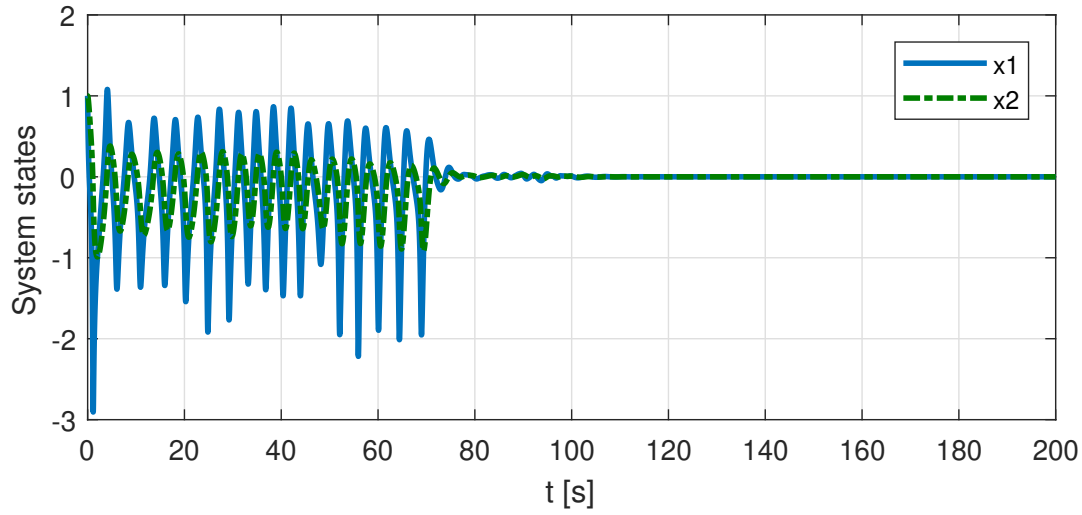

Figure 3.4: System trajectory with exploration noise with Q-learning.

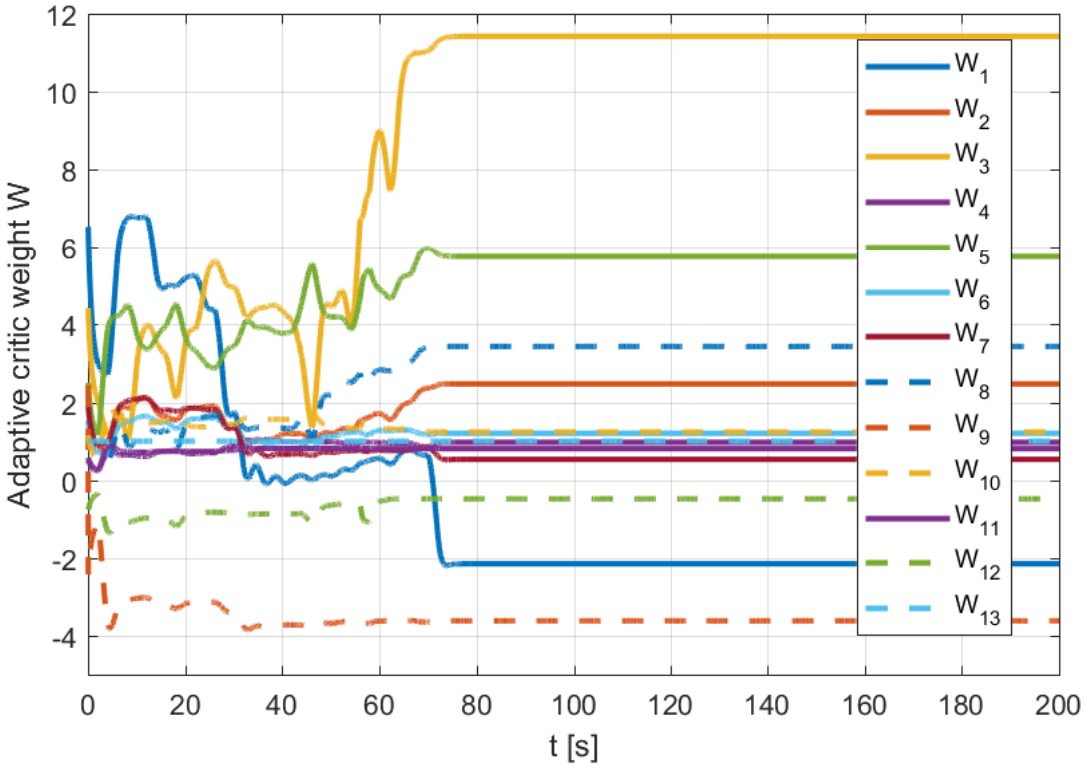

Figure 3.5: The weight convergence of the adaptive critic (3.30) in Q-learning.

3.6 Conclusions

In this chapter, we have provided two novel adaptive optimal control algorithms for continuous-time nonlinear input-affine systems using reinforcement learning ideas, i.e. GPI and Q-learning. The adaptive critic and actor are continuously and simultaneously updating each other with neither iterative steps nor an initial stabilising policy. The two approaches can approximate the value function/Q-function *online* and are partially/completely model-free. The new adaptive approach enables the online verification of the PE condition and guarantees the overall stability and finite-time convergence. The next chapter will extend this result to an observer design problem.

CHAPTER 4

A First Model-Free Adaptive Optimal Observer: An OHJB Approach

This chapter presents an adaptive optimal observer design using reinforcement learning/approximate dynamic programming (ADP) principles for deterministic nonlinear systems. A continuous-time Q-learning algorithm is proposed to solve the problem online while ensuring stability and optimality. Optimal observer design has been rarely studied beyond the Kalman filter. We first formulate the general optimal observer design problem for deterministic nonlinear systems, i.e., finding an admissible control that minimises a pre-defined cost functional. We show that the optimal solution can be obtained by solving an observer Hamilton-Jacobi-Bellman (OHJB) equation. Specifically, we justify the existence, stability, and optimality of the solution when the problem is in infinite horizon. This allows us to build general results in policy iteration that successively approximate the optimal value functional. Then, we define the Q-functional in the continuous-time context as a control-dependent value functional. The Q-functional is approximated by an adaptive critic neural network that online solves the Q-learning Bellman equation. The convergence is rigorously proved via an overall Lyapunov stability analysis. The pro-

posed algorithm is model-free in the sense that the Q-learning Bellman equation can be approximately solved without knowing the system dynamics. A case study on observer design for the Van der Pol oscillator is provided. Numerical simulations demonstrate the effectiveness of the proposed algorithm compared with a high-gain observer.

4.1 Introduction

Estimating the state of a dynamical system has always been an important issue in general systems and control theory. Given that often not all the states are available in practice, one can design an “observer" that reconstructs or approximates the state using available input and output. For linear systems, the observer design problem has been extensively studied and especially applied as observer-based control on account of the separation principle [168]. In a wide sense, an observer is termed as the Luenberger observer [169][170] if it is in a deterministic setting, to distinguish it from the Kalman filter [171] which is usually stochastic in noise. Since this early seminal work in linear observers, many researchers have extended the theory to nonlinear problems so as to deal with a broader class of systems: deterministic/stochastic; continuous/discrete-time; time-invariant/varying. Detailed surveys can be found in [172][173][42][174][175] and references therein. Compared to the satisfactory results for linear systems, nonlinear observer designs still suffer from a significant lack of generality without a unified framework. The majority of literature on the subject contains scattered solutions under specific assumptions using different nonlinear techniques. The most celebrated results include the high-gain observer [176], the extended Kalman filter (EKF) [42], adaptive observers [177][178], sliding-mode observers [179], etc.

A common feature of observer designs is that certain compensation or correction (e.g., output error) is added to drive the observer states to track the actual dynamical system state. However, observers are not usually designed to be optimal in the sense of

minimising a user-prescribed performance index. A special case for an observer to be "optimal" is the Kalman filter and its relatives, e.g., the EKF, the unscented Kalman filter, the particle filter, where the system and measurement are corrupted by noise with certain stochastic properties such that the expected squared error is minimised. To be more specific, the Kalman filter is only optimal when the errors (including noises) are Gaussian and it also requires the specification of the covariance matrices of both the system and measurement noise. A control designer can be hard-pressed when finding reasonable values for covariance, let alone if the system itself is essentially deterministic or noise-free. Therefore, we are looking for an alternative optimal observer design procedure for essentially deterministic problems.

Optimal observer design has been rarely studied beyond the Kalman filter. The duality of Kalman filter and linear quadratic regulation (LQR) was well established in [33]. Recent work by Possieri and Sassano [180] provided a deterministic characterisation of optimality of the Kalman filter for linear time-invariant systems. The most interesting point of [180] is that a "linear quadratic optimal observer" was proposed which was radically different from the results derived in the duality theory. The observer design problem can be formulated as designing the correction term of a Luenberger observer as if it is an LQR optimal control problem, which is distinguished from the deterministic Kalman filter that solves a dual *algebraic Riccati equation* (ARE). Other similar formulations are found in [181], [182], and [183], which mostly involve solving an ARE to design a linear optimal observer. Following these new ideas, in this chapter, we synthesise the design of a nonlinear observer in an optimal control theoretic sense.

Optimal control [30][184][35] is primarily derived by offline solving the *Hamilton-Jacobi-Bellman* (HJB) equation, or, in a linear quadratic case, the ARE. The nonlinear HJB equation is often difficult or impossible to solve due to requirement of the complete knowledge of the system. On the other hand, adaptive control [185][25][26] learns online to control unknown systems using data measured in real time along

the system trajectories. Recent ideas of incorporating reinforcement learning [85][159] [41][144] into feedback control have prompted extensive research on adaptive optimal control [41]. This is also referred to as approximate/adaptive dynamic programming (ADP) [84][160][152]. Vrabie *et al.* [148] proposed an adaptive optimal controller for the fundamental LQR problem which forms an integral reinforcement learning (IRL) approach in continuous time. The IRL [150] then generates a large family of algorithms but most of them require complete or at least partial knowledge of the system dynamics.

As a model-free reinforcement learning technique, Q-learning has been developed mainly for discrete-time systems under the Markov decision process (MDP) framework [85]. Nonetheless, there is still a lack of research regarding Q-learning studies in a deterministic continuous-time case. A proper definition of a Q-function in continuous time is still disputed. In fact, Q-learning was first posed in continuous time as advantage updating [156]. It was mentioned in [157] that the Q-function can be viewed as an extension of the Hamiltonian, which connects Q-learning with continuous-time optimal control. Then researchers started to mimic the Q-learning from discrete-time systems to continuous-time systems, which resulted in stepwise iterative algorithms in [158][160][161]. The consequence is that the control signal follows inherently a discrete-time batch learning process while the system is in continuous time. Furthermore, it should be noted that the “Q-function" in [157][158] has a different meaning from that in reinforcement learning [159]. Later, the above issues were overcome by Vamvoudakis [162] via a synchronous method for IRL, where the algorithm employed two neural networks in a critic/actor configuration and was restricted to the LQR case. Indeed, all the solutions above [158]-[162] were limited to linear systems. Chen and Herrmann [143] for the first time extended Q-learning to nonlinear input-affine systems in a non-iterative manner. Instead of the gradient algorithm with normalisation in [162], the adaptive law in [143] used a sliding mode technique which guarantees the convergence towards the optimal solution in finite time.

This chapter presents the formulation of an optimal observer design problem for a class of deterministic continuous-time nonlinear systems and its observer Hamilton-Jacobi-Bellman (OHJB) solution. The successive approximation theory used in IRL policy iteration is reviewed which forms the basis for new Q-learning algorithm. We provide the definition and parameterisation of the Q-functional and the Q-learning Bellman equation. The online adaptive algorithm ensures the convergence of the critic neural network. The overall stability is rigorously proved by Lyapunov analysis. We also provide a case study of an observer design for the Van der Pol oscillator and compare the results with benchmark high-gain observers.

4.2 Nonlinear Optimal Observer Design Problem Formulation and Its OHJB Solution

4.2.1 Problem Formulation

Consider a nonlinear system

$$\begin{cases} \dot{\bar{x}}(t) = A\bar{x}(t) + f(\bar{y}(t), \bar{u}(t)), \quad \bar{x}(t_0) = \bar{x}_0 \\ \bar{y}(t) = C\bar{x}(t) \end{cases} \tag{4.1}$$

where $\bar{x}(t) \in \mathbb{R}^n$ is the system state, $\bar{y}(t) \in \mathbb{R}^q$ is the system output, $\bar{u}(t) \in \mathbb{R}^m$ is the system control input, t is time, t_0 is the initial time, and x_0 is the initial state. Both $\bar{x}(t)$ and $u(t)$ are continuous functions of t. The additive output nonlinearity $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ is Lipschitz continuous on $\bar{x}(t)$ and $\bar{u}(t)$. The pair (A, B) is controllable and (A, C) is observable. The system (4.1) admits a Luenberger-like observer of the form

$$\dot{x}(t) = Ax(t) + f(\bar{y}(t), \bar{u}(t)) - K(Cx(t) - \bar{y}(t)) \tag{4.2}$$

with appropriate choice of K such that $A - KC$ is stable [172]. We show that the above observer problem can be viewed as an optimal tracking control problem. The observer state $x(t)$ in (4.2) should be controlled to track the system state $\bar{x}(t)$. The term $-K(Cx - \bar{y})$ in (4.2) can be seen as some correction term (control input) to be designed.

Therefore, one can rewrite a new form of the state observer for the system (4.1) from a Luenberger-like observer as

$$\begin{cases} \dot{x}(t) = Ax(t) + f(\bar{y}(t), \bar{u}(t)) + Bu(t), \quad x(t_0) = x_0 \\ y(t) = Cx(t) \end{cases} \tag{4.3}$$

where $x(t) \in \mathbb{R}^n$, $y(t) \in \mathbb{R}^q$, $u \in U \subset \mathbb{R}^m$ with U being a compact set of admissible [186] controls (See *Definition 4.1*).

We use a low-pass filter for the output

$$k\dot{y}_f(t) + y_f(t) = \bar{y}(t) \tag{4.4}$$

with $k > 0$ being the filter's time constant. The benefits of using the filtered output $y_f(t)$ are twofold: 1) smoothing the output signal $\bar{y}(t)$, if subject to high-frequency noise; 2) constructing filtered output dynamics which later turn out to be essential for learning. In practice, one may choose an arbitrarily small k so that $y_f(t) \approx \bar{y}(t)$.

Considering the system (4.1) with the initial condition $\bar{x}(t_0) = \bar{x}_0$, the nonlinear optimal observer (4.3) shall generate an estimated state trajectory $x(t)$ for $\bar{x}(t)$ from the knowledge of $\bar{u}(t)$, $u(t)$, $\bar{y}(t)$, and $y_f(t)$ for $t_0 \le t \le t_f$ by minimising some performance index or cost functional

$$\begin{aligned} &J_0(t_0, x(t), y_f(t), u(t)) \\ &\quad := \int_{t_0}^{t_f} [(Cx - y_f)^T L (Cx - y_f) + u^T R u] dt \\ &\qquad + (Cx(t_f) - y_f(t_f))^T P (Cx(t_f) - y_f(t_f)) \end{aligned} \tag{4.5}$$

where L and P are real symmetric positive semi-definite matrices, R is a real symmetric positive definite matrix, i.e., $L = L^T \succeq 0$, $P = P^T \succeq 0$, $R = R^T \succ 0$; the initial point $\{t_0, x_0.y_f(t_0)\}$ is specified while the final point $\{t_f, x(t_f), y_f(t_f)\} \in S_0 \subset [t_0, \infty) \times \mathbb{R}^n \times \mathbb{R}^p$ with S_0 being some compact target set. The cost functional (4.5) associated with the quadratic terms is reasonable: since all matrices L, R, and P are positive (semi-) definite, it penalises the size of observer error, the correction (control) effort, and the

terminal observer error, with L, R, and P determining their relative weights, respectively.

In general, we can define an augmented state $X(t)$ as

$$X(t) := \begin{bmatrix} x(t) \\ y_f(t) \end{bmatrix}; \quad X(t_0) = X_0 \tag{4.6}$$

and a more generic cost functional $J(t_0, X(t), u(t))$ such that

$$\begin{aligned} J(t_0, X(t), u(t)) := & \int_{t_0}^{t_f} \mathcal{L}(t, X(t), u(t)) dt \\ & + \mathcal{M}(t_f, X(t_f)) \end{aligned} \tag{4.7}$$

where $\mathcal{L}(t, X(t), u(t)) > 0$, $\mathcal{M}(t_f, X(t_f)) \geq 0$, and $\{t_f, X(t_f)\} \in S \subset [t_0, \infty) \times \Omega$ with S being some compact target set. To ensure the observer design problem is well posed, we shall make the following assumptions

Assumption 4.1 *There exists a compact set $\Omega \subset \mathbb{R}^{n+p}$ containing the origin as an interior point such that the augmented state trajectory $X(t) \in \Omega$ is selected that the system* (4.1) *and its observer* (4.3) *are stabilisable on Ω. To be precise, the reference trajectory $\bar{x}(t)$ so as $\bar{y}(t)$, $y_f(t)$ remain bounded for a stabilising control $\bar{u}(t) \in \bar{U}$ and there exists a continuous control $u = \mu(X)$ on Ω so that the observer* (4.3) *is asymptotically stable for all initial conditions $X_0 \in \Omega$ and for $\bar{y} = 0$.* ◇

Assumption 4.2 *(A, B) is controllable and (A, C) is observable.* ◇

Definition 4.1 *(Admissible control) Given the system* (4.1) *and its observer* (4.3)*, a feedback control policy $u = \mu(X) \in U(\Omega)$ is said to be admissible with respect to* (4.7) *on Ω if*

- *$\mu(X)$ is a continuous function on Ω,*
- *$\mu(0) = 0$,*
- *$u = \mu(X)$ stabilises* (4.3)*,*

- $J(t_0, X(t), \mu(t) \in \mathcal{L}^\infty$, $\forall t, X(t)$.

$\diamond$

Now we formulate the optimal observer design problem as an optimal tracking control problem:

Problem 4.1 *Considering a nonlinear uniformly observable system* (4.1) *and its state observer given by* (4.3)*, find an admissible correction (control) input* $u(t) \in U$ *that minimises the cost functional* (4.7)*.*

In the above discussion, the specification of the compact set Ω has been somewhat arbitrary. To be specific, Ω can be made as large as the region of attraction of the system under closed-loop stabilising controls $\bar{u}$ and u.

4.2.2 OHJB Equation

Let us begin by supposing that the initial point $\{t_0, X_0\}$ is specified and $\bar{u}$ is a stabilising control. The control $u(t)$ is an admissible control which transfers $\{t_0, X_0\}$ to the target set S. The observer state trajectory $X(t)$ originating at x_0 is generated by $u(t)$ and t_f is the first instant of time when $X(t)$ meets S. Thus, for $t \in [t_0, t_f]$, the control $u(t)$ will transfer $\{t, X(t)\}$ to S in view of the transition property of dynamical systems. Then, we have the cost-to-go value functional defined as

$$\begin{aligned} V(t, X(t)) &:= J(t, X(t), u(t)) \\ &= \int_t^{t_f} \mathcal{L}(\tau, X(\tau), u(\tau)) d\tau + \mathcal{M}(t_f, X(t_f)) \end{aligned} \tag{4.8}$$

for $t \in [t_0, t_f]$ being the lower limit of the integral.

Define the Hamiltonian $\mathcal{H}(\cdot)$ of our problem for some control $u(t)$ and its associated

$V(t,X(t))$:

$$\mathcal{H}(t,X(t),\frac{\partial V(t,X(t))}{\partial X(t)},u(t)):= \mathcal{L}(t,X(t),u(t))+\langle\frac{\partial V(t,X(t))}{\partial X(t)},\dot{X}(t)\rangle \tag{4.9}$$

Lemma 4.1 *If there exists an admissible feedback control $u=\mu(X)\in U$, $t\in[t_0,t_f]$ such that its associated value functional $V(t,X(t))\in\mathcal{L}^\infty(\Omega)\cap\mathcal{C}^1(\Omega)$, then the value function satisfies the generalised Hamilton-Jacobi-Bellman equation, written GHJB$(V,u)=0$, as*

$$\frac{\partial V(t,X(t))}{\partial t}+\mathcal{H}(t,X(t),\frac{\partial V(t,X(t))}{\partial X(t)},u(t))=0, \quad \{t,X(t)\}\in[t_0,t_f]\times\Omega \tag{4.10}$$

with the boundary condition

$$V(t_f,X(t_f))=\mathcal{M}(t_f,X(t_f)),\quad \{t_f,X(t_f)\}\in S \tag{4.11}$$

Conversely, if there exists a positive definite solution $V(t,X(t))\in\mathcal{L}^\infty(\Omega)\cap\mathcal{C}^1(\Omega)$ that satisfies GHJB$(V,u)=0$ (4.10) along with the boundary condition (4.11), then it is the value functional for the problem, i.e.

$$V(t,X(t))\equiv J(t,X(t),u(t)),\quad \{t,X(t)\}\in[t_0,t_f]\times\Omega \tag{4.12}$$

◇

Proof. First result: If there exists an admissible feedback control $u=\mu(X)\in U$, $t\in[t_0,t_f]$ and its associated value functional, using (4.8), $V(t,X(t))$ can be expanded for a time interval T as

$$\begin{aligned} V(t,X(t)) &= \int_t^{t+T}\mathcal{L}(\tau,X(\tau),u(\tau))d\tau \\ &\quad+\int_{t+T}^{t_f}\mathcal{L}(\tau,X(\tau),u(\tau))d\tau+\mathcal{M}(t_f,X(t_f)) \\ &= \int_t^{t+T}\mathcal{L}(\tau,X(\tau),u(\tau))d\tau+V(t+T,X(t+T)) \end{aligned} \tag{4.13}$$

Taking the limit as $T\to 0$ yields

$$\begin{aligned} &\lim_{T\to 0}\frac{V(t+T,X(t+T))-V(t,X(t))}{T} \\ &=-\lim_{T\to 0}\frac{1}{T}\int_t^{t+T}\mathcal{L}(\tau,X(\tau),u(\tau))d\tau \end{aligned} \tag{4.14}$$

Since $V(t,X(t)) \in \mathcal{L}^\infty(\Omega) \cap \mathcal{C}^1(\Omega)$, we have

$$\begin{aligned} \dot{V}(t,X(t)) &= \frac{\partial V(t,X(t))}{\partial t} + \frac{\partial V(t,X(t))}{\partial X(t)} \frac{\partial X(t)}{\partial t} \\ &= -\mathcal{L}(t,X(t),u(t)) \end{aligned} \tag{4.15}$$

Using the Hamiltonion (4.9), the above becomes (4.10). The boundary condition (4.11) can be obtained by setting $t = t_f$ in (4.8).

Second result: If there exists a positive definite solution $V(t,X(t)) \in \mathcal{L}^\infty(\Omega) \cap \mathcal{C}^1(\Omega)$, then

$$\begin{aligned} &V(t_f,X(t_f)) - V(t,X(t)) \\ &= \int_t^{t_f} [\frac{\partial V(\tau,X(\tau))}{\partial \tau} + \frac{\partial V(\tau,X(\tau))}{\partial X(\tau)} \frac{\partial X(\tau)}{\partial \tau}] d\tau \end{aligned} \tag{4.16}$$

Adding to the above both sides of

$$J(t,X(t),u(t)) = \int_t^{t_f} \mathcal{L}(\tau,X(\tau),u(\tau)) d\tau + \mathcal{M}(t_f,X(t_f)) \tag{4.17}$$

and using the Hamiltonian (4.9), we have

$$\begin{aligned} &J(t,X(t),u(t)) - V(t,X(t)) \\ &= \int_t^{t_f} (\frac{\partial V}{\partial \tau} + \mathcal{H}) d\tau + \mathcal{M}(t_f,X(t_f)) - V(t_f,X(t_f)) \end{aligned} \tag{4.18}$$

Since $V(t,X(t))$ satisfies GHJB$(V,u) = 0$ (4.10) along with the boundary condition (4.11), we have

$$J(t,X(t),u(t)) - V(t,X(t)) \equiv 0 \tag{4.19}$$

This completes the proof. □

Remark 4.1 *The GHJB equation has also been referred as the (nonlinear) Lyapunov equation in the literature [30][164]. In this chapter, we adopt the GHJB equation following the terminology in [186]. In the linear time-invariant case, the GHJB equation reduces to the Lyapunov equation, and the HJB equation reduces to the well-known algebraic Riccati equation.* ◇

Lemma 4.2 *If there exists a unique optimal control $u^*(X)$ and the associated optimal value functional*

$$V^*(t,X(t)) = \min_{u \in U} V(t,X(t)) \tag{4.20}$$

then they satisfy Lemma 4.1, i.e., $V^*(t,X(t))$ is the unique solution to the Hamilton-Jacobi-Bellman (HJB) equation

$$\frac{\partial V^*(t,X(t))}{\partial t} + \min_{u\in U} \mathcal{H}(t,X(t),\frac{\partial V^*(t,X(t))}{\partial X(t)},u(t)) = 0, \quad \{t,X(t)\} \in [t_0,t_f]\times\Omega \tag{4.21}$$

with the boundary condition

$$V^*(t_f,X(t_f)) = \mathcal{M}(t_f,X(t_f)), \quad \{t_f,X(t_f)\} \in S \tag{4.22}$$

Then the optimal control is determined by

$$u^* = \arg\min_{u\in U} \mathcal{H}(t,X(t),\frac{\partial V^*(t,X(t))}{\partial X(t)},u(t)) \tag{4.23}$$

◇

Proof. The proof of this lemma is obvious from *Lemma 4.1* and the optimal control theory [30]. Equation (4.23) is from Pontryagin's minimum principle (a necessary condition for optimality), in which the optimal control shall minimise the Hamiltonian for all admissible controls. □

4.2.3 Infinite-Horizon Problem: Optimality and Stability

Note that the existence of a value functional is still questionable, let alone when the problem horizon is infinite, i.e., when $t_f \to \infty$. Even if the filtered output $y_f(t)$ is stable in the sense of Lyapunov, the resulting control $u(X)$ containing a steady-state part in general makes the cost functional $J(t,X(t),u(t))$ so as the value functional $V(t,X(t))$ unbounded, thus, the meaning of optimality (minimality) is lost. The existence of such a finite value functional is established only when i) the system (4.1) and the observer (4.3) are stabilisable (Assumption 4.1); and ii) the signal $y_f(t)$ is at least asymptotically stable [187]. This latter requirement is not favourable since we want the observer to keep tracking the system trajectory which is usually nonzero. To ensure an optimal value functional exists and is positive definite and finite even if the problem

horizon is infinite, we define a discounted cost functional for (4.7) such that

$$\begin{aligned}\mathcal{L}_0(t_0,t,X(t),u(t)) &= e^{-\gamma(t-t_0)}\mathcal{L}(t,X(t),u(t))\\ &= e^{-\gamma(t-t_0)}((Cx-y_f)^T L(Cx-y_f)+u^T Ru)\\ &= e^{-\gamma(t-t_0)}(X^T(t)L_a X(t)+u(t)^T Ru(t))\end{aligned} \tag{4.24}$$

$$\mathcal{M}(t_f,X(t_f))\equiv 0 \tag{4.25}$$

where $\gamma>0$ is a properly selected discount factor so that the cost functional is finite, i.e., $V\in\mathcal{L}^\infty(\Omega)$; the matrix $L_a=\begin{bmatrix}C^TLC & -C^TL\\ -C^TL & L\end{bmatrix}$ is chosen by a proper matrix L so that the cost functional (4.7) satisfies the zero-state observability [188] with the following definition.

Definition 4.2 *(Zero-state observability) The discounted cost functional* (4.7) *defined from* (4.24)(4.25) *satisfies the zero-state observability if for any trajectory such that* $\sqrt{L_a}X(t)\equiv 0$, $\bar{u}(t)\equiv 0$, $u(t)\equiv 0$ *implies that* $X(t)\equiv 0$ *for all* $t\geq t_0$. ◇

To be more specific, the following assumption is required.

Assumption 4.3 *The discount factor* γ *is sufficiently large so that the value functional in terms of* $\mathcal{L}_0$ *is essentially bounded, i.e.,* $V(t,X(t))\in\mathcal{L}^\infty$*; the matrix* L *is chosen properly so that* $(A,\sqrt{L_a})$ *is observable.* ◇

Remark 4.2 *Under Assumption 4.3,* $(A,\sqrt{L_a})$ *being observable implies that the augmented state* X *is "observable by the cost functional* (4.7)*" so that variations in any direction of the state have an effect on* $J(t)$*, i.e.,* $J(t)$ *satisfies the zero-state observability. This will guarantee the closed-loop dynamics are stabilisible since all the potentially unstable states are weighted in the cost functional. Moreover, the observability assumption ensures the uniqueness and positive definiteness of the solution of the HJB equation* (4.21)*. If* $(A,\sqrt{L_a})$ *is assumed to be detectable (all unobservable modes are convergent), only positive semi-definiteness can be concluded.* ◇

Remark 4.3 *The choice of the discount factor $\gamma > 0$ will depend on the dynamics of the filtered output y_f (driven by the control $\bar{u}$). If the filtered output y_f is asymptotically stable, γ is not essential, i.e., we can set $\gamma = 0$. If y_f is subject to some linear dynamics such as $\dot{y}_f = \boldsymbol{F} y_f$, the discount factor γ can be chosen such that $2\boldsymbol{F} - \gamma \boldsymbol{I}$ is Hurwitz [189]. In a nutshell, given any reasonable $y_f(t)$, we can always find a large enough γ to make the cost functional bounded.* $\diamond$

Theorem 4.1 *With the discounted cost functional* (4.7) *defined from* (4.24)(4.25)*, Lemma 4.2 holds for $t_f \to \infty$, i.e., $V(t, X(t)) \in \mathcal{L}^\infty(\Omega) \cap \mathcal{C}^1(\Omega)$* (4.8) *becomes an infinite-horizon value functional written as*

$$V(t, X(t)) = \int_t^\infty e^{-\gamma(\tau - t)} (X^T(\tau) L_a X(\tau) + u(\tau)^T R u(\tau)) d\tau \tag{4.26}$$

for all $t \in [t_0, \infty)$ and $\tau \in [t, \infty)$, which satisfies the following GHJB equation

$$\begin{aligned} &-\gamma V(t, X(t)) + X^T(t) L_a X(t) + u(t)^T R u(t) + \frac{\partial V(t, X(t))}{\partial t} \\ &+ \langle \frac{\partial V(t, X(t))}{\partial X(t)}, \dot{X}(t) \rangle = 0, \ \{t, X(t)\} \in [t_0, \infty) \times \Omega \end{aligned} \tag{4.27}$$

Moreover, suppose there exists a smooth positive definite solution $V^*(t, X(t)) \in \mathcal{L}^\infty(\Omega) \cap \mathcal{C}^1(\Omega)$ of the HJB equation

$$\begin{aligned} &\frac{\partial V(t, X(t))}{\partial t} + \min_{u \in U} \mathcal{H}(t, X, \frac{\partial V^*(t, X)}{\partial X}, u(X)) = 0, \\ &\{t, X(t)\} \in [t_0, \infty) \times \Omega \end{aligned} \tag{4.28}$$

Define $u^*(X)$ as

$$u^*(X) = -\frac{1}{2} R^{-1} B^T \frac{\partial V^*(t, X(t))}{\partial X(t)} \tag{4.29}$$

Then under a stabilising control $\bar{u}(t)$ for (4.1) *and $u = u^*(X)$,*

- *(Stability) The augmented state $X(t)$ and the observer output error $\epsilon(t) = Cx(t) - y_f(t)$ are exponentially bounded. Moreover, in the limit as the discount factor γ goes to zero (which can only be chosen when the observed plant $lim_{t \to \infty} \bar{x}(t) = 0$), the observer output error $\epsilon(t) = 0$ is asymptotically stable.*

- *(Optimality) The control $u = u^*(X)$ is the optimal control which minimises the cost functional* (4.7) *over all admissible controls in U, and $V^*(t, X(t))$ is the optimal (minimal) value functional.*

Proof. Note t denotes the lower limit of the integration interval in the value functional (4.8). Replacing the time variables t_0 and t in (4.24) with t and τ yields

$$\begin{aligned} &\mathcal{L}_0(t, \tau, X(\tau), u(\tau)) \\ &\quad = e^{-\gamma(\tau - t)}(X^T(\tau) L_a X(\tau) + u(\tau)^T R u(\tau)) \end{aligned} \tag{4.30}$$

Combining (4.30) with (4.8)(4.25) gives the discounted infinite-horizon value functional $V(t, X(t))$ (4.26).

Differentiating $V(t, X(t))$ (4.26) along the augmented state trajectories $X(t)$ via the Leibniz integral rule yields

$$\begin{aligned} \dot{V}(t, X(t)) = \gamma V(t, X(t)) - X^T(t) L_a X(t) - u(t)^T R u(t), \\ \{t, X(t)\} \in [t_0, \infty) \times \Omega \end{aligned} \tag{4.31}$$

We know from (4.15) that the right hand side of (4.31) is actually $-\mathcal{L}(t, X(t), u(t))$. The Hamiltonian (4.9) becomes

$$\begin{aligned} &\mathcal{H}(t, X(t), \frac{\partial V(t, X(t))}{\partial X(t)}, u(t)) := X^T(t) L_a X(t) \\ &\quad + u(t)^T R u(t) - \gamma V(t, X(t)) + \langle \frac{\partial V(t, X(t))}{\partial X(t)}, \dot{X}(t) \rangle \end{aligned} \tag{4.32}$$

Hence, the GHJB equation (4.27) and the HJB equation (4.28) follow on from (4.10) in *Lemma 4.1* and (4.21) in *Lemma 4.2*, respectively.

Stability: This stability proof follows [190]. The value functional $V(t, X(t))$ is a natural Lyapunov function candidate for examining the boundedness of $X(t)$ and $\epsilon(t)$. Rearrange (4.31) as

$$\begin{aligned} \dot{V}(t, X(t)) - \gamma V(t, X(t)) = -X^T(t) L_a X(t) - u(t)^T R u(t), \\ \{t, X(t)\} \in [t_0, \infty) \times \Omega \end{aligned} \tag{4.33}$$

Multiplying $e^{-\gamma t}$ its both sides gives

$$\begin{aligned} &\frac{d}{dt}(e^{-\gamma t}V(t,X(t))) = \\ &\qquad -e^{-\gamma t}(X^T(t)L_aX(t)+u(t)^TRu(t)) \le 0, \\ &\qquad\qquad \{t,X(t)\} \in [t_0,\infty)\times\Omega \end{aligned} \tag{4.34}$$

so the augmented state $X(t)$ and the observer error $\epsilon(t) = Cx(t) - y_f(t)$ are ultimately bounded. Based on LaSalle's extension, the augmented state $X(t)$ goes to a region of attraction where $\dot{V}(t,X(t)) = 0$. In the limit as the discount factor γ goes to zero (which can only be chosen when $\lim_{t\to\infty}\bar{x}(t) = 0$), it can be concluded that the observer output error $\epsilon(t) = 0$ is asymptotically stable since $\lim_{t\to\infty}\bar{x}(t) = 0$, it follows $\lim_{t\to\infty}Cx(t) = 0$ and zero-state observability implies $\lim_{t\to\infty}x(t) = 0$.

Optimality: The solution $V^*(t,X(t))$ to the HJB equation (4.28) is the optimal value functional as shown in *Lemma 4.2*. From (4.23), we know that the optimal control shall minimise the Hamiltonian (Pontryagin's minimum principle). Since $u(t)$ is unconstrained, letting $\partial\mathcal{H}/\partial u = 0$ by (4.9) yields

$$u = -\frac{1}{2}R^{-1}B^T\frac{\partial V(t,X(t))}{\partial X(t)} \tag{4.35}$$

Hence, for the optimal value functional

$$V^*(t,X(t)) = \min_{u\in U} V(t,X(t)) \tag{4.36}$$

We can conclude that (4.29) is the optimal control. □

In order to find the optimal control solution for the problem one only needs to solve the OHJB equation (4.28) for the value functional and then substitute the solution in (4.29) to obtain the optimal control.

Remark 4.4 *Optimal control problems do not necessarily have a smooth or even continuous value functional, but may have the so-called viscosity solutions. Various assumptions guarantee the existence of smooth solutions to the GHJB equation and the HJB equation, such as that*

there are no cross-terms of the state and control in the integrand $\mathcal{L}(t,X,(t),u(t))$ and the dynamics is not bilinear [30]. In this chapter, all derivations are performed under the assumption of a smooth solution, i.e., $V \in \mathcal{C}^1(\Omega)$. ◇

4.3 Policy Iteration: Successive Approximation Theory

The OHJB equation (4.28) is a nonlinear partial differential equation of $V^*(t,X)$ which is difficult or impossible to solve. However, it is easy to see that the observer GHJB equation (4.27) is linear in $V(t,X)$. The successive approximation theory was first developed in [191] where a sequence of GHJB equations is used to successively improve a given initial admissible control. The theory resulted in a family of offline algorithms in [164][186], the online version was proposed in [149] as a policy iteration algorithm. Policy iteration is an iterative method of reinforcement learning [159] for solving optimal control problems, and consists of policy evaluation based on (4.27) and policy improvement based on (4.29).

We present a policy iteration algorithm for the optimal observer design problem in the following.

Algorithm 1 (Policy iteration via the GHJB equation)

1. (Policy Evaluation) Given the admissible policy (control) $u^{(i)}$, solve for $V^{(i)}(t,X,u^{(i)})$ using the GHJB equation

$$\begin{aligned} &-\gamma V^{(i)}(t,X(t)) + X^T(t)L_a X(t) + u^{(i)T} R u^{(i)} \\ &\frac{\partial V^{(i)}(t,X(t))}{\partial t} + \langle \frac{\partial V^{(i)}(t,X(t))}{\partial X(t)}, \dot{X}(t) \rangle = 0 \end{aligned} \tag{4.37}$$

2. (Policy Improvement) Update the policy (control) using

$$u^{(i+1)} = -\frac{1}{2} R^{-1} B^T \frac{\partial V^{(i)}(t,X(t))}{\partial X(t)} \tag{4.38}$$

To ensure convergence of the policy iteration algorithm, an initial admissible policy (control) is required. Proofs of convergence have been given in [191][164][186]. This

method reduces to the well-known Kleinman iterative method [147] for solving the algebraic Riccati equation using the Lyapunov equations.

Theorem 4.2 *If $u^{(0)} \in U(\Omega)$, then $u^{(i)} \in U(\Omega)$, $\forall i$. Moreover, $V^{(i)} \to V^*$, $u^{(i)} \to u^*$ uniformly on Ω.*

Proof. It was shown in [164] that, conditioned by an initial admissible policy $u^{(0)} \in U(\Omega)$, all the subsequent policies by iterating on (4.37) and (4.38) will be admissible and will converge to the solution of the HJB equation, i.e., the optimal value functional V^*. The uniform convergence is proved by Dini's theorem [192] in view of Ω being a compact set. See [164] and [186] for a more detailed proof. □

Note that solving the GHJB equation (4.37) in policy iteration requires the complete knowledge of the system dynamics as $\dot{X}$ appears in (4.37). In order to find an equivalent formulation of the GHJB equation that does not involve the dynamics, we use the IRL idea from [149] for the optimal observer design problem and propose the following lemma.

Lemma 4.3 *For any time interval $[t-T, t]$ with $T > 0$, the value functional $V(t, X)$ satisfies the IRL Bellman equation*

$$
\begin{aligned}
&V(t-T, X(t-T)) = \\
&\int_{t-T}^{t} e^{-\gamma(\tau-t+T)}(X^T(\tau) L_a X(\tau) + u(\tau)^T R u(\tau)) d\tau \\
&+ e^{-\gamma T} V(t, X(t)), \\
&\qquad\qquad \{t, X(t)\} \in [t_0, \infty) \times \Omega
\end{aligned}
\tag{4.39}
$$

which is equivalent to the GHJB equation (4.27) *and has the same positive-definite solution.*⋄

Proof. We first derive the IRL Bellman equation by expanding $V(t-T, X(t-T))$ over

a time interval $[t-T,t]$ for $t \in [t_0,\infty)$.

$$
\begin{aligned}
&V(t-T,X(t-T)) \\
&= \int_{t-T}^{\infty} e^{-\gamma(\tau-t+T)}(X^T(\tau)L_aX(\tau)+u(\tau)^TRu(\tau))d\tau \\
&= \int_{t-T}^{t} e^{-\gamma(\tau-t+T)}(X^T(\tau)L_aX(\tau)+u(\tau)^TRu(\tau))d\tau \\
&\quad + \int_{t}^{\infty} e^{-\gamma(\tau-t+T)}(X^T(\tau)L_aX(\tau)+u(\tau)^TRu(\tau))d\tau \\
&= \int_{t-T}^{t} e^{-\gamma(\tau-t+T)}(X^T(\tau)L_aX(\tau)+u(\tau)^TRu(\tau))d\tau \\
&\quad + e^{-\gamma T}V(t,X(t)), \qquad\qquad \{t,X(t)\} \in [t_0,\infty)\times\Omega
\end{aligned}
\tag{4.40}
$$

To demonstrate the equivalence of the IRL Bellman equation (4.39) and the GHJB equation (4.27), we write the GHJB equation (4.27) as

$$
\begin{aligned}
&X^T(t)L_aX(t)+u(t)^TRu(t)-\gamma V(t,X(t)) \\
&\qquad + \frac{\partial V(t,X(t))}{\partial t} + \langle \frac{\partial V(t,X(t))}{\partial X(t)}, \dot{X}(t)\rangle = 0, \\
&\qquad\qquad\qquad\qquad \{t,X(t)\} \in [t_0,\infty)\times\Omega
\end{aligned}
\tag{4.41}
$$

Integrating (4.41) over $[t-T,t]$, we obtain

$$
\begin{aligned}
&\int_{t-T}^{t}(X^T(\tau)L_aX(\tau)+u(\tau)^TRu(\tau))d\tau \\
&= -\int_{t-T}^{t}(\dot{V}(\tau,X(\tau)-\gamma V(t,X(t)))d\tau \\
&\qquad\qquad\qquad\qquad \{t,X(t)\} \in [t_0,\infty)\times\Omega
\end{aligned}
\tag{4.42}
$$

Equation (4.42) does not account for the discount factor γ. However, we show that it is feasible to transform (4.42) into the IRL Bellman equation (4.39).

Multiplying $-e^{-\gamma t}$ to the both sides of (4.42) yields

$$
\begin{aligned}
&-\int_{t-T}^{t} e^{-\gamma t}(X^T(\tau)L_a X(\tau)+u(\tau)^T R u(\tau))d\tau \\
&\qquad = \int_{t-T}^{t} [e^{-\gamma t}\dot{V}(\tau,X(\tau)-e^{-\gamma t}\gamma V(t,X(t))]d\tau \\
&\qquad = \int_{t-T}^{t} \frac{d}{dt}(e^{-\gamma t}V(t,X(t)))d\tau \\
&\qquad = e^{-\gamma t}V(t,X(t))\Big|_{t-T}^{t} \\
&\qquad = e^{-\gamma t}V(t,X(t))-e^{-\gamma(t-T)}V(t-T,X(t-T)) \\
&\qquad\qquad \{t,X(t)\}\in[t_0,\infty)\times\Omega
\end{aligned}
\tag{4.43}
$$

Then, multiplying $e^{\gamma(t-T)}$ to the both sides (4.43) of gives

$$
\begin{aligned}
&-e^{\gamma(t-T)}\int_{t-T}^{t} e^{-\gamma t}(X^T(\tau)L_a X(\tau)+u(\tau)^T R u(\tau))d\tau \\
&\qquad = -\int_{t-T}^{t} e^{-\gamma(\tau-t+T)}(X^T(\tau)L_a X(\tau)+u(\tau)^T R u(\tau))d\tau \\
&\qquad = e^{-\gamma T}V(t,X(t))-V(t-T,X(t-T)) \\
&\qquad\qquad \{t,X(t)\}\in[t_0,\infty)\times\Omega
\end{aligned}
\tag{4.44}
$$

Rearrange (4.44) such that

$$
\begin{aligned}
&\int_{t-T}^{t} e^{-\gamma(\tau-t+T)}(X^T(\tau)L_a X(\tau)+u(\tau)^T R u(\tau))d\tau \\
&= V(t-T,X(t-T))-e^{-\gamma T}V(t,X(t)), \\
&\qquad\qquad \{t,X(t)\}\in[t_0,\infty)\times\Omega
\end{aligned}
\tag{4.45}
$$

This means that the solution of the GHJB equation (4.27) also satisfies the IRL Bellman equation (4.39). This completes the proof. □

Corollary 4.1 *For any time interval* $[t-T,t]$ *with* $T>0$*, the optimal value functional* $V^*(t,X)$

satisfies the Bellman optimality equation

$$
\begin{aligned}
&V^*(t-T, X(t-T)) = \\
&\int_{t-T}^{t} e^{-\gamma(\tau-t+T)}(X^T(\tau)L_a X(\tau) + u^*(\tau)^T R u^*(\tau))d\tau \\
&+ e^{-\gamma T} V^*(t, X(t)), \\
&\qquad\qquad \{t, X(t)\} \in [t_0, \infty) \times \Omega
\end{aligned} \tag{4.46}
$$

which is equivalent to the solution to the HJB equation (4.28). ◇

Proof. The proof is by substituting the optimal control $u = u^*$ into (4.39) in *Lemma 4.3*.

Remark 4.5 *Equation* (4.42) *is essentially the well-known integral reinforcement form in [41]. The difference between* (4.42) *and the IRL Bellman equation* (4.39) *is that* (4.39) *takes account of the forgetting factor* γ*. Hence, equation* (4.39) *is a more general form of* (4.42) *as it also considers the case of the integral reinforcement being discounted throughout* $[t_0, \infty)$*. The IRL Bellman equation* (4.39) *reduces to* (4.42) *when the forgetting factor* $\gamma = 0$*.* ◇

Lemma 4.3 enables the use of the IRL Bellman equation (4.39) for policy evaluation. The major benefit of it is that the following policy iteration algorithm does not require the knowledge of the system dynamics $\dot{X}$.

Algorithm 2 (Policy iteration via the IRL Bellman equation)

1. (Policy Evaluation) Given the admissible policy (control) $u^{(i)}$, solve for $V^{(i)}(t, X)$ using the IRL Bellman equation

$$
\begin{aligned}
&V^{(i)}(t-T, X(t-T)) = \\
&\int_{t-T}^{t} e^{-\gamma(\tau-t+T)}(X^T(\tau)L_a X(\tau) + u^{(i)T}(\tau)R u^{(i)}(\tau))d\tau \\
&+ e^{-\gamma T} V^{(i)}(t, X(t))
\end{aligned} \tag{4.47}
$$

2. (Policy Improvement) Update the policy (control) using

$$
u^{(i+1)} = -\frac{1}{2} R^{-1} B^T \frac{\partial V^{(i)}(t, X(t))}{\partial X(t)} \tag{4.48}
$$

4.4 Q-functional and Q-Learning Bellman Equation

We have the following Lemma [30] which is instrumental.

Lemma 4.4 *For any admissible control $u = \mu(t) \in U$ with its associated value functional $V(t,X(t))$, and the optimal control $u^*(X)$ given by* (4.29)*, we have*

$$\begin{aligned} \mathcal{H}(t,X,\frac{\partial V(t,X)}{\partial X(t)},\mu(t)) &= \mathcal{H}(t,X,\frac{\partial V(t,X)}{\partial X(t)},u^*(X)) \\ &+ (\mu(t)-u^*(X))^T R(\mu(t)-u^*(X), \\ &\{t,X(t)\} \in [t_0,\infty)\times\Omega \end{aligned} \tag{4.49}$$

◇

Proof. Expand the Hamiltonian

$$\begin{aligned} &\mathcal{H}(t,X,\frac{\partial V}{\partial X},u) = X^T L_a X + u^T R u + \langle \frac{\partial V}{\partial X}, \dot{X}\rangle \\ &= (Cx-y_f)^T L_a (Cx-y_f) + (\frac{\partial V}{\partial x})^T (Ax + f(\bar{y},\bar{u})) \\ &+ (\frac{\partial V}{\partial y_f})^T (\frac{\bar{y}-y_f}{k}) + (\frac{\partial V}{\partial x})^T Bu + u^T R u \end{aligned} \tag{4.50}$$

Completing the square for u yields

$$\begin{aligned} (\frac{\partial V}{\partial x})^T Bu + u^T R u &= -\frac{1}{4}(\frac{\partial V}{\partial x})^T B R^{-1} B^T \frac{\partial V}{\partial x} \\ &+ [\frac{1}{2}(\frac{\partial V}{\partial x})^T B R^{-1} + u^T] R [\frac{1}{2} R^{-1} B^T \frac{\partial V}{\partial x} + u] \end{aligned} \tag{4.51}$$

Inserting the optimal control $u^*(X)$ (4.29), then the difference of Hamiltonians becomes

$$\begin{aligned} &\mathcal{H}(t,X,\frac{\partial V}{\partial X},\mu(t)) - \mathcal{H}(t,X,\frac{\partial V}{\partial X},u^*(X)) \\ &= (\mu(t)-u^*(X))^T R(\mu(t)-u^*(X)) \end{aligned} \tag{4.52}$$

This completes the proof. □

It shows that the Hamiltonian is quadratic in the control deviation from the optimal control. This inpires us to create a Q-functional containing the Hamiltonian so as to quantify how far it differs from the optimal value functional, which results in an

action-dependent version for the value functional. This formulation will make use of the information from the online policy or control that leads to a model-free reinforcement learning framework.

Definition 4.3 *(Q-functional) A Q-functional $Q(t,X,u)$ for the optimal control Problem 4.1 is defined by adding the left-hand side of the GHJB equation* (4.10) *to the optimal value functional $V^*(t,X)$* (4.8) *such that*

$$Q(t,X,u) := V^*(t,X) + \frac{\partial V^*(t,X)}{\partial t} + \mathcal{H}(t,X,\frac{\partial V^*(t,X)}{\partial X},u) \tag{4.53}$$

Similar to the relation between $J(t_0,X(t,u(t)))$ and $V(t,X(t))$, if considering the discount factor γ with respect to an initial time t_0, a more sophisticated definition of the Q-functional should be

$$Q(t_0,X,u) := e^{-\gamma(t-t_0)}V^*(t,X) + \frac{\partial V^*(t,X)}{\partial t} + \mathcal{H}(t,X,\frac{\partial V^*(t,X)}{\partial X},u) \tag{4.54}$$

Note that (4.54) *reduces to* (4.53) *for every $t_0 = t$, i.e., the Q-functional defined as* (4.53) *is the free-initial-time version of* (4.54). ◇

Lemma 4.5 *Given the definition of the Q-functional* (4.53)*, the minimisation $Q^*(t,X,u^*) = \min_{u\in U} Q(t,X,u)$ is equivalent to the optimisation scheme $V^*(t,X) = \min_{u\in U} V(t,X)$. Furthermore, for any time $t \in [t_0,\infty)$, the optimal value $Q^*(t,X,u^*) = V^*(t,X)$.* ◇

Proof. Since $V^*(t,X) = \min_{u\in U} V(t,X)$, considering the HJB equation (4.28), we can

write

$$
\begin{aligned}
Q^*(t,X,u^*) &= \min_{u\in U} Q(t,X,u) \\
&= \min_{u\in U}\{V^*(t,X) + \frac{\partial V^*(t,X)}{\partial t} + \mathcal{H}(t,X,\frac{\partial V^*(t,X)}{\partial X},u)\} \\
&= V^*(t,X) \\
&\quad + \underbrace{\frac{\partial V^*(t,X)}{\partial t} + \min_{u\in U}\mathcal{H}(t,X,\frac{\partial V^*(t,X)}{\partial X},u(X))}_{=0 \text{ due to (4.28)}} \\
&= V^*(t,X)
\end{aligned}
\tag{4.55}
$$

It is obvious from the above that the minimiser u^* for the Q-functional is also the minimiser for the value functional, i.e., $u^* = \min_{u\in U} Q(t,X,u) = \min_{u\in U} V(t,X)$. In other words, the Q-functional has the same value of the value functional under the optimal trajectory. This completes the proof. □

Remark 4.6 *The definition of the Q-functional is different from the existing literature. Up to now, there is no standard characterisation of a Q-functional in the context of adaptive optimal control in continuous time. At an earlier time, Mehta and Meyn [157] for the first time related "Q-function" to the Hamiltonian in the minimum principle, but the "Q-function" in [157] had a different meaning from an action-dependent value function in reinforcement learning [159]. Vamvoudakis [162] later introduced a Q-function for a linear quadratic problem similar to that of reinforcement learning by adding the Hamiltonian onto the optimal value function. In this chapter, we justify a proper definition of Q-functional by adding the left-hand side of the GHJB equation onto the value functional.* ◇

The following lemma is extremely important throughout the chapter and will be used to prove the convergence of the proposed Q-learning algorithm.

Lemma 4.6 *For any time interval* $[t-T,t]$ *with* $T>0$*, the Q-functional* $Q(t,X(t),u(t))$ (4.54)

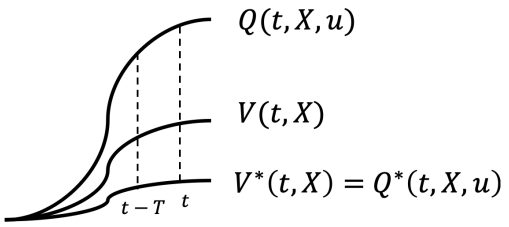

Figure 4.1: The relations between an Q functional and a value functional.

satisfies the Q-learning Bellman equation

$$\begin{aligned} &Q(t-T, X(t-T), u(t-T)) = \\ &\int_{t-T}^{t} e^{-\gamma(\tau-t+T)}(X^T(\tau)L_a X(\tau) + u(\tau)^T R u(\tau))d\tau \\ &+ e^{-\gamma T} Q(t, X(t), u(t)) + \psi, \\ &\qquad\qquad \{t, X(t)\} \in [t_0, \infty) \times \Omega \end{aligned} \tag{4.56}$$

with ψ being a residual error as

$$\begin{aligned} \psi = &-\int_{t-T}^{t} (u(\tau) - u^*(\tau))^{\mathsf{T}} R (u(\tau) - u^*(\tau)) d\tau \\ &+ (u(t-T) - u^*(t-T))^{\mathsf{T}} R (u(t-T) - u^*(t-T)) \\ &- (u(t) - u^*(t))^{\mathsf{T}} R (u(t) - u^*(t)) \end{aligned} \tag{4.57}$$

Moreover, equation (4.56) *reduces to the Bellman optimality equation* (4.46) *when $u(t)$ is the optimal control $u^*(t)$, i.e., solving for Q^* in* (4.56) *is equivalent to finding V^* in* (4.46). ⋄

Proof. We write the Hamiltonian $\mathcal{H}(t, X, \frac{\partial V^*(t,X)}{\partial X}, u)$ using (4.9) as

$$\mathcal{H}(t, X, \frac{\partial V^*}{\partial X}, u) = X^T L_a X + u^T R u - \gamma V + \langle \frac{\partial V^*}{\partial X}, \dot{X} \rangle \tag{4.58}$$

Integrating (4.58) with respect to t over $[t-T, t]$ gives

$$\begin{aligned} &-\int_{t-T}^{t} (X^T(\tau)L_a X(\tau) + u(\tau)^T R u(\tau)) d\tau \\ &= \int_{t-T}^{t} [\langle \frac{\partial V^*}{\partial X}, \dot{X} \rangle - \mathcal{H}(\tau, X, \frac{\partial V^*}{\partial X}, u)] d\tau \end{aligned} \tag{4.59}$$

Using *Lemma 4.4* and the HJB equation (4.21), equation (4.59) becomes

$$
\begin{aligned}
&-\int_{t-T}^{t}(X^T(\tau)L_aX(\tau)+u(\tau)^TRu(\tau))d\tau \\
&=\int_{t-T}^{t}[\langle\frac{\partial V^*}{\partial X},\dot{X}\rangle-\mathcal{H}(\tau,X,\frac{\partial V^*}{\partial X},u^*) \\
&\qquad -(u-u^*)^{\mathsf{T}}R(u-u^*)]d\tau \\
&=\int_{t-T}^{t}[(\frac{\partial V^*}{\partial \tau}+\langle\frac{\partial V^*}{\partial X},\dot{X}\rangle)-(u-u^*)^{\mathsf{T}}R(u-u^*)]d\tau \\
&=\int_{t-T}^{t}\dot{V}^*d\tau-\int_{t-T}^{t}[(u-u^*)^{\mathsf{T}}R(u-u^*)d\tau \\
&=V^*(t)-V^*(t-T)-\int_{t-T}^{t}[(u-u^*)^{\mathsf{T}}R(u-u^*)d\tau, \\
&\qquad\qquad \{t,X(t)\}\in[t_0,\infty)\times\Omega
\end{aligned}
\tag{4.60}
$$

Following a similar procedure in the proof for *Lemma 4.3*, we can transform (4.60) into a discounted form

$$
\begin{aligned}
&V^*(t-T,X(t-T))=e^{-\gamma T}V^*(t,X(t)) \\
&\quad+\int_{t-T}^{t}e^{-\gamma(\tau-t+T)}[(X^T(\tau)L_aX(\tau)+u(\tau)^TRu(\tau))d\tau \\
&\quad-\int_{t-T}^{t}(u-u^*)^{\mathsf{T}}R(u-u^*)d\tau \\
&\qquad\qquad \{t,X(t)\}\in[t_0,\infty)\times\Omega
\end{aligned}
\tag{4.61}
$$

Using the definition (4.54) and *Lemma 4.4*, we evaluate $Q(t_0,X,u)$ for the time interval $[t-T,t]$, hence, $t_0=t-T$ and we can write

$$
\begin{aligned}
&Q(t-T,X(t-T),u(t-T))= \\
&\qquad e^{-\gamma[(t-T)-(t-T)]}V^*(t-T,X(t-T)) \\
&\quad+\frac{\partial V^*(t-T,X(t-T))}{\partial t}+\mathcal{H}(t-T,X(t-T),\frac{\partial V^*}{\partial X},u(t-T)) \\
&=V^*(t-T,X(t-T))+\frac{\partial V^*(t-T)}{\partial t}-\frac{\partial V^*(t-T)}{\partial t} \\
&\quad+(u(t-T)-u^*(t-T))^{\mathsf{T}}R(u(t-T)-u^*(t-T)) \\
&=V^*(t-T,X(t-T)) \\
&\quad+(u(t-T)-u^*(t-T))^{\mathsf{T}}R(u(t-T)-u^*(t-T))
\end{aligned}
\tag{4.62}
$$

and

$$\begin{aligned} Q(t,X(t),u(t)) &= e^{-\gamma[t-(t-T)]}V^*(t,X(t)) \\ &\quad + \frac{\partial V^*(t)}{\partial t} + \mathcal{H}(t,X(t),\frac{\partial V^*}{\partial X},u(t)) \\ &= e^{-\gamma T}V^*(t,X(t)) + (u(t)-u^*(t))^\top R(u(t)-u^*(t)) \end{aligned} \tag{4.63}$$

Then we can rewrite (4.61) as the Bellman equation (4.56) with a residual error ψ (4.57). This proves the first part.

Substituting the optimal control u^*, we have $\psi = 0$ and the Q-learning Bellman equation becomes

$$\begin{aligned} &Q^*(t-T,X(t-T),u^*(t-T)) = \\ &\int_{t-T}^{t} e^{-\gamma(\tau-t+T)}(X^T(\tau)L_aX(\tau)+u^*(\tau)^TRu^*(\tau))d\tau \\ &+ e^{-\gamma T}Q^*(t,X(t),u^*(t)), \\ &\qquad\qquad \{t,X(t)\} \in [t_0,\infty)\times\Omega \end{aligned} \tag{4.64}$$

Using *Lemma 4.5*, we can replace Q^* with V^* in (4.64). Thus, solving for Q^* in (4.56) is equivalent to finding V^* in (4.46). This completes the proof. □

4.5 On the Functional Continuity and Parameterisation

Although the Q-learning Bellman equation (4.56) is a linear differential equation, solving for $Q^*(t,X,u)$ is still difficult. Moreover, extracting $\partial V^*(t,X)/\partial X(t)$ from the solution $Q^*(t,X,u)$ for the optimal control law is also a demanding task. Inspired by the neural network approximation for the value function in [164][149], it is reasonable to use neural networks to approximate a Q-functional $Q(t,X,u)$ that is smooth on some prescribed compact set, namely, Ω. This also provides possible ways to extract useful information for control laws based on the networks.

First, the assumption on smoothness of the Q-functional $Q(t,X,u)$ is desired for its approximation in the Sobolev norm, i.e., the approximation of the value function and

its gradient. It is necessary to place the solution of the Q-learning Bellman equation in a Hilbert space [186], or equivalently, in a certain Sobolev space [164][149][151]. We use the Sobolev space to define functions that are $\mathcal{L}^2(\Omega)$ with their partial derivative.

Definition 4.5 *(Sobolev space) Over the compact set Ω, the functional space*

$$\mathcal{W}^{m,p}(\Omega) := \{V \in \mathcal{L}^p(\Omega) : D^\alpha V \in \mathcal{L}^p(\Omega), 0 \leq |\alpha| \leq m\} \tag{4.65}$$

is called the Sobolev space, where m is a nonnegative integer, $1 \leq p \leq \infty$, and $D^\alpha V$ denotes the α-order weak (or distributional) partial derivative of a function V. ⋄

The Sobolev space $\mathcal{W}^{m,p}(\Omega)$ consists of functions in $\mathcal{L}^p(\Omega)$ that have weak partial derivatives up to order m and they belong to $\mathcal{L}^p(\Omega)$. For $p = 2$, the Sobolev space is a Hilbert space [193].

Assumption 4.4 *Given a state $X(t) \in \Omega$, a stabilising control $\bar{u} \in \bar{U}$ and an admissible control $u \in U(\Omega)$, there exists a compact set $\Omega_Q \subset \Omega \times \mathbb{R}^q \times \bar{U} \times U(\Omega)$ such that the associated tuple $(X, \bar{y}, \bar{u}, u) \in \Omega_Q$. Assume that the compact set Ω_Q is sufficiently regular so that $\mathcal{C}_0^\infty(\mathbb{R}^{n+2q+2m})$ is dense in $\mathcal{W}^{m,p}(\Omega_Q)$. The value functional $V(t,X)$ and the Q-functional $Q(t,X,u)$ are continuous and differentiable. Therefore, $V(t,X) \in \mathcal{W}^{1,2}(\Omega)$ and $Q(t,X,u) \in \mathcal{W}^{1,2}(\Omega_Q)$.* ⋄

In a certain sense, a function that belongs to the Sobolev space has a weaker notion of smoothness compared to differentiability, which can be used in a wider practical applications. We have essentially assumed that $Q \in \mathcal{C}^1(\Omega_Q)$ for the sake of simplicity. However, it should be noted that this can be relaxed to $Q(t,X,u) \in \mathcal{W}^{1,2}(\Omega_Q)$ at best when the optimal control problem does not have a smooth or continuous solution $V(t,X)$.

These assumptions allow the use of neural network approximation for the Q-functional. We consider a class of single hidden-layer feedforward neural network, of which the

output functions belong to the set

$$\begin{aligned}\Sigma(\Phi) =&\{Q_L(X,\bar{y},\bar{u},u):\Omega_Q\to\mathbb{R}\mid Q_L(X,\bar{y},\bar{u},u)=\\ &\sum_{j=1}^{N} w_j\varphi_j(X,\bar{y},\bar{u},u)=W^T\Phi(X,\bar{y},\bar{u},u)\}\end{aligned}\tag{4.66}$$

where $Q_L(X,\bar{y},\bar{u},u)$ is the neural network output, w_j is the neural network weights, $\varphi_j(X,\bar{y},\bar{u},u)$ is the activation functions, N is the number of hidden-layer neurons, $W := [w_1\ w_2\ \dots\ w_L]^T$ is the weight vector, and $\Phi := [\varphi_1\ \varphi_2\ \dots\ \varphi_L]^T$ is the activation function vector.

The following lemma is an extension from [194, Corollary 3.8]:

Lemma 4.7 *Given some nonnegative integer $l \geq m$, if the activation function $\Phi(X,\bar{y},\bar{u},u)$ is l-definite, i.e., $\varphi(X,\bar{y},\bar{u},u) \in \mathcal{C}^l(\Omega_Q)$ and $0 < \int_{\Omega_Q} |D^l\varphi| d\lambda < \infty$ in which the integral is defined in the sense of Lebesgue with a measure λ, then $\Sigma(\Phi)$ is dense in $\mathcal{W}^{m,p}(\Omega_Q)$.* ⋄

Proof. Refer to [194] for the detailed proof. □

This shows that an element in $\mathcal{W}^{m,p}(\Omega_Q)$ can always be approximated by functions smooth on Ω_Q. To be specific, the activation functions $\{\varphi_j : j = 1,2,\dots,N\}$ are selected such that the set $\{\varphi_j : j = 1,2,\dots,\infty\}$ is complete and linearly independent in order to achieve uniform approximation [186][164].

Expanding (4.53) using the GHJB equation (4.27), the Q-functional can be parame-

terised such that

$$
\begin{aligned}
Q(t,X,u) &= (1-\gamma)V^* + \frac{\partial V^*}{\partial t} + X^T L_a X + u^T R u + \langle \frac{\partial V^*}{\partial X}, \dot{X} \rangle \\
&= (Cx - y_f)^T L_a (Cx - y_f) + (\frac{\partial V^*}{\partial x})^T (Ax + f(\bar{y},\bar{u})) \\
&\quad \underbrace{+(\frac{\partial V^*}{\partial y_f})^T (\frac{\bar{y} - y_f}{k}) + (1-\gamma)V^* + \frac{\partial V^*}{\partial t}}_{Q_{xx}(t,X,\bar{y},\bar{u})} \\
&\quad + \underbrace{(\frac{\partial V^*}{\partial x})^T B u}_{Q_{xu}(t,X,u)} + \underbrace{u^T R u}_{Q_{uu}(u)} \\
&= Q_{xx}(t,X,\bar{y},\bar{u}) + Q_{xu}(t,X,u) + Q_{uu}(u)
\end{aligned}
\tag{4.67}
$$

where $Q_{xx}(t,X,\bar{y},\bar{u})$, $Q_{xu}(t,X,u)$, and $Q_{uu}(u)$ are lumped functions. Such parameterisation allows the approximation via neural networks since X, $\bar{y}$, $\bar{u}$, and u are all known signals.

Using the parameterisation (4.67), we write the neural network $Q_L(t,X,u)$ according

to the three components $Q_{xx}(X,\bar{y},\bar{u})$, $Q_{xu}(t,X,u)$, and $Q_{uu}(u)$ as

$$
\begin{aligned}
Q_L(t,X,u) &= W^{\mathsf{T}}\Phi(t,X,\bar{y},\bar{u},u) \\
&= \begin{bmatrix} W_{xx}^{\mathsf{T}} & W_{xu}^{\mathsf{T}} & W_{uu}^{\mathsf{T}} \end{bmatrix}
\begin{bmatrix} \Phi_{xx}(t,X,\bar{y},\bar{u}) \\ ------- \\ vec(\Phi_{xu}(t,X)\otimes u) \\ ------- \\ \Phi_{uu}(u) \end{bmatrix} \\
&= \begin{bmatrix} W_{xx} \\ --- \\ W_{xu1} \\ W_{xu2} \\ \vdots \\ W_{xum} \\ --- \\ W_{uu1} \\ W_{uu2} \\ \vdots \\ W_{uum} \end{bmatrix}^{\mathsf{T}}
\begin{bmatrix} \Phi_{xx}(t,X,\bar{y},\bar{u}) \\ ---- \\ \Phi_{xu}(t,X)u_1 \\ \Phi_{xu}(t,X)u_2 \\ \vdots \\ \Phi_{xu}(t,X)u_m \\ ---- \\ u_1^2 \\ u_2^2 \\ \vdots \\ u_m^2 \end{bmatrix}
\end{aligned}
\tag{4.68}
$$

where $\otimes$ denotes the Kronecker product and $vec(\cdot)$ is the vectorisation function which stacks the columns of a matrix together.

Denote the residual error $\varepsilon(t) = Q(t,X,u) - Q_L(t,X,u)$ due to the neural network approximation. Then using the neural network description for the Q-functional, the Q-learning Bellman equation (4.56) can be written as

$$
\begin{aligned}
&\underbrace{W^T[e^{-\gamma T}\Phi(t)-\Phi(t-T)]}_{W^T\Delta\Phi} + \underbrace{e^{-\gamma T}\varepsilon(t)-\varepsilon(t-T)+\psi}_{\varepsilon_B} = \\
&\underbrace{-\int_{t-T}^{t} e^{-\gamma(\tau-t+T)}(X^T(\tau)L_aX(\tau)+u^T(\tau)Ru(\tau))d\tau}_{-\rho(X,u)}
\end{aligned}
\tag{4.69}
$$

where $\varepsilon_B = \Delta\varepsilon + \psi$ with $\Delta\varepsilon = e^{-\gamma T}\varepsilon(t) - \varepsilon(t-T)$ is equivalent to the temporal difference (TD) error (sometimes called the Bellman error) in reinforcement learning [159][195] and $\rho(X,u)$ is the integral reinforcement over the interval $[t-T,\ t]$.

We reconstruct the optimal control u^* from (4.29) based on the parameterisation of $Q(x,u)$ (4.68) such that

$$u^* = u_L + \varepsilon_u \tag{4.70}$$

with the actor neural network

$$u_L = -\frac{1}{2} diag(W_{uu})^{-1} W_{xu}^T \Phi_{xu}(X) \tag{4.71}$$

where ε_u is a bounded approximation error due to ε, $W_{xu}^{*\top}\Phi_{xu}(x)$ accounts for the term $g(x)^\top \nabla V_x^*$, and $diag(W_{uu}^*)$ is essentially the pre-defined R. However, we keep the notation $diag(W_{uu}^*)$ for the sake of consistency.

Algorithm 3 (Policy iteration via the Q-learning Bellman equation)

1. (Policy Evaluation) Given the admissible policy (control) $u^{(i)}$, solve for $W^{(i)}$ using the Q-learning Bellman equation

$$W^{(i)T}\Delta\Phi(t-T, X(t-T)) + \psi = -\rho(X, u^{(i)}) \tag{4.72}$$

2. (Policy Improvement) Update the policy (control) using

$$u^{(i+1)} = -\frac{1}{2} diag(W_{uu}^{(i)})^{-1} W_{xu}^{(i)T} \Phi_{xu}(X) \tag{4.73}$$

The convergence is given by the following theorem.

Theorem 4.3 *Over the compact set Ω_Q, approximate solutions to the Q-learning Bellman equation* (4.69) *using the method of least squares exist and are unique for each N. In addition, $\exists i_0, i \in \mathbb{Z}^+$, $\forall i \geq i_0$, we have*

- $\sup_\Omega |\varepsilon| \to 0$ *and* $\sup_\Omega |\varepsilon_u| \to 0$ *as* $N \to \infty$,

- $\sup_{\Omega_Q} |W^{(i)\top}\Phi - Q^*| \to 0$ *as* $N \to \infty$,
- $\sup_{\Omega_Q} ||u^{(i+1)} - u^*|| \to 0$ *as* $N \to \infty$,
- $\exists N_0 \in \mathbb{Z}^+$ *such that* $u^{(i+1)} \in U(\Omega_Q)$ *for* $N \geq N_0$.

Proof. For proof by induction, see theorem 4.2 and 4.3 in [164]. If $u^{(0)} \in U(\Omega)$, then $u^{(i)} \in U(\Omega)$, $\forall i$. Moreover, $Q^{(i)} \to V^*$, $u^{(i)} \to u^*$ uniformly on Ω_Q.

The proof follows from *Theorem 2* and *Lemma 4.6*.

□

Remark 4.7 *We address the approximation of functions in the Sobolev space. Hence, a Q-functional* $Q(t,X,u)$*, as long as it is* $\mathcal{W}^{m,p}(\Omega_Q)$*, can be approximated by some smooth functions on* Ω_Q*. However, if we have a stronger condition on* $Q(t,X,u)$ *such as* $Q(t,X,u) \in \mathcal{C}^m(\Omega_Q)$*, Lemma 24 reduces to the high-order Stone-Weierstrass appoximation theorem, i.e., there exists a polynomial* $P \in \Sigma(\Phi)$ *such that it converges uniformly to* $Q(t,X,u)$ *and such that all its partial derivatives up to order* m *converge uniformly. In this case,* P *is* m*-uniformly dense and one can use neural networks based on power series (polynomials* P*) since they hold the useful property that they are termwise differentiable.* ◇

Remark 4.8 *Solving the Q-learning Bellman equation* (4.72) *in policy iteration is not a trivial task given that* ψ *is not available. We will design an adaptive critic that can online learn the neural network weights and approximate the optimal Q-functional without knowing neither the system dynamics* $\dot{X}$ *nor* ψ *by using a novel adaptive technique.* ◇

4.6 Adaptive Critic: Online Tuning and Convergence

Denote the ideal neural network weights W^* that provide the best approximate solution $Q^* = W^{*\top}\Phi(X,\bar{y},\bar{u},u) + \varepsilon$ for Q_L with ε being the residual error due to the approximation. Since the ideal weights W^* are unknown, we write the adaptive critic neural

network as

$$\hat{Q}(t,X,u) = \hat{W}^{\mathsf{T}}\Phi(t,X,\bar{y},\bar{u},u) \tag{4.74}$$

with $\hat{W}$ being the estimate of the ideal weights W^*.

The standard policy iteration algorithm proceeds by alternately updating the critic Q-functional and the actor policy (control). However, as a continuous-time system model is considered, one desires to update the policy in a continuous-time manner instead of discrete-time iteration. For example, inspired by the adaptive control principle, one can design the adaptation law using a differential expression $\dot{\hat{W}}$ instead of a difference $\hat{W}^{(i+1)}$. We propose the adaptation scheme that simultaneously updates the critic and actor to guarantee the convergence as well as the stability.

We design an auxiliary vector $M(t)$ being the adaptation driver as

$$M(t) = \mathcal{P}(t)\hat{W}(t) + \mathcal{Q}(t) \tag{4.75}$$

where the *information matrix* $\mathcal{P}(t) \in \mathbb{R}^{N\times N}$ and the *reinforcement vector* $\mathcal{Q}(t) \in \mathbb{R}^N$ are defined as

$$\dot{\mathcal{P}}(t) = -\ell\mathcal{P}(t) + \Delta\Phi\Delta\Phi^{\mathsf{T}}, \qquad \mathcal{P}(t_0) = 0 \tag{4.76a}$$

$$\dot{\mathcal{Q}}(t) = -\ell\mathcal{Q}(t) + \Delta\Phi\rho, \qquad \mathcal{Q}(t_0) = 0 \tag{4.76b}$$

with ℓ being a positive constant. Note that the information matrix $\mathcal{P}(t)$ acts as a regressor while the vector $\mathcal{Q}(t)$ is subject to the integral reinforcement ρ. We will show that the adaptation driver $M(t)$ contains the explicit information of the weight estimation error $\tilde{W}(t)$ which can be employed in the adaptation law to guarantee the weight convergence.

Lemma 4.8 *The adaptation driver $M(t)$ satisfies*

$$M(t) = -\mathcal{P}(t)\tilde{W}(t) + \psi_\varepsilon(t) \tag{4.77}$$

where $\tilde{W}(t) = W^* - \hat{W}(t)$ *is the weight estimation error and* $\psi_\varepsilon(t)$ *is the residual error due to* ε_B*, i.e.,*

$$\psi_\varepsilon(t) = -\int_{t_0}^{t} e^{-\ell(t-\tau)}\Delta\Phi(\tau)\varepsilon_B(\tau)d\tau,\ \psi_\varepsilon(t_0) = 0 \tag{4.78}$$

Moreover, $\psi_\varepsilon(t)$ *is a bounded variable for bounded state* $X(t)$ *and control* $u(t)$*.* ◇

Proof. We compute the solution of (4.76) to obtain

$$\mathcal{P}(t) = \int_{t_0}^{t} e^{-\ell(t-\tau)}\Delta\Phi\Delta\Phi^\mathsf{T}d\tau, \qquad \mathcal{P}(t_0) = 0 \tag{4.79a}$$

$$\mathcal{Q}(t) = \int_{t_0}^{t} e^{-\ell(t-\tau)}\Delta\Phi\rho d\tau, \qquad \mathcal{Q}(t_0) = 0 \tag{4.79b}$$

Considering the weight estimation error

$$\tilde{W}(t) = W^* - \hat{W}(t) \tag{4.80}$$

for the Q-learning Bellman equation in (4.69), the approximate Q-learning Bellman equation is

$$\tilde{W}^{*\mathsf{T}}\Delta\Phi(X, \bar{y}, \bar{u}, u) + \varepsilon_B = -\rho(X, u) \tag{4.81}$$

Combining (4.81) and (4.79) for $M(t)$ (4.75) yields (4.77). $\psi_\varepsilon(t)$ is an integral error due to $\varepsilon_B = \Delta\varepsilon + \psi$, where ε is the bounded neural network approximation error and ψ as shown in (4.57) is also bounded for bounded $X, \bar{y}, \bar{u}$, and u. □

Hence, an adaptation law driven by $M(t)$ can be written as

$$\dot{\hat{W}}(t) = -\Gamma M(t) \tag{4.82}$$

with $\Gamma > 0$ being a positive learning gain.

The PE condition is widely required in adaptive control to guarantee parameter convergence. Here we present its formal definition and use it for the analysis of the neural network weight convergence.

Definition 4.6 (*PE condition*) *The signal $\Delta\Phi(t)$ is said to be persistently excited (PE) if there exist $\mathcal{T} > 0$ and $\sigma_1 > 0$ such that*

$$\int_t^{t+\mathcal{T}} \Delta\Phi(\tau)\Delta\Phi(\tau)^\top \mathrm{d}\tau \geq \sigma_1 I, \quad \forall t \in [t_0, \infty) \tag{4.83}$$

◇

Lemma 4.9 *If the signal $\Delta\Phi(t)$ is persistently excited for $\forall t > 0$, the auxiliary variable $\mathcal{P}$ subject to* (4.76) *is positive definite, i.e. $\mathcal{P} \succ 0$ and its minimum eigenvalue $\lambda_{min}(\mathcal{P}) > \sigma_1 > 0$, $\forall t > 0$ for some positive constant σ_1.* ◇

Proof. The detailed proof follows from [75]. □

4.7 Main Results

We have shown the convergence of the proposed adaptive critic by proving that the neural network weight estimation error $\tilde{W}$ is uniformly ultimately bounded given bounded state X and control u.

Following (4.70)(4.71), we leverage the parameterisation of the Q-function to determine the actor that generates the control or policy. The approximate optimal control obtained through the adaptive critic neural network can be written as

$$u = -\frac{1}{2} diag(\hat{W}_{uu})^{-1} \hat{W}_{xu}^\top \Phi_{xu}(X) \tag{4.84}$$

or, more simply,

$$u = -\frac{1}{2} R^{-1} \hat{W}_{xu}^\top \Phi_{xu}(X) \tag{4.85}$$

since the ideal value of $diag(\hat{W}_{uu})$ is *a priori* known as R. Now we present the main results of this chapter that show the overall convergence by proving the overall stability of the complete adaptive optimal observer. In summary, considering a nonlinear system (4.1) and its optimal observer in the form of (4.3) associated with the value functional (4.26), we have the following theorem:

Theorem 4.4 *Provided that $\Delta\Phi(t)$ is PE for $\forall t > 0$, the adaptive critic* (4.74)*, the actor* (4.85)*, and the adaptation law* (4.82) *along with* (4.75)(4.76) *form an adaptive optimal control so that*

a) the state $\boldsymbol{X}(t)$ and the weight estimation error $\tilde{W}(t)$ are uniformly ultimately bounded in a semi-global sense;

b) the control u will ultimately enter and stay in a small bounded region around the optimal control u^*;

c) if there is no neural network approximation error, i.e., $\varepsilon = 0$, the weight estimation error $\tilde{W}(t)$ will exponentially converge to zero and the control u will converge exponentially to the optimal control u^*.

Proof. We construct an overall Lyapunov function that takes into account the weight estimation error $\tilde{W}$, the optimal value functional V^* of X and u, and the effect of the residual error ψ_ε (4.78). Note that $\psi_\varepsilon(t)$ is the residual approximation error due to ε_B, i.e., $\psi_\varepsilon(t) = -\int_{t_0}^{t} e^{-\ell(t-\tau)}\Delta\Phi(\tau)\varepsilon_B(\tau)d\tau,\ \psi_\varepsilon(t_0) = 0$, where $\varepsilon_B = \Delta\varepsilon + \psi$. For simplicity, we split the term into two parts written as

$$\psi_\varepsilon(t) = \psi_1 + \psi_2 \tag{4.86}$$

with

$$\psi_1(t) = -\int_{t_0}^{t} e^{-\ell(t-\tau)}\Delta\Phi(\tau)\Delta\varepsilon(\tau)d\tau,\ \psi_1(t_0) = 0 \tag{4.87}$$

$$\psi_2(t) = -\int_{t_0}^{t} e^{-\ell(t-\tau)}\Delta\Phi(\tau)\psi(\tau)d\tau,\ \psi_2(t_0) = 0 \tag{4.88}$$

The first term ψ_1 characterises the effect of the neural network appoximation error ε. If there is no approximation error, i.e., $\varepsilon = 0$, then $\Delta\varepsilon = 0$ and $\psi_1 = 0$. The second term ψ_2 is due to the residual error ψ (4.57) that appeared in the Q-learning Bellman equation (4.56). Because ψ is a function of $u(t)$, $u^*(t)$ and time-delayed variables $u(t -$

$T)$, $u^*(t-T)$, we express it as

$$\psi = \psi_a + \psi_b + \psi_c \tag{4.89a}$$

$$\psi_a = -\int_{t-T}^{t} (u(\tau) - u^*(\tau))^\mathsf{T} R(u(\tau) - u^*(\tau)) d\tau \tag{4.89b}$$

$$\psi_b = -(u(t) - u^*(t))^\mathsf{T} R(u(t) - u^*(t)) \tag{4.89c}$$

$$\psi_c = (u(t-T) - u^*(t-T))^\mathsf{T} R(u(t-T) - u^*(t-T)) \tag{4.89d}$$

Using the idea of delay dependent stability [196], we design a Lyapunov function candidate $\mathscr{L}$ as

$$\mathscr{L} = \mathcal{L}_1 + k_2\mathcal{L}_2 + k_3\mathcal{L}_3 + k_4\mathcal{L}_4 + k_5\mathcal{L}_5 + k_6\mathcal{L}_6 \tag{4.90}$$

where sub-Lyapunov functions $\mathcal{L}_1 = \frac{1}{2}\tilde{W}^\mathsf{T}\Gamma^{-1}\tilde{W}$, $\mathcal{L}_2 = Q^*(t, X, u^*)$, $\mathcal{L}_3 = \frac{1}{2}\psi_1^\mathsf{T}\psi_1$, $\mathcal{L}_4 = \frac{1}{2}\psi_2^\mathsf{T}\psi_2$, $\mathcal{L}_5 = \int_{-T}^{0} \tilde{W}^\mathsf{T}(t+\tau)\tilde{W}(t+\tau)d\tau$; and $\mathcal{L}_6 = \int_{T}^{0}\int_{t+\theta}^{t} \tilde{W}^\mathsf{T}(\tau)\tilde{W}(\tau)d\tau d\theta$, and k_2, k_3, k_4, k_5, and k_6 are some positive constants. The stability analysis is carried out in a semi-global sense, i.e., the overall Lyapunov function $:\mathscr{L}$ is calculated over a sufficiently large but fixed compact set $\tilde{\Omega} \subset \mathbb{R}^N \times \Omega \times \mathbb{R}^N \times \mathbb{R}^N$ in the tuple $(\tilde{W}, X, \psi_1, \psi_2)$ that contains the origin.

We analyse the derivative of each term in $\mathscr{L}$ (4.90). Using (4.82) and *Lemma 4.8* and *Lemma 4.9*, we calculate for the first term $\dot{\mathcal{L}}_1$ using Young's inequality with η: $||a||||b|| \leq \frac{1}{2\eta}||a||^2 + \frac{\eta}{2}||b||^2$ (valid for every $\eta > 0$) as

$$\begin{aligned}
\dot{\mathcal{L}}_1 &= \tilde{W}^\mathsf{T}\Gamma^{-1}\dot{\tilde{W}} = \tilde{W}^\mathsf{T} M = -\tilde{W}^\mathsf{T}\mathcal{P}\tilde{W} + \tilde{W}^\mathsf{T}\psi_\varepsilon \\
&\leq -\sigma_1||\tilde{W}|| + ||\tilde{W}||\,||\psi_1 + \psi_2|| \\
&\leq -(\sigma_1 - \frac{1}{2\eta_1} - \frac{1}{2\eta_2})||\tilde{W}||^2 + \frac{\eta_1}{2}||\psi_1||^2 + \frac{\eta_2}{2}||\psi_2||^2
\end{aligned} \tag{4.91}$$

where $\eta_1 > 0$, $\eta_2 > 0$ are properly chosen constants such that $\sigma_1 - \frac{1}{2\eta_1} - \frac{1}{2\eta_2} > 0$ holds.

For the second term, using the inequality $2||a||||b|| \leq ||a||^2 + ||b||^2$, we have

$$\begin{aligned}
\dot{\mathcal{L}}_2 &= \dot{Q}^*(t,X,u^*) = \dot{V}^*(t,X) = \frac{\partial V^*}{\partial t} + \langle \frac{\partial V^*}{\partial X}, \dot{X} \rangle \\
&= -X^\top L_a X - u^{*\top} R u^* \\
&\leq -\lambda_{\min}(L_a)||X||^2 - (\frac{1}{4}\lambda_{\min}(R^{-1}) - 1)||\Phi_{xu}||^2||W_{xu}^*||^2 \\
&\quad - (\lambda_{\min}(R) - 1)||\varepsilon_u||^2
\end{aligned} \tag{4.92}$$

with ε_u is a bounded approximation error for a bounded ε. Note that $\varepsilon = 0$ implies that $\varepsilon_u = 0$.

For the third term

$$\begin{aligned}
\dot{\mathcal{L}}_3 &= \psi_1^\top \dot{\psi}_1 = \psi_1^\top(-\ell\psi_1 + \Delta\Phi\Delta\varepsilon) \\
&\leq -(\ell - \frac{1}{2\eta_3})||\psi_1||^2 + \frac{\eta_3}{2}||\Delta\Phi\Delta\varepsilon||^2
\end{aligned} \tag{4.93}$$

where $\eta_3 > 0$ is a properly chosen constant such that $\eta_3 > -\frac{1}{2\ell}$ holds.

Similarly, for the fourth term

$$\begin{aligned}
\dot{\mathcal{L}}_4 &= \psi_2^\top \dot{\psi}_2 = \psi_2^\top(-\ell\psi_2 + \Delta\Phi(\psi_a + \psi_b + \psi_c)) \\
&\leq -(\ell - \frac{3}{2\eta_4})||\psi_2||^2 + \frac{\eta_4}{2}||\Delta\Phi\psi_a||^2 \\
&\quad + \frac{\eta_4}{2}||\Delta\Phi\psi_b||^2 + \frac{\eta_4}{2}||\Delta\Phi\psi_c||^2
\end{aligned} \tag{4.94}$$

The fifth term $\mathcal{L}_5 = \int_{-T}^{0} \tilde{W}^\top(t+\tau)\tilde{W}(t+\tau)d\tau = \int_{t-T}^{t} \tilde{W}^\top(\tau)\tilde{W}(\tau)d\tau$ so its derivative

$$\dot{\mathcal{L}}_5 = ||\tilde{W}(t)||^2 - ||\tilde{W}(t-T)||^2 \tag{4.95}$$

For the sixth term

$$\dot{\mathcal{L}}_6 = T||\tilde{W}(t)||^2 - \int_{t-T}^{t} ||\tilde{W}(\tau)||^2 d\tau \tag{4.96}$$

Combining the above, the derivative $:\dot{\mathscr{L}}$ can be written as

$$
\begin{aligned}
&\dot{\mathscr{L}} = \dot{\mathcal{L}}_1 + k_2\dot{\mathcal{L}}_2 + k_3\dot{\mathcal{L}}_3 + k_4\dot{\mathcal{L}}_4 + k_5\dot{\mathcal{L}}_5 + k_6\dot{\mathcal{L}}_6 \leq \\
&-(\sigma_1 - \frac{1}{2\eta_1} - \frac{1}{2\eta_2} - k_5 - k_6 T)||\tilde{W}||^2 \\
&-(k_3\ell - \frac{k_3}{2\eta_3} - \frac{\eta_1}{2})||\psi_1||^2 - (k_4\ell - \frac{3k_4}{2\eta_4} - \frac{\eta_2}{2})||\psi_2||^2 \\
&-k_2\lambda_{\min}(L_a)||X||^2 - k_2(\frac{1}{4}\lambda_{\min}(R^{-1}) - 1)||\Phi_{xu}||^2||W_{xu}^*||^2 \\
&-k_5||\tilde{W}(t-T)||^2 - k_6\int_{t-T}^{t}||\tilde{W}(\tau)||^2 d\tau \\
&-k_2(\lambda_{\min}(R) - 1)||\varepsilon_u||^2 + \frac{k_3\eta_3}{2}||\Delta\Phi\Delta\varepsilon||^2 \\
&+\frac{k_4\eta_4}{2}||\Delta\Phi\psi_a||^2 + \frac{k_4\eta_4}{2}||\Delta\Phi\psi_b||^2 + \frac{k_4\eta_4}{2}||\Delta\Phi\psi_c||^2
\end{aligned}
\tag{4.97}
$$

As we analyse the semi-global stability over $\tilde{\Omega}$, any initial value of the tuple $(\tilde{W}, X, \psi_1, \psi_2)$ is assumed to be within the interior of $\tilde{\Omega}$. Thus, for any initial trajectory, the state $||X||$ remains bounded and subsequently $||\Phi_{xu}||$ and $||\Delta\Phi||$ remain bounded for at least finite time. This implies that there exist positive constants $\xi_1 > 0,\ \xi_2 > 0$ so that

$$||\Phi_{xu}||^2 \leq \xi_1,\ ||\Delta\Phi||^2 \leq \xi_2$$

for at least finite time.

By inspection of (4.89) and from (4.70)(4.85), we can derive the following results for $||\psi_a||$, $||\psi_b||$, and $||\psi_c||$ using the inequality $(||a|| + ||b||)^2 \leq 2||a||^2 + 2||b||^2$:

$$
\begin{aligned}
||\psi_a|| &\leq \int_{t-T}^{t}[\frac{1}{2}\lambda_{\max}(R^{-1})||\tilde{W}_{xu}^{\mathsf{T}}\Phi_{xu}\Phi_{xu}^{\mathsf{T}}\tilde{W}_{xu}|| \\
&\qquad + 2\lambda_{\max}(R)||\varepsilon_u||^2]d\tau \\
&\leq \frac{\xi_1}{2}\lambda_{\max}(R^{-1})\int_{t-T}^{t}||\tilde{W}||^2 d\tau + 2\lambda_{\max}(R)\int_{t-T}^{t}||\varepsilon_u||^2 d\tau
\end{aligned}
\tag{4.98}
$$

$$
\begin{aligned}
||\psi_b|| &\leq \frac{1}{2}\lambda_{\max}(R^{-1})||\tilde{W}_{xu}^{\mathsf{T}}\Phi_{xu}(t)\Phi_{xu}^{\mathsf{T}}\tilde{W}_{xu}(t)|| \\
&\qquad + 2\lambda_{\max}(R)||\varepsilon_u(t)||^2 \\
&\leq \frac{\xi_1}{2}\lambda_{\max}(R^{-1})||\tilde{W}(t)||^2 + 2\lambda_{\max}(R)||\varepsilon_u(t)||^2
\end{aligned}
\tag{4.99}
$$

$$
\begin{aligned}
||\psi_c|| &\leq \frac{1}{2}\lambda_{\max}(R^{-1})||\tilde{W}_{xu}^{\top}\Phi_{xu}(t-T)\Phi_{xu}^{\top}\tilde{W}_{xu}(t-T)|| \\
&\quad + 2\lambda_{\max}(R)||\varepsilon_u(t-T)||^2 \\
&\leq \frac{\xi_1}{2}\lambda_{\max}(R^{-1})||\tilde{W}(t-T)||^2 + 2\lambda_{\max}(R)||\varepsilon_u(t-T)||^2
\end{aligned} \tag{4.100}
$$

where $\tilde{W}_{xu} = W_{xu}^* - \hat{W}_{xu}$. We use again the inequality $(||a|| + ||b||)^2 \leq 2||a||^2 + 2||b||^2$ for $||\psi_a||^2$, $||\psi_b||^2$, and $||\psi_c||^2$ and then we have

$$
\begin{aligned}
\dot{\mathscr{L}} \leq & -(\sigma_1 - \frac{1}{2\eta_1} - \frac{1}{2\eta_2} - k_5 - k_6 T \\
& - \frac{k_4\eta_4\xi_1^2\xi_2\lambda_{\max}(R^{-1})^2}{4}||\tilde{W}(t)||^2)||\tilde{W}(t)||^2 \\
& - (k_3\ell - \frac{k_3}{2\eta_3} - \frac{\eta_1}{2})||\psi_1||^2 - (k_4\ell - \frac{3k_4}{2\eta_4} - \frac{\eta_2}{2})||\psi_2||^2 \\
& - k_2\lambda_{\min}(L_a)||X||^2 - k_2(\frac{1}{4}\lambda_{\min}(R^{-1}) - 1)||\Phi_{xu}||^2||W_{xu}^*||^2 \\
& - (k_5 - \frac{k_4\eta_4\xi_1^2\xi_2\lambda_{\max}(R^{-1})^2}{4}||\tilde{W}(t-T)||^2)||\tilde{W}(t-T)||^2 \\
& - (k_6 - \frac{k_4\eta_4\xi_1^2\xi_2\lambda_{\max}(R^{-1})^2}{4}\int_{t-T}^{t}||\tilde{W}(\tau)||^2 d\tau) \\
& \int_{t-T}^{t}||\tilde{W}(\tau)||^2 d\tau + \frac{k_3\eta_3}{2}||\Delta\Phi\Delta\varepsilon||^2 \\
& + [4k_4\eta_4\xi_2\lambda_{\max}(R)^2 + k_2(\lambda_{\min}(R) - 1)]||\varepsilon_u(t)||^2 \\
& + 4k_4\eta_4\xi_2\lambda_{\max}(R)^2(||\varepsilon_u(t-T)||^2 + \int_{t-T}^{t}||\varepsilon_u||^2 d\tau) \\
& \leq -\alpha_1||\tilde{W}(t)||^2 - \alpha_2||\psi_1||^2 - \alpha_3||\psi_2||^2 - \alpha_4||X||^2 \\
& - \alpha_5||\tilde{W}(t-T)||^2 - \alpha_6\int_{t-T}^{t}||\tilde{W}(\tau)||^2 d\tau + \beta_1 + \beta_2
\end{aligned} \tag{4.101}
$$

where $\alpha_i > 0$ ($i \in \mathbb{Z}^+$) are the positive coefficents with properly chosen k_i and η_i; $\beta_1 = \frac{k_3\eta_3}{2}||\Delta\Phi\Delta\varepsilon||^2$, $\beta_2 = [4k_4\eta_4\xi_2\lambda_{\max}(R)^2 + k_2(\lambda_{\min}(R) - 1)]||\varepsilon_u(t)||^2 + 4k_4\eta_4\xi_2\lambda_{\max}(R)^2(||\varepsilon_u(t-T)||^2 + \int_{t-T}^{t}||\varepsilon_u||^2 d\tau)$ are bounded constants that characterise the effect of the neural network approxiamtion error ε. The first four terms in the last inequality of (4.101) form a negative definite function in $\tilde{\Omega}$ so that a set of ultimate boundedness $\tilde{\Omega}'$ exists and it depends on the size of $(\beta_1 + \beta_2)$, i.e. a smaller value of $(\beta_1 + \beta_2)$ will decrease the size of $\tilde{\Omega}$. Assuming that N has been chosen large enough, it is possible to obtain ε along with $\Delta\varepsilon$, ε_u, $(\beta_1 + \beta_2)$ to be sufficiently small so that $\tilde{\Omega}' \subset \tilde{\Omega}$. Hence, it

is impossible for any trajectory to leave $\tilde{\Omega}$, i.e. it is an invariant set. The states $X(t)$ remain bounded and subsequently also $u(t)$, ε, Φ_{xu} and $\Delta\Phi$ are always bounded functions over a compact set. According to Lyapunov theorem [165], $X(t)$ and $\tilde{W}(t)$ are uniformly ultimately bounded. This proves a).

The difference of the actor to the optimal control

$$||u - u^*|| \leq \frac{1}{2}\lambda_{\max}(R^{-1})||\Phi_{xu}(x)||\,||\tilde{W}_{xu}|| + ||\varepsilon_u|| \tag{4.102}$$

remains bounded. This implies that u will stay close to u^*. This proves b).

If $\varepsilon = 0$, we have $\Delta\varepsilon = 0$, $\varepsilon_u = 0$ so that $\beta_1 + \beta_2 = 0$, then

$$\begin{aligned}\dot{\mathcal{L}}_4 \leq & -\alpha_1||\tilde{W}(t)||^2 - \alpha_2||\psi_1||^2 - \alpha_3||\psi_2||^2 - \alpha_4||X||^2 \\ & - \alpha_5||\tilde{W}(t-T)||^2 - \alpha_6\int_{t-T}^{t}||\tilde{W}(\tau)||^2 d\tau \leq 0\end{aligned} \tag{4.103}$$

According to Lyapunov theorem [165], $X(t)$ and $\tilde{W}(t)$ as well as $||u - u^*||$ will exponentially converge to zero. This proves c). □

Corollary 4.2 *If state $X(t)$ and control $u(t)$ remain bounded, and $\Delta\Phi(t)$ is PE for $\forall t > 0$, the adaptive critic* (4.74) *and the adaptation law* (4.82) *along with* (4.75)(4.76) *guarantees for the weight estimation error $\tilde{W}(t)$ that $||\tilde{W}(t)||$ is uniformly ultimately bounded and will converge towards a small compact set around zero.* ◇

Proof. From *Lemma 4.8*, we know that the residual error $\psi_\varepsilon(t)$ is a bounded variable for bounded state $X(t)$ and control $u(t)$, e.g., there exists a positive constant $\bar{\psi}_\varepsilon > 0$ so that $||\psi_\varepsilon|| < c\bar{\psi}_\varepsilon$ holds with $0 < c < 1$. Consider a Lyapunov function candidate

$$\mathcal{L}_1 = \frac{1}{2}\tilde{W}^\top\Gamma^{-1}\tilde{W} \tag{4.104}$$

From the proof of *Theorem 4.4*, its time derivative can be calculated as

$$\begin{aligned}\dot{\mathcal{L}}_1 &= -\tilde{W}^\top\mathcal{P}\tilde{W} + \tilde{W}^\top\psi_\varepsilon \\ &\leq -\sigma_1||\tilde{W}||^2 + ||\psi_\varepsilon||\,||\tilde{W}|| \\ &\leq -(\sigma_1 - \eta_5/2)||\tilde{W}||^2 + c^2\bar{\psi}_\varepsilon^2/2\eta_5 \\ &\leq -\alpha_7\mathcal{L}_1 + \beta_3\end{aligned} \tag{4.105}$$

with $\alpha_7 = (2\sigma_1 - \eta_5)\lambda_{\max}(\Gamma^{-1})$ being a positive constant provided that η_5 is properly chosen such that $0 < \eta_5 < 2\sigma_1$; $\beta_3 = c^2\bar{\psi}_\varepsilon^2/2\eta_5$ denoting the upper bound of the residual error. Thus, the first term $-\alpha_7\mathcal{L}_1$ in the last inequality form a negative definite function.

From the extended Lyapunov theorem, the Lyapunov function $\mathcal{L}_1$ is uniformly ultimately bounded, i.e. $\mathcal{L}_1(t) \leq e^{-\alpha_7 t}\mathcal{L}(t_0) + \beta_3/\alpha_7$. This implies that the weight estimation error $\tilde{W}(t)$ will ultimately enter the compact set

$$\Omega_W := \{\tilde{W}(t) | \|\tilde{W}(t)\| \leq \sqrt{c^2\bar{\psi}_\varepsilon^2\lambda_{\max}(\Gamma^{-1})/\eta_5\lambda_{\min}(\Gamma^{-1})(2\sigma_1 - \eta_5)}\}$$

of which the size depends on the bound of the residual error ψ_ε and the excitation level σ_1. Clearly, higher excitation level σ_1 or smaller residual error ψ_ε, e.g. smaller neural network approximation error ε, will reduce the size of Ω_W. The convergence rate can be also improved by increasing the learning gain Γ. □

4.8 Adaptive Optimal Observer for the Van der Pol Oscillator: A Case Study

Among different nonlinear systems, their equilibrium point appears more often as the origin, i.e., when the state is zero. This creates difficulty for the learning process of an adaptive optimal controller/observer as the PE condition is not satisfied after the state goes to zero. Various works have tried to cope with it by injecting some small exploratory noise (e.g., sinusoids of varying frequencies) to the control input to excite the system. Alternatively, some nonlinear systems can display periodic oscillations without external excitation, e.g., limit cycles. Hence, a typical system showing the limit cycle phenomenon such as the Van der Pol oscillator becomes a great option to maintain the PE condition without artificially injecting noise.

4.8.1 Problem Formulation

In this section, we justify the effectiveness of the proposed adaptive optimal observer for a Van der Pol oscillator given by

$$\begin{cases} \begin{bmatrix} \dot{\bar{x}}_1 \\ \dot{\bar{x}}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} \bar{x}_1^2 \bar{x}_2, \\ \bar{y} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix} \end{cases} \tag{4.106}$$

The oscillator is self-excited without the need for an extra stablising control, i.e., $\bar{u} = 0$ in this case. This ensures that the state $(\bar{x}_1, \bar{x}_2)$ belongs to a compact invariant set in $\mathbb{R}^2$ which is the limit cycle of the oscillator. Fig. 4.2 presents the phase portrait (limit cycle) of the Van der Pol oscillator (4.106). We write $A = \begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix}$, $B = [0 \ -1]^\mathsf{T}$, and $C = [1 \ 0]$; it is easy to find that the pair (A, B) is controllable and the pair (A, C) is observable. Thus, a state observer for the oscillator can be written as

$$\begin{cases} \dot{x} = Ax + Bx_1^2 x_2 + Bu, \\ y = Cx \end{cases} \tag{4.107}$$

where $x = [x_1 \ x_2]^\mathsf{T}$ is the observer state and u is the correction input to minimise the following discounted infinite-horizon value function

$$V = \int_t^\infty e^{-\gamma(\tau - t)} (L(Cx - y_f)^2 + Ru^2) d\tau \tag{4.108}$$

where we use the filtered output $y_f = \frac{1}{ks+1}\bar{y}$ as equivalent to (4.4).

Considering the nonlinear observable Van der Pol oscillator (4.106) and its state observer given by (4.107), the optimal observer design problem is to find an admissible correction (control) input u that minimises the value functional (4.108).

4.8.2 Numerical Simulation

We implement the proposed adaptive optimal control, i.e. the adaptive critic (4.74), the actor (4.85), and the adaptation law (4.82) along with (4.75)(4.76), for the above Van der Pol oscillator observer design problem. The parameter setting for simulation is as follows. The filter time constant is chosen to be small, $k = 0.01$. For the value functional, the discount factor $\gamma = 0.1$, the output error weight $L = 1$, and the control input weight $R = 1$. The initial oscillator state $\bar{x}(0) = [\bar{x}_1(0)\ \bar{x}_2(0)]^\mathsf{T} = [1\ 1]^\mathsf{T}$. The reinforcement interval $T = 0.1s$. The initial observer state $x(0) = [x_1(0)\ x_2(0)]^\mathsf{T} = [0\ 2]^\mathsf{T}$.

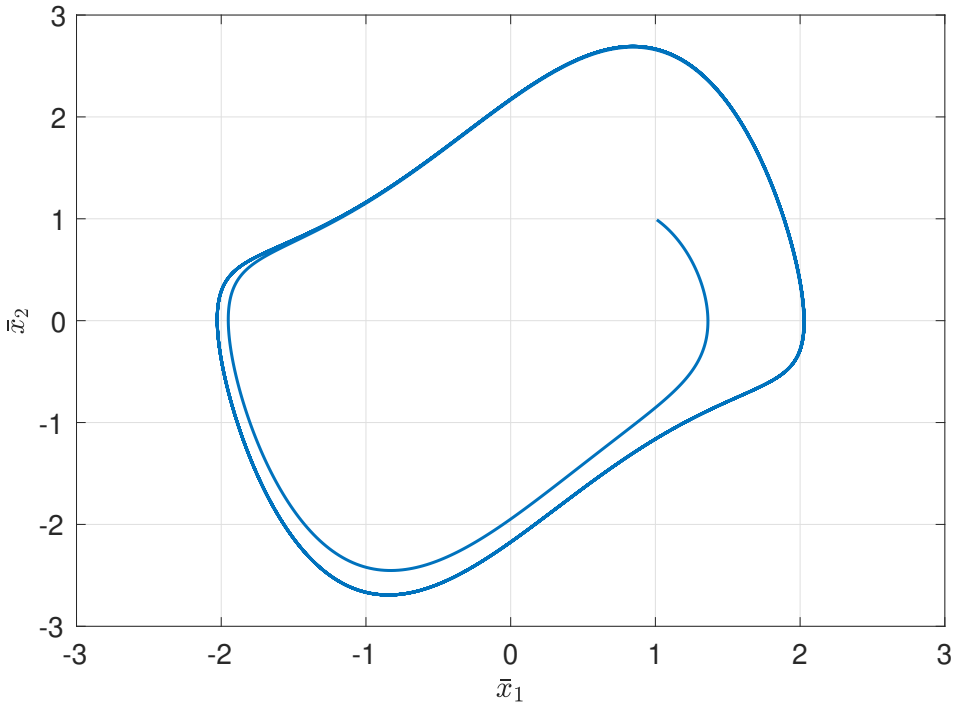

FIGURE 4.2. Phase portrait of the Van der Pol oscillator limit cycle.

The design of an adaptive critic neural network is often not a trivial task. One needs to be selective on choices of the activation function Φ, especially if any insight or pieces of information on the system dynamics are given. For this example of the Van der Pol oscillator, we know the plant system that we want to observe here is an oscillator. Even though we do not hold any knowledge of the coefficients of the oscillator itself,

we do know that the state response of such a system is periodic. The time-dependence of the functional V worsens the problem as it increases the complexity by introducing another dimension t. In fact, in our simulations we found that time-dependence of V makes a comparably small impact on the convergence but adds complexity when approximating the Q-function via a Q-learning Bellman equation. We will first show a result for the observer without considering time-dependence in the adaptive critic and then compare it with the results for a time-dependent critic. Hence, the critic neural network activation function vector is designed as $\Phi = [x_1^2,\ x_1x_2,\ x_2^2,\ x_1y_f,\ x_2y_f,\ y_f^2,\ x_1^2\bar{y},\ x_1x_2\bar{y},\ x_2^2\bar{y},\ x_1y_f\bar{y},\ x_2y_f\bar{y},\ y_f^2\bar{y},\ x_1^2u,\ x_1x_2u,\ x_2^2u,\ x_1y_fu,\ x_2y_fu, y_f^2u,\ u^2]^\mathsf{T}$. Accordingly, the critic weight vector without time-dependent elements can be written as $W = [W_1, W_2, ..., W_{19}]^\mathsf{T}$. For the adaptation law, $\ell = 1$ and $\Gamma = 0.7$. Since it is known that the actual value of $W_{19} = R$, we initialise all the weights to 0.5 except that $W_{19} = 1$ and its learning rate is set to zero. Fig. 4.3 shows the observer states against the desired oscillator state trajectories. Fig. 4.4 shows the convergence of the adaptive critic weights. These figures indicate that the learning process takes place over a period of roughly 90s. In fact, the adaptive critic weights finally converge towards $W = [0.7884, -0.0677, -0.0215, -1.1162,\ 0.07753,\ 0.3126,\ -0.0934,\ 0.2540, 0.0045,\ 0.0244,\ -0.2732, 0.0672, -0.1960, 0.2550, -0.0781, 0.5648, -0.0282, -0.3550, 1]^\mathsf{T}$.

We compare the final near-optimal control u that is found at the end of the learning process with a high-gain observer [197]. The high-gain observer has been implemented as

$$\dot{x}_h = Ax_h + Bx_{h1}^2 x_{h2} + BK(Cx_h - \bar{y}) \tag{4.109}$$

with a positive constant gain $K = [1/\epsilon \quad 1/\epsilon^2]^\mathsf{T}$. Reducing ϵ, i.e., a higher gain, often diminishes the steady state error but demands more control effort. Fig. 4.5 shows the comparison results of the adaptively-learned optimal observer against the high-gain observer with $\epsilon = 0.1$ and 1. The one with $\epsilon = 0.1$ tracks the state trajectories after significant transients of high-frequency oscillation due to the control u. On the other hand, the high-gain observer with $\epsilon = 1$ improves the transient performance and

greatly reduces the absolute value of control u. However, the tracking performance is not as good especially when $t = 5 \sim 10$ s. The optimal observer is optimal in the sense of minimising a pre-defined cost functional set by L and R. The output error weight L and control input weight R render a trade-off between the estimation error and the control effort. It can be seen from the state responses x_1, x_2 that the optimal observer tracks the Van der Pol oscillator better than the high-gain observer with $\epsilon = 1$ with faster transients. Meanwhile, the control input u shows that the optimal observer requires lower control effort than the one with $\epsilon = 0.1$.

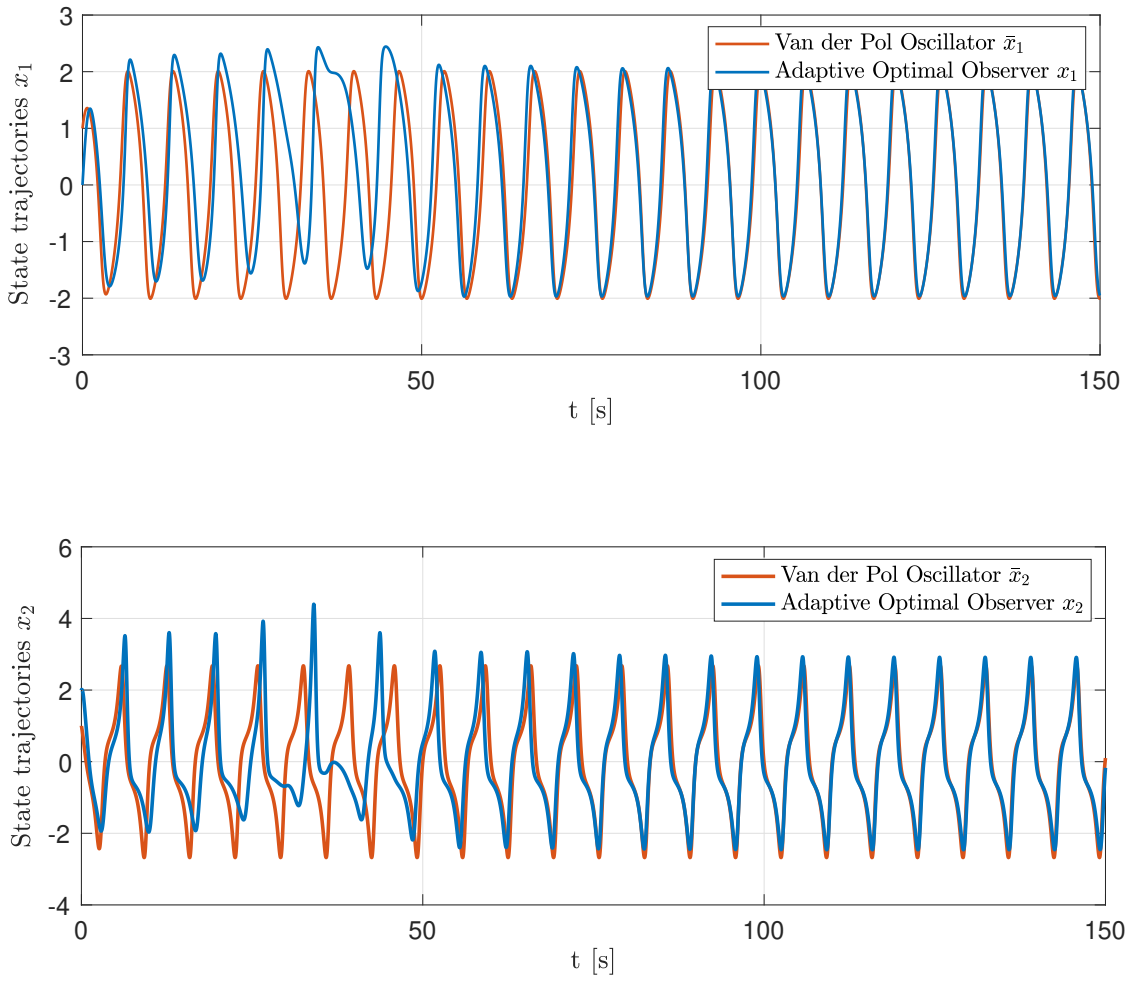

Figure 4.3: Adaptive optimal observer state estimation against the system trajectories (19 nodes).

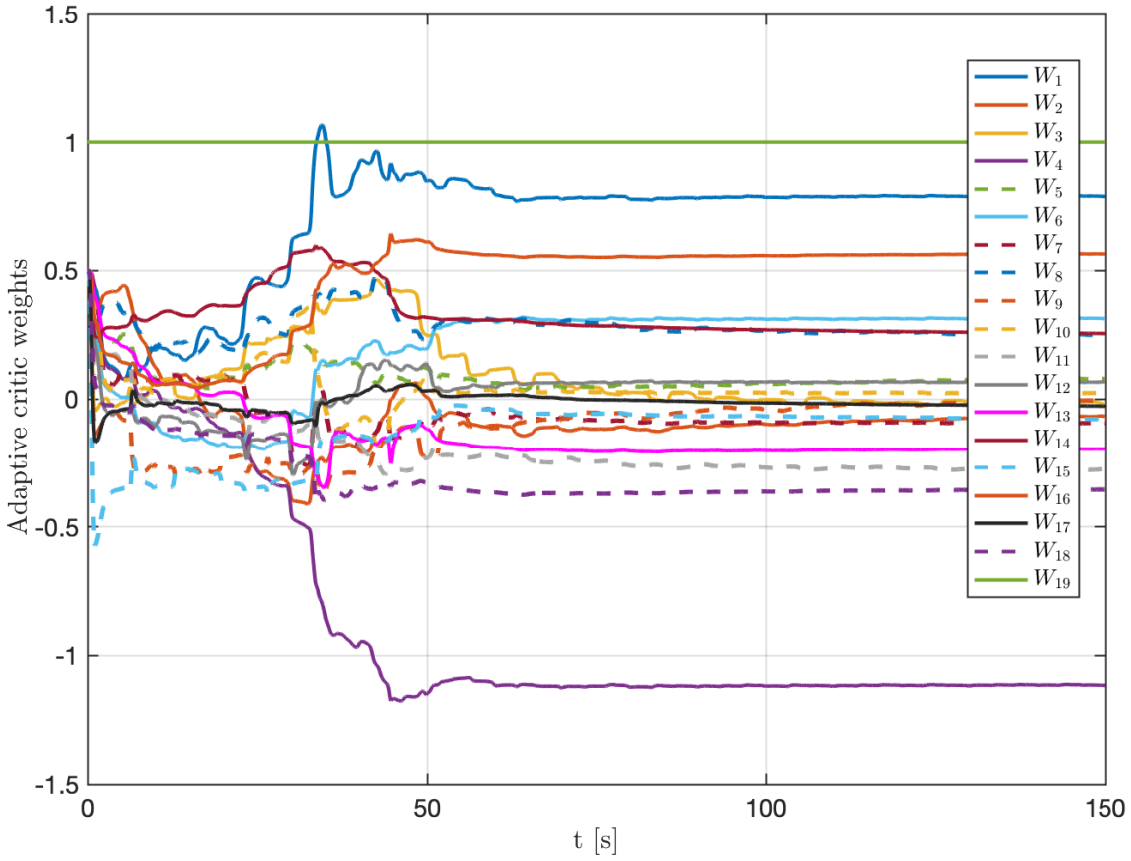

Figure 4.4: The convergence of the adaptive critic weights (19 nodes).

Now we investigate the effect of considering the time-dependence of functional V in the adaptive critic design. Given the periodicity of the Van der Pol oscillator, an intuitive way to compensate the polynomial-only activation function is by adding regressor basis functions that mimic the Fourier series. We know from the simulation that the period T_0 of the oscillator is $6.7s$ so the angular frequency $\omega = 2\pi/T_0 \approx 0.94$. Not every node of the neural network can contribute significantly to the overall approximation. We can start adding new extra nodes $W_{20}x_1y_f\cos(\omega t)$ and $W_{21}x_1y_f\sin(\omega t)$ based on the primary node $W_4x_1y_f$ (i.e. the one of which the weight has the largest absolute value as shown in Fig. 4.4). Then, more nodes have been be added such as $W_{22}x_1^2\cos(\omega t)$ and $W_{23}x_1^2\sin(\omega t)$ as per $W_1x_1^2$; then $W_{24}x_1x_2\cos(\omega t)$ and $W_{25}x_1x_2\sin(\omega t)$

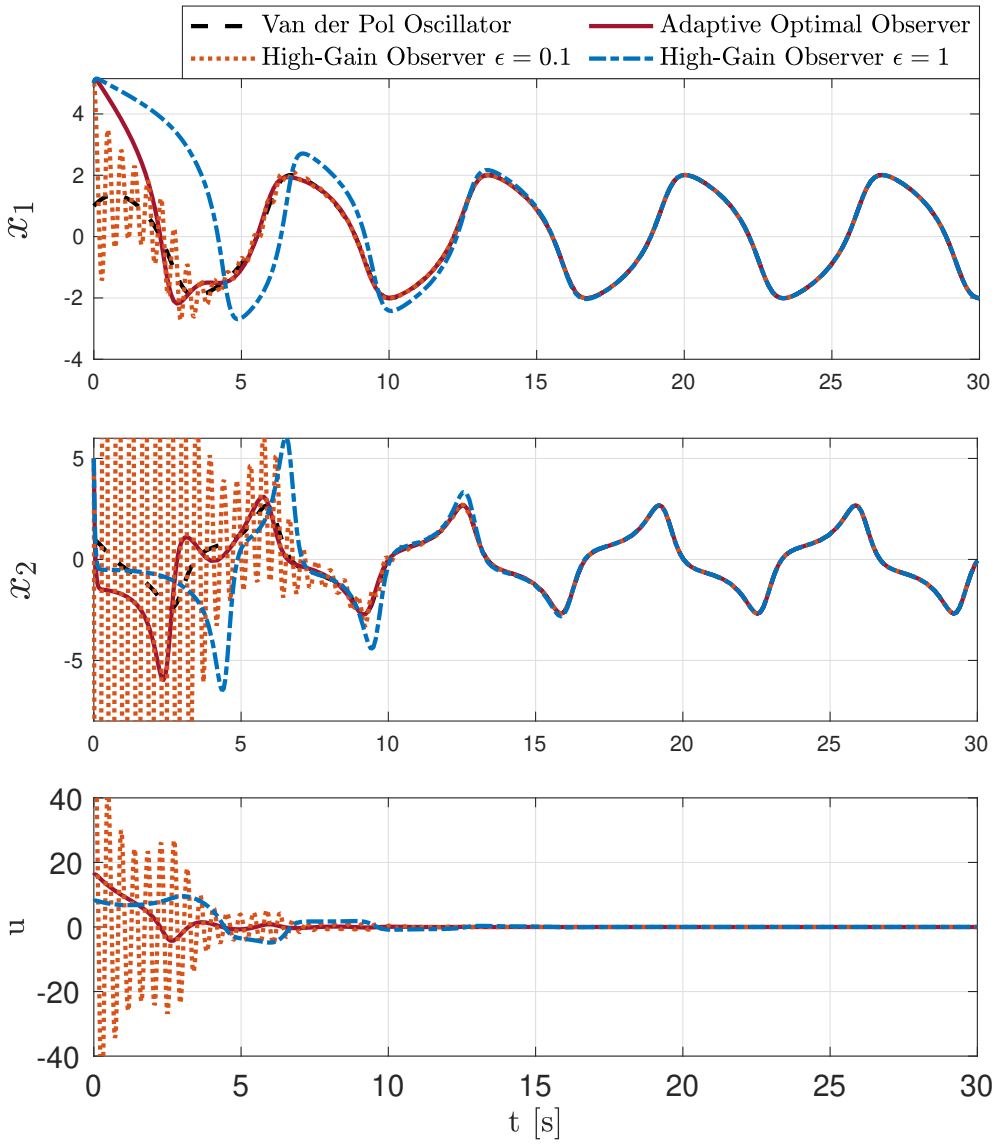

Figure 4.5: The comparison results of the adaptively-learned optimal observer (19 nodes) against the high-gain observer with $\epsilon = 0.1$ and 1.

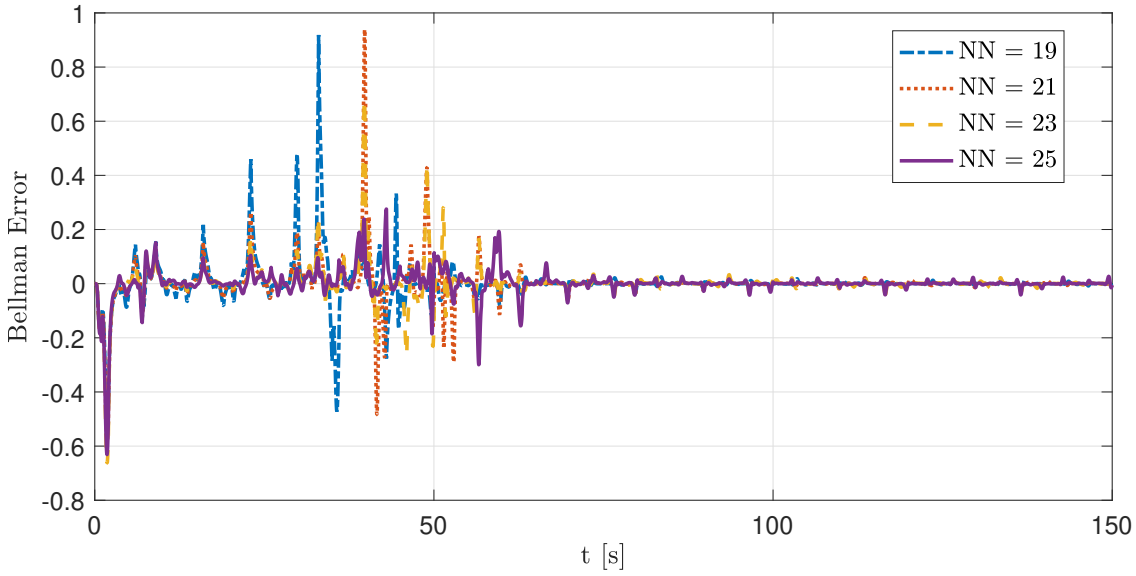

Figure 4.6: The Bellman error of different number of nodes (NN) over time.

as per $W_2 x_1 x_2$; and so forth.

Table 4.1: Results considering the time-dependence of value functional V with different number of nodes in the adaptive critic neural network.

Number of nodes	19	21	23	25
MAE of Bellman error	0.9187	0.9389	0.6811	0.6311
ISE of Bellman error	37.74	34.82	25.44	17.17
Time-dependence	No	Yes	Yes	Yes

We run the simulation for different number of nodes to evaluate the impact of considering the time-dependence of V on the Bellman error in terms of maximum absolute error (MAE) and integral squared error (ISE). Fig. 4.6 shows the Bellman error over time due to different number of nodes. The MAE and ISE tend to be reduced when increasing the number of nodes as shown in Table 4.1. However, the slight improvement on the Bellman error does not bring considerable advantage for the convergence but only increases the complexity of the neural network. One can see, in this case, the adaptive critic without considering the time-dependence of V delivers as good track-

ing performance. Therefore, it might be fair to neglect t in the critic design for our example to trade off the complexity over accuracy, which is also a practical way for implementation on real engineering problems.

4.9 Conclusions

In this chapter, we formulated for the first time a model-free, adaptive optimal observer design problem for a class of nonlinear systems using a new representation, i.e., finding an admissible control that minimises a pre-defined cost functional. Note that this is preliminary work and the method can be extended to address a wider class of nonlinear systems. An OHJB equation was established that gives the optimal solution. Specifically, we ensured the existence, stability, and optimality of the optimal solution of an infinite-horizon cost functional. This allowed us to build general results in policy iteration that successively approximate the optimal value functional. Incorporating the idea of Q-learning, a novel online adaptive solution was proposed. We justified a proper definition of a Q-functional in a continuous-time context. The Q-functional can be approximated by an adaptive critic neural network, of which the convergence was rigorously proved by stability analysis in the Lyapunov sense. The algorithm is model-free in the sense that the Q-learning Bellman equation is approximately solved without knowing the system dynamics. The effectiveness of the proposed algorithm was addressed in a Van der Pol oscillator case study. Although it was not the focus of this chapter, the proposed results can also be easily extended to general stabilisation and tracking control problems.

CHAPTER 5

ENGINE DYNAMICS AND MODELLING*

The use of Wankel rotary engines as a range extender has been recognised as an appealing method to enhance the performance of hybrid electric vehicles (HEV). They are effective alternatives to conventional reciprocating piston engines due to their considerable merits such as lightness, compactness, and higher power-to-weight ratio. However, further improvements on Wankel engines in terms of fuel economy and emissions are still needed. The objective of this work is to investigate the engine modelling methodology that is particularly suitable for the theoretical studies on Wankel engine dynamics and new control development.

In this chapter, control-oriented models are developed for a 225CS Wankel rotary engine produced by Advanced Innovative Engineering (AIE) UK Ltd. Through a synthesis approach that involves state space (SS) principles and artificial neural networks (NN), the Wankel engine models are derived by leveraging both first-principle knowl-

*The content of this chapter is adapted from the author's own work [2], where some materials have been re-used. The experiments were carried out with the assistance from the University of Bath.

edge and engine test data. We first re-investigate the classical physics-based mean value engine model (MVEM). It consists of differential equations mixed with empirical static maps, which are inherently nonlinear and coupled. Therefore, we derive a SS formulation which introduces a compact control-oriented structure with low computational demand. It avoids the cumbersome structure of the MVEM and can further facilitate the advanced modern control design. On the other hand, via black-box system identification techniques, we compare the different NN architectures that are suitable for engine modelling using time-series test data: 1) the Multi-Layer Perceptron (MLP) feedforward network; 2) the Elman recurrent network; 3) the Nonlinear AutoRegressive with eXogenous inputs (NARX) recurrent network. The NN models overall tend to achieve higher accuracy than the MVEM and the SS model and do not require *a priori* knowledge of the underlying physics of the engine.

5.1 Introduction

By far the most important reasons for the limited use of the Wankel rotary engine are the fuel economy and emissions [198]. The high hydrocarbon emissions and poor fuel economy are generally believed to be caused by the unburnt air-fuel mixture leaking past the apex seals among chambers [4]. Moreover, the common port fuel injection (PFI) configuration for the Wankel engine can result in the typical wall-wetting phenomenon, where a considerable portion of fuel keeps condensing on and evaporating at the intake manifold wall as fuel puddles [199]. When the fuel puddles heavily, it contributes to the air-fuel ratio excursion. The air-fuel ratio control is a critical task in order to assure a satisfactory efficiency of the three-way catalytic converters and to meet the emission requirement. An intuitive way to overcome this issue is to apply direct fuel injection (DFI), the feasibility of which was investigated in previous work [200]. On the other hand, we show in the later Chapters 6 and 7 that novel nonlinear observer-based controllers can be developed for the Wankel engine where the fuel puddle dynamics are estimated in real time.

The shortcomings of high emissions have severely limited the application of Wankel engines in the automotive industry. The major Wankel engine producer Mazda ceased the last production of the RX-8 car in 2012. The modern Wankel engine finds use outside the automotive industries, where its application prevails only in bespoke areas such as unmanned aerial vehicles (UAVs), marine, auxiliary power units [201]. In recent years, investigations have been made on the idea of incorporating a Wankel engine as a range extender for hybrid electric vehicles (HEV) [202]. This motivates the work on the Wankel engine modelling and control with the objectives of improving the fuel economy and reducing the emissions of the Wankel engine. At the present time, control-oriented mathematical models that specifically describe the Wankel engine dynamics are lacking.

Engine modelling has been playing an important role in the engine development and optimisation process. Accurate and fast engine simulations could allow for rapid incorporation of new control design. The control development process can be significantly shortened when using efficient computational tools. A high-fidelity model may reduce the number of hardware prototypes needed and development cost. Commercial software such as AVL BOOST offers a dedicated platform for modelling the Wankel engine dynamics as a one-dimensional computational fluid dynamics (CFD) model [200]. The CFD models usually have high fidelity and require a large amount of computational time. In the case of real-time control development, a physics-based mean value engine model (MVEM) developed by Hendricks [126][125] is widely used with low-fidelity but fast running speed. Instead of cycle by cycle analysis, the MVEM presents the average response of multiple ignition cycles in the time domain. It was originally derived mainly for reciprocating engines and may not be able to directly apply to the Wankel rotary engine. Furthermore, the MVEM contains nonlinear differential equations mixed with empirical static maps and often has a cumbersome structure. One way to overcome the complexity and to provide fast solution for control design is linear state space (SS) modelling. The SS model plays a central role in modern control theory [9] and has been commonly used as a framework for robust

control, optimal control, etc. It can effectively deal with a multi-input, multi-output (MIMO) system such as engines. A linear SS model for engines was proposed in [127] for the application of linear quadratic control. The model was then generalised in [128] for internal combustion engines for control analysis. In the last decade, the artificial Neural Networks (NN) have been seen as an attractive approach for dynamic system modelling and control. There are many studies on the application of NN on engine modelling, e.g., [129–134] therein. NN can be regarded as a black-box system identification approach that is conceptually simple, easy to use, and has excellent approximation properties.

In this chapter, we first establish the MVEM for the Wankel engine. By properly selecting the state variables, a SS formulation is realised and then linearised around a nominal operating point. We then investigate different classes of NN for the Wankel engine modelling: 1) the Multi-Layer Perceptron (MLP) network; 2) the Elman network; 3) the Nonlinear AutoRegressive with eXogenous inputs (NARX) network. The comparison and analysis for the performance and complexity of the proposed models are presented in the end.

5.2 Experimental Setup and Data Collection

This section presents the experimental setup for the engine tests. The test data collected are subsequently used for the Wankel engine modelling and validation procedure.

The Wankel engine under investigation for dynamic modelling is a 225CS rotary engine, produced by Advanced Innovative Engineering (AIE) UK Ltd. It is a single-rotor, peripheral-port-injected, twin-spark engine and was previously configured to have a nominal peak power output of 30kW for aerospace use on drones. Due to its high specific power output, there is a recent interest in using it as a range extender for HEV. Figure 5.1 shows an image of the computer-aided design (CAD) model of

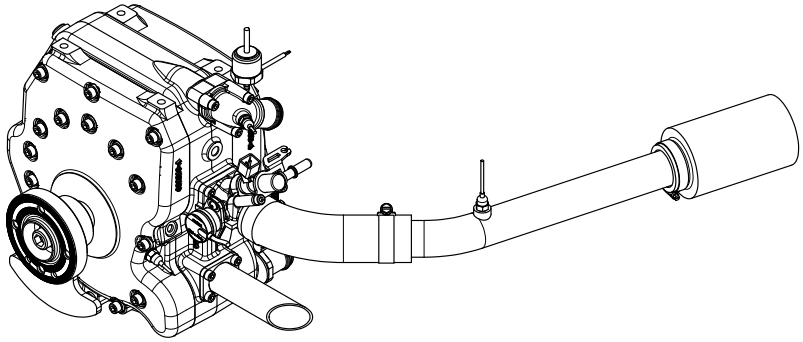

Figure 5.1: A CAD model of AIE (UK) Ltd 225CS Wankel rotary engine with the intake and exhaust pipe installed [2].

the engine. Table 5.1 presents the fundamental engine properties of the AIE 225CS Wankel rotary engine.

Table 5.1: The fundamental properties of the AIE 225CS Wankel rotary engine.

Definition	Value
Generating Radius	69.5 mm
Eccentricity	11.6 mm
Offset/Equidistance	2 mm
Width of Rotor Housing	51.941 mm
Total Displacement	225 cc
Mass (excluding ancillaries)	10 kg
Compression Ratio	9.6:1

The engine experiments are carried out in an engine test cell at the Institute for Advanced Automotive Propulsion Systems (IAAPS) at the University of Bath. The test cell is equipped with an AC dynamometer for the assessment of the engine performance. The maximum nominal power and speed allowed from the AC motor is around 50 kW and 8500 RPM, respectively. The test facilities include a Sierra CP Test Automation System with the proprietary CADET software, which enables the dynamometer control and data acquisition via the transducers installed on the engine. The system can also collect data via the automotive standard Controller Area

Network (CAN) bus, which can be compared online with the data from the Engine Control Unit (ECU). The ECU employed by AIE is an EM80 model produced by the project partner General Engine Management Systems (UK) Ltd (GEMS). It is online configurable when connected to the GEMS GWv4 proprietary software so that the user is able to control and monitor the engine parameters such as air-fuel ratio control, and fly-by-wire throttle control.

Data Collection

To be able to calibrate the engine model (or to train the NN) and validate its fidelity, one needs to capture as much information as possible from the engine tests. The data collected should cover a broad engine operation range. We run the engine from 3500 [RPM] to 6000 [RPM], the throttle angle is swept from 20 [^{o}] to 90 [^{o}] (i.e., fully opening) for each fixed speed with the resolution of 500 [RPM]. Figure 5.2 shows the operation trajectories covered in the engine tests. The collected data including the throttle angle, the intake manifold pressure and temperature, the engine speed and torque, and the air-fuel ratio are shown in Figure 5.3. The sampling period of the signals is chosen as 0.02 [s].

5.3 Engine Dynamics and Modelling

The complexity of the internal combustion engines includes highly nonlinear characteristics. It is often necessary to first model the engine dynamics for control design and simulation before implementing a real controller. Unlike one power pulse per two revolutions in four-stroke reciprocating engines, Wankel rotary engines generate three power pulses per revolution, which delivers advantages of high revolutions per minute and smoothness [203]. Following our work on engine modelling in [136] and the M.Sc. thesis of the author [3] as Fig. 5.4, we extend the engine model to Wankel rotary engines. There has been a large number of studies, e.g. the mean-value engine model (MVEM) developed by Hendricks [126][125], on the dynamics of reciprocating

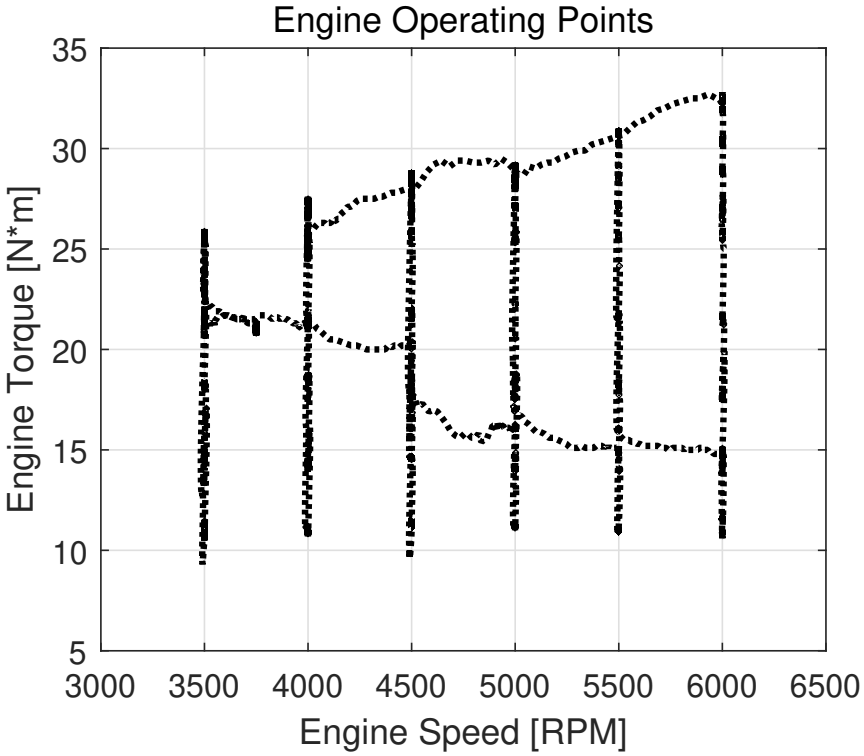

FIGURE 5.2. Operating points collected during engine tests with the range: Engine speed 3500 - 6000 [RPM]; Engine torque 5 - 35 [Nm].

engines whereas few on the modelling of Wankel engines, e.g. [4][203][204] around the early 1980s. However, it is feasible to model the Wankel engine using an equivalent reciprocating MVEM since it operates with the same Otto cycle, i.e. a single rotor Wankel engine is equivalent to a two-cylinder four-stroke reciprocating engine [4]. Fig. 5.5 shows the correspondence of the intake, compression, expansion and exhaust phases of the two types of engines. This section describes a zero-dimension model of the port fuel injected Wankel engine dynamics.

5.3.1 Intake Air flow Model

The air mass flow rate $\dot{m}_{at}$ passing the throttle can be described [126] as

$$\dot{m}_{at}(\alpha, p_m) = m_{at1}\frac{p_a}{\sqrt{T_a}}TC(\alpha)PRI(p_m) + m_{at0} \tag{5.1}$$

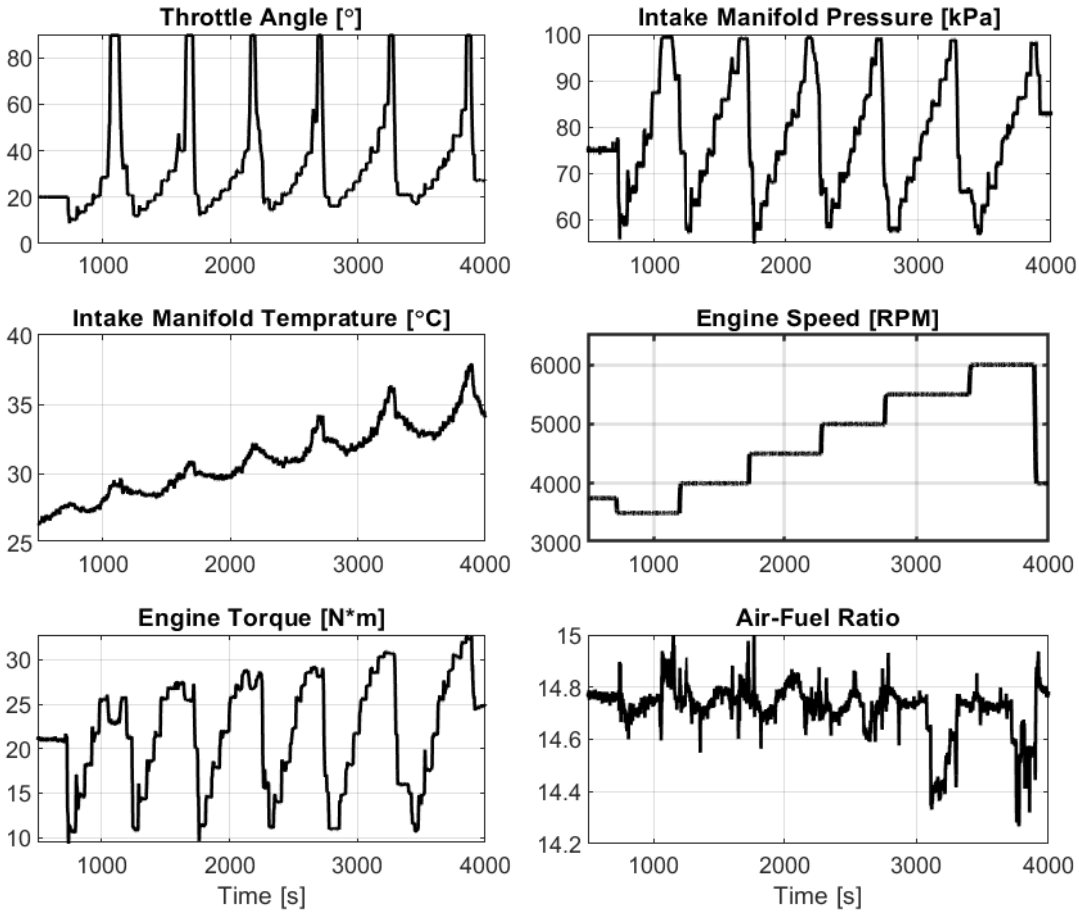

FIGURE 5.3. Engine test data collected for modelling and validation. The data traces start at the time 500 s after warming up the engine.

where $m_{at1} = c_t \frac{\pi}{4} D^2 \sqrt{2\kappa/R(\kappa-1)}$ is a physical constant related to the ratio of the specific heats κ, the gas constant R, the flow coefficient c_t and the diameter D of throttle body throat; m_{at0} is a fitting constant; p_a and T_a are the ambient pressure and temperature in Kelvin, respectively; $TC(\alpha) = 1 - \cos(\alpha - \alpha_0)$ denotes the throttle characteristics function of the throttle plate angle α and the leakage constant α_0, which approximates the effective throttle area; $PRI(p_m)$ refers to the pressure ratio influence from

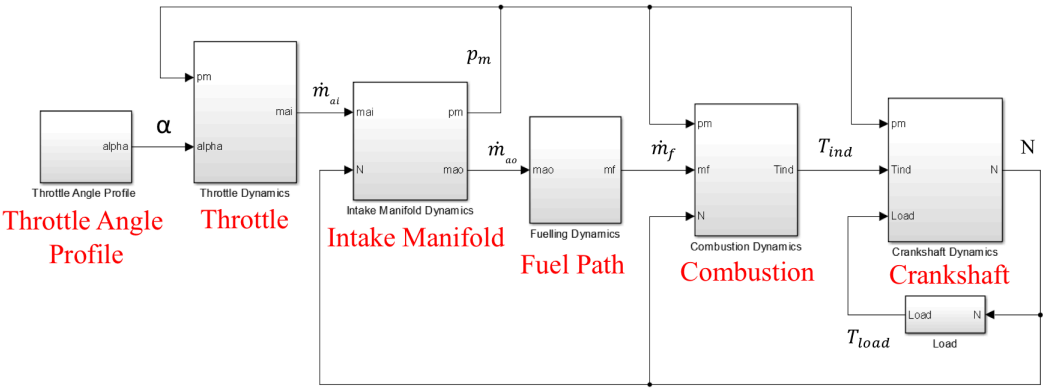

Figure 5.4: The MVEM block diagram in MATLAB Simulink [3].

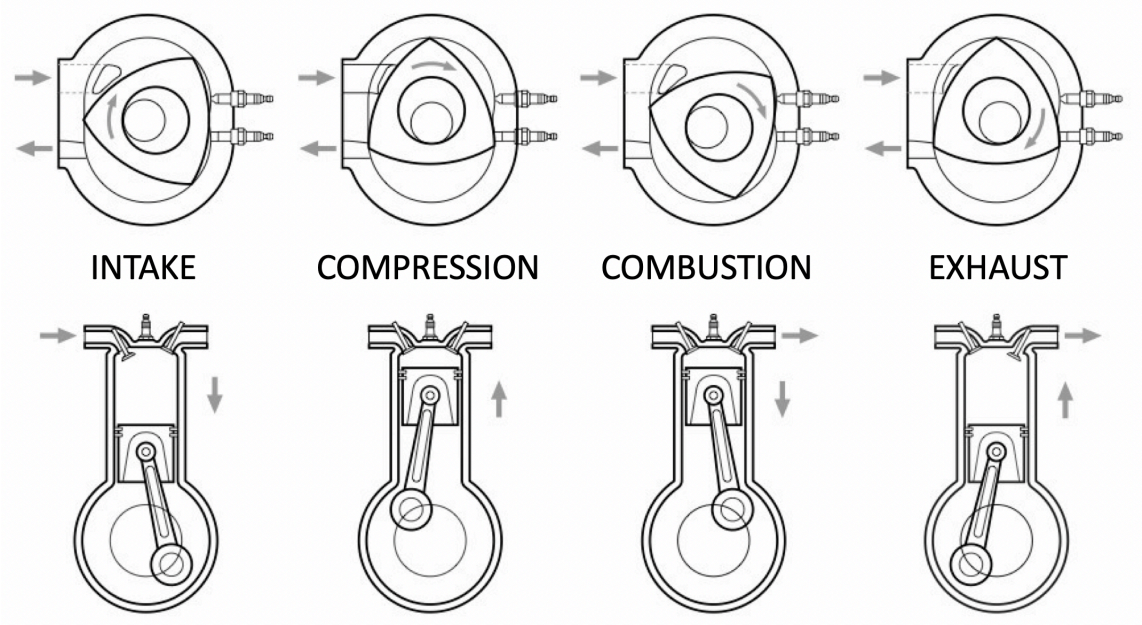

FIGURE 5.5. Comparison of Otto cycles between the Wankel rotary engine (above) and reciprocating engine (below). The correspondence of the four strokes (i.e., intake, compression, combustion, and exhaust) between both designs is presented (Modified from [205]).

the choke/sonic compressible flow, which can be expressed as

$$PRI(p_m) = \begin{cases} \sqrt{1-(\frac{p_r-p_c}{1-p_c})^2} & \text{if } p_r \geq p_c \text{ (choked)} \\ 1 & \text{if } p_r < p_c \text{ (sonic)} \end{cases} \tag{5.2}$$

where p_c is the threshold point and $p_r = p_m/p_a$ is the ratio of the intake manifold pressure p_m to the ambient pressure p_a.

Neglecting the heat transfer [125], an adiabatic model of the air-filling dynamics in the intake manifold can be given as

$$\dot{p}_m = \frac{\kappa R}{V_m}(\dot{m}_{at}T_a - \dot{m}_a T_m) \tag{5.3}$$

$$\dot{T}_m = \frac{RT_m}{p_m V_m}[\dot{m}_{at}(T_a\kappa - T_m) - \dot{m}_a(T_m\kappa - T_m)] \tag{5.4}$$

where T_m is the manifold temperature and V_m is the manifold volume. Then the port air mass rate $\dot{m}_a$ can be given as a nonlinear function of the manifold pressure p_m and engine speed N such that

$$\dot{m}_a(p_m, N) = \sqrt{\frac{T_m}{T_a}}\frac{V_d}{120RT_m}\eta_{vol}(p_m, N)p_m N \tag{5.5}$$

where V_d is the engine displacement and η_{vol} is the volumetric efficiency.

5.3.2 Fuel Puddle Model

Due to the "wall-wetting" phenomenon, the final fuel flow rate $\dot{m}_f$ is the sum of the fuel puddle flow rate $\dot{m}_{fpe}$ and the fuel vapour flow rate $\dot{m}_{fve}$ entering the combustion chamber

$$\dot{m}_f = \dot{m}_{fpe} + \dot{m}_{fve} = m_{fp}/\tau_p + m_{fv}/\tau_m \tag{5.6}$$

where τ_p and τ_m are the characteristic manifold time constants for the puddle m_{fp} and vapour m_{fv} fuel mass, respectively [206]. Their dynamics can be taken as a set of

two first-order processes with time constant τ_f as

$$\begin{cases} \dot{m}_{fp} = \chi \dot{m}_{fi} - (1/\tau_f) m_{fp} - \dot{m}_{fpe} \\ \dot{m}_{fv} = (1-\chi)\dot{m}_{fi} + (1/\tau_f) m_{fp} - m_{fv}/\tau_m \end{cases} \tag{5.7}$$

where $\dot{m}_{fi}$ is the injected fuel flow rate (i.e. the control command for the fuel injector) and χ, $0 \leq \chi < 1$, is a fraction of injected fuel that is deposited on the manifold wall as fuel puddles. It should be noted that the fuel puddle model (5.6)(5.7) is perceived to be more correct compared to our previous version in [136] since it considers the effect of the fuel puddle flow $\dot{m}_{fpe}$ (Couette flow) entering the chamber. Fig. 5.6 illustrates the fuel injection process for a PFI engine.

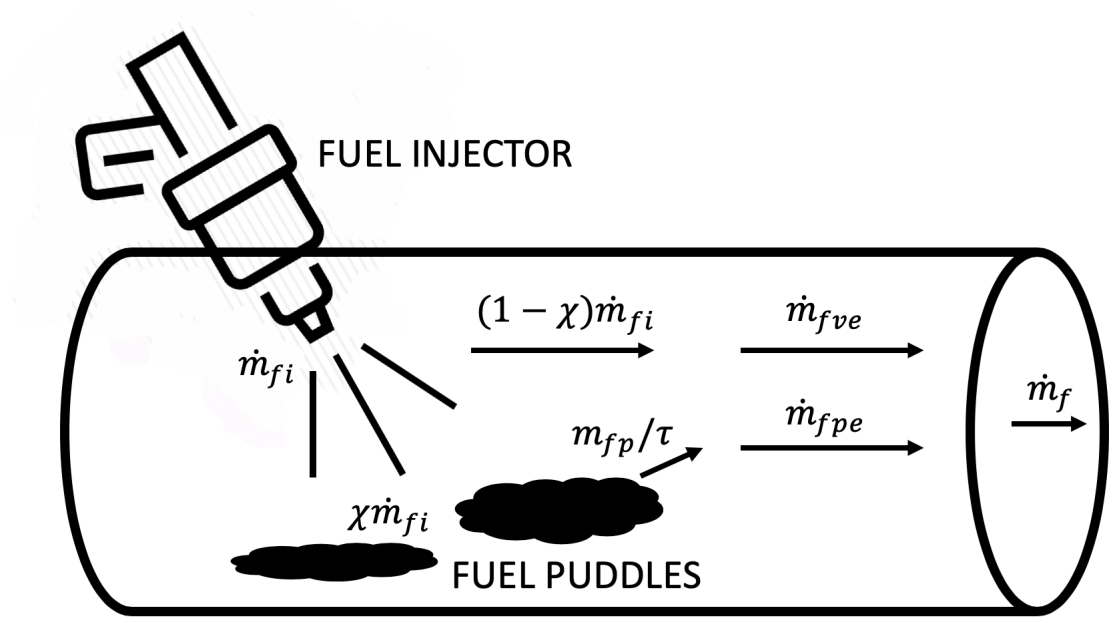

FIGURE 5.6. Fuel injection process for a Port Fuel-Injected (PFI) engine. The effect from the fuel puddles on the final fuel flow are illustrated.

5.3.3 Combustion Model

The combustion characteristics can be modelled in terms of the engine indicated torque τ_{ind} using a heat release approach, where the total power created by heat re-

lease during combustion is subject to the actual burnt fuel mass flow rate $\dot{m}_{fb}$. A significant feature in the combustion dynamics of Wankel engines is the leakage and the crevice volume between chambers. The leakage past the apex and side seals must be considered when evaluating the combustion performance [203]. The crevice volume and leakage flow is shown in Fig. 5.7. The actual burned fuel flow rate $\dot{m}_{fb}$ can be written [4] as

$$\dot{m}_{fb} = \frac{1}{\lambda}[\dot{m}_a - \dot{m}_{leakage} - \frac{\dot{p}_b}{p_b} m_{crevice}] \tag{5.8}$$

where $\dot{m}_{leakage}$ and $m_{crevice}$ denote the leakage rate and crevice mass of the air-fuel mixture, p_b is the chamber pressure, and $\lambda = \dot{m}_a/\dot{m}_f$ is the air-fuel ratio, which is the control object to be regulated around the stoichiometric value, i.e. $\lambda_d = 14.67$ for petrol. Hence, the indicated engine torque τ_{ind} [126] can be determined as

$$\tau_{ind} = H_u \frac{\eta_{th}(N, p_m, \theta_{SA}, \lambda)\dot{m}_{fb}}{N} \tag{5.9}$$

where H_u is the fuel energy constant and η_{th} is a complex nonlinear function of the engine speed N, the manifold pressure p_m, the spark advance angle θ_{SA}, and the air-fuel ratio λ.

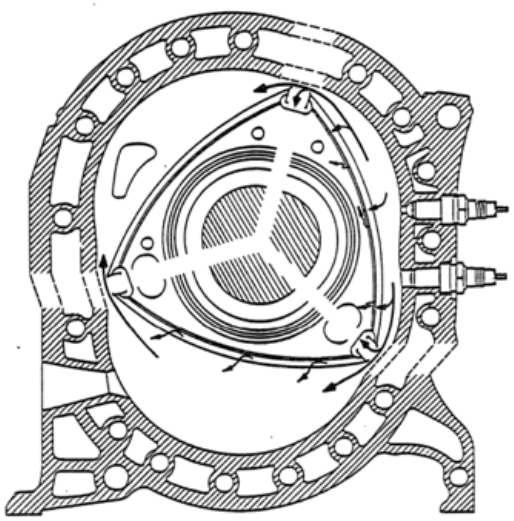

Figure 5.7: Crevice volume and leakage gas flow [4].

5.3.4 Eccentric Shaft Model

The eccentric shaft dynamics can be expressed using Newton's second law as

$$J\dot{N} = \tau_{ind} - \tau_{fric} - \tau_{load} \tag{5.10}$$

where J is the scaled engine moment of inertia, τ_{fric} and τ_{load} refer to the friction and the load torque, respectively [126].

An MVEM simulator for the AIE 225CS Wankel engine is established in Matlab/Simulink. The model is verified with the experimental data sets and the fuel injection model is integrated with a nonlinear observer-based air-fuel ratio controller (see our previous work [207] for details). The load torque in the simulator is a user-defined function of the engine speed. Fig. 5.8 outlines the block diagram of the MVEM simulator. Fig. 5.9 shows the throttle profile and the other corresponding signals generated by the MVEM simulator.

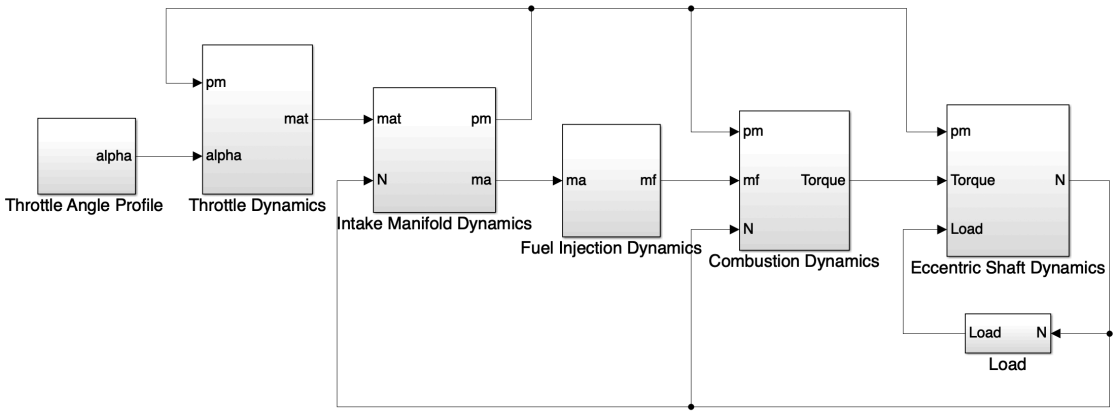

FIGURE 5.8. The block diagram of the MVEM simulator for the Wankel engine. The simulator consists of seven major blocks for 1) throttle angle profile; 2) throttle dynamics; 3) intake manifold dynamics; 4) fuel injection dynamics; 5) combustion dynamics; 6) eccentric shaft dynamics; and 7) load profile.

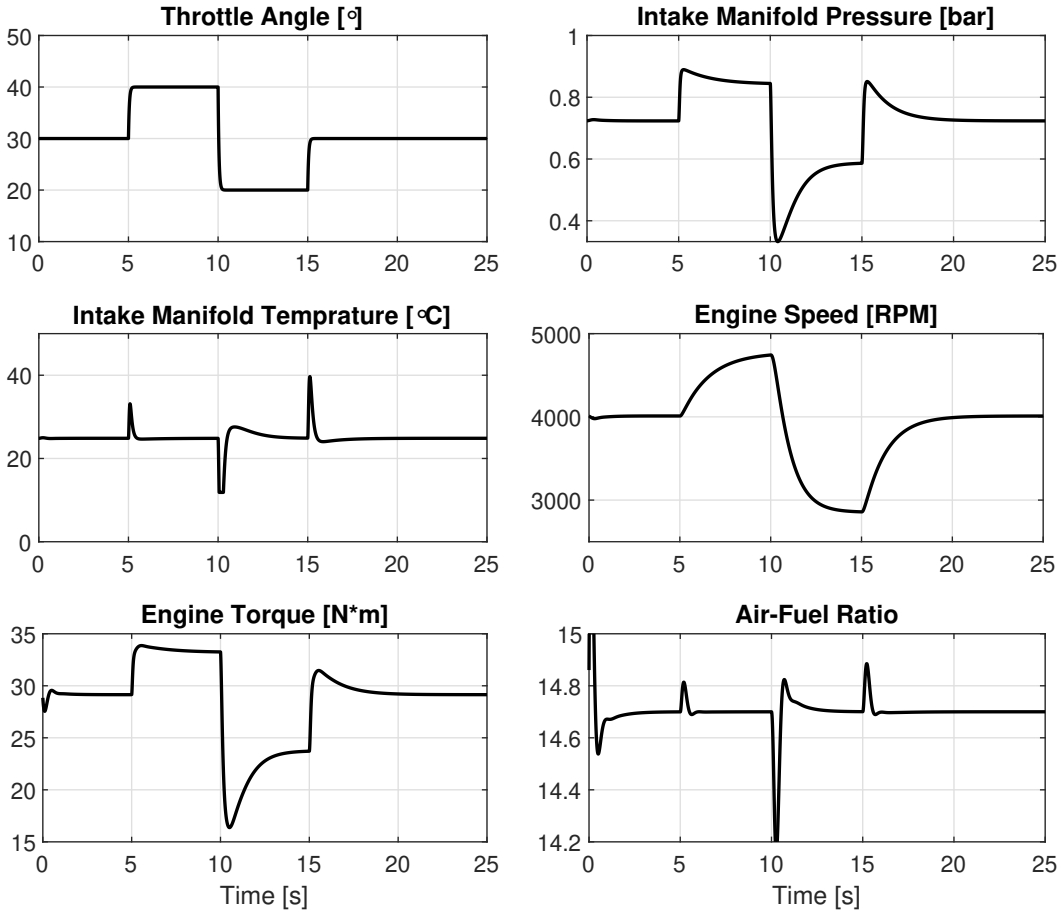

FIGURE 5.9. Engine dynamic responses generated by the MVEM simulator for a given throttle profile, which includes the intake manifold pressure, the intake manifold temperature, the engine speed, the engine torque, the air-fuel ratio.

5.4 State Space Realisation

The MVEM consists of differential equations mixed with empirical static maps to model the inherently nonlinear and coupled dynamics of the engine. In modern control engineering, it is common to develop a more compact model, compared to the MVEM, that uses state variables to describe the system by a set of first-order differential equations. This is known as State-Space (SS) modelling. To leverage the previous results, we can first convert the MVEM into a nonlinear SS representation, then derive a linear SS model via linearisation techniques around a nominal operating point.

5.4.1 Nonlinear State Space Model

$$\begin{cases} \dot{x} = f(x,u,w) \\ y = h(x,u) \end{cases} \tag{5.11}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $w \in \mathbb{R}^l$, and $y \in \mathbb{R}^r$ represent the state, control input, disturbance, and output vectors, respectively; $f(x,u,w)$, $h(x,u)$ are the nonlinear functions that lump the MVEM together. We define the state vector from the MVEM for the Wankel engines as

$$x = \begin{bmatrix} p_m \\ T_m \\ N \\ m_{fp} \\ m_{fv} \end{bmatrix} \begin{matrix} \text{intake manifold pressure} \\ \text{intake manifold temperature} \\ \text{engine speed} \\ \text{fuel puddle mass} \\ \text{fuel vapour mass} \end{matrix} \qquad n = 5 \tag{5.12}$$

with the control input, disturbance, and output as

$$u = \begin{bmatrix} \alpha \\ \dot{m}_{fi} \\ \theta_{SA} \end{bmatrix} \begin{matrix} \text{throttle angle} \\ \text{injected fuel flow rate} \\ \text{spark advance angle} \end{matrix} \qquad m = 3 \tag{5.13}$$

$$w = \tau_{load} \text{load torque} \qquad l = 1 \tag{5.14}$$

$$y = \begin{bmatrix} \tau_{ind} \\ \dot{m}_f \\ \dot{m}_a \end{bmatrix} \begin{matrix} \text{indicated torque} \\ \text{fuel flow rate entering the chamber} \\ \text{air flow rate entering the chamber} \end{matrix} \qquad r = 3 \tag{5.15}$$

5.4.2 Linearisation

We linearise the SS model (5.11) around the nominal operating point shown in Table 5.2. The operator δ denotes the new variable centred about the operating point, e.g. $\delta x(t) = x(t) - x^0$. For the throttle body model, we can write $\delta\dot{m}_{at}$ from (5.1) and (5.2) by taking partial derivatives with respect to α and p_m as

$$\begin{aligned} \delta\dot{m}_{at} &= m_{at1}\frac{p_a}{\sqrt{T_a}}[PRI(p_m^0)\frac{\partial TC}{\partial \alpha}\delta\alpha + TC(\alpha^0)\frac{\partial PRI}{\partial p_m}\delta p_m] \\ &= c_{11}\delta\alpha + c_{12}\delta p_m \end{aligned} \tag{5.16}$$

with constants c_{11} and c_{12}.

In the intake manifold model, for (5.5), taking partial derivatives with respect to p_m and N for $\delta\dot{m}_a$ yields

$$\begin{aligned} \delta\dot{m}_a &= \sqrt{\frac{T_m}{T_a}}\frac{V_d}{120RT_m}\eta_{vol}(p_m^0, N^0)(N^0\delta p_m + p_m^0\delta N) \\ &= c_{21}\delta p_m + c_{22}\delta N \end{aligned} \tag{5.17}$$

with constants c_{21} and c_{22}. Similarly, we can write for $\delta\dot{T}_m$ and $\delta\dot{p}_m$ based on (5.4) and (5.3) as

$$\begin{aligned} \delta\dot{T}_m &= \frac{\partial\dot{T}_m}{\partial T_m}\delta T_m + \frac{\partial\dot{T}_m}{\partial p_m}\delta p_m + \frac{\partial\dot{T}_m}{\partial N}\delta N \\ &= c_{31}\delta T_m + c_{32}\delta p_m + c_{33}\delta N \end{aligned} \tag{5.18}$$

$$\begin{aligned} \delta\dot{T}_m &= \frac{\kappa R}{V_m}(c_{12}T_a - \frac{\partial\dot{m}_a T_m + \dot{m}_a\partial T_m}{\partial p_m})\delta p_m + \frac{c_{11}\kappa R}{V_m}\delta\alpha - \frac{\kappa R}{V_m}\frac{\dot{m}_a T_m + \dot{m}_a\partial T_m}{\partial N}\delta N \\ &= c_{41}\delta p_m + c_{42}\delta\alpha + c_{43}\delta N + c_{44}\delta T_m \end{aligned} \tag{5.19}$$

with constants c_{31}, c_{32}, c_{33}, c_{41}, c_{42}, c_{43}, and c_{44}.

For fuel puddle dynamics, substituting (5.6) into (5.7) gives

$$\delta \dot{m}_{fp} = \chi \delta \dot{m}_{fi} - (1/\tau_f + 1/\tau_p)\delta m_{fp} \tag{5.20}$$

$$\delta \dot{m}_{fp} = (1-\chi)\delta \dot{m}_{fi} + (1/\tau_f)\delta m_{fp} - 1/\tau_m)\delta m_{fv} \tag{5.21}$$

It is reasonable to assume the leakage and crevice volume is negligible for the linearisation, i.e. $\dot{m}_f \approx \dot{m}_{fb}$ for complete combustion. Thus, we can write $\delta \dot{N}$ based on (5.9) and (5.10) as

$$\begin{aligned} \delta \dot{N} &= \frac{1}{J}\frac{\partial(\tau_{ind} - \tau_{friction})}{\partial N}\delta N + \frac{1}{J}\frac{\partial(\tau_{ind} - \tau_{friction})}{\partial p_m}\delta p_m + \frac{1}{J}\frac{\partial(\tau_{ind})}{\partial \theta_{SA}}\delta\theta_{SA} \\ &+ \frac{1}{J}\frac{\partial(\tau_{ind})}{\partial \lambda}\delta\lambda + \frac{1}{J}\frac{\partial(\tau_{ind})}{\partial \dot{m}_f}\delta\dot{m}_f - \frac{1}{J}\delta\tau_{load} \\ &= c_{51}\delta N + c_{52}\delta p_m + c_{53}\delta\theta_{SA} + c_{54}\delta\lambda + c_{55}\delta\dot{m}_f - \frac{1}{J}\delta\tau_{load} \end{aligned} \tag{5.22}$$

with constants c_{51}, c_{52}, c_{53}, c_{54}, and c_{55}. It should be noted that the term $c_{54}\delta\lambda \approx 0$ as the air-fuel ratio is closely regulated around a desired constant, e.g. the stoichiometric value.

Table 5.2: Nominal operating point chosen for engine SS model linearisation.

Engine Variable	Symbol	Nominal Operating Point
Throttle Angle	α^0	30 [^{o}]
Engine Torque	τ^0_{ind}	30 [Nm]
Engine Speed	N^0	4000 [RPM]
Intake Manifold Temperature	T^0_m	25 [^oC]
Intake Manifold Pressure	p^0_m	0.7 [bar]
Spark Advance Angle	θ^0_{SA}	18 [^{o}]
Air-Fuel Ratio	λ^0	14.7:1

Thus, we can determine the linear SS model for a typical nominal operating point shown in Table 5.2 based on the above derivation as

$$\begin{cases} \delta\dot{x} = A\delta x + B\delta u + E\delta w \\ \delta y = C\delta x + D\delta u \end{cases} \tag{5.23}$$

with the matrices A, B, C, D, and E as

$$A = \begin{bmatrix} c_{41} & c_{44} & c_{43} & 0 & 0 \\ c_{32} & c_{31} & c_{33} & 0 & 0 \\ c_{52} & 0 & c_{51} & c_{55}/\tau_p & -c_{55}/\tau_m \\ 0 & 0 & 0 & \frac{(\tau_f+\tau_p)}{\tau_f\tau_p} & 0 \\ 0 & 0 & 0 & 1/\tau_f & -1/\tau_m \end{bmatrix}$$

$$B = \begin{bmatrix} c_{42} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & c_{53} \\ 0 & \chi & 0 \\ 0 & 1-\chi & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} c_{52}J & 0 & c_{51}J & c_{55}J/\tau_p & -c_{55}J/\tau_m \\ 0 & 0 & 0 & 1/\tau_p & -1/\tau_m \\ c_{21} & 0 & c_{22} & 0 & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & c_{53} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$E = \begin{bmatrix} 0 & 0 & 1/J & 0 & 0 \end{bmatrix}^T$$

5.5 Neural Networks

Engine dynamics are inherently nonlinear and highly coupled. The MVEM and the SS models shown in the previous sections are essentially developed on the basis of physical principles. One can see them as grey-box models that combine a partial theoretical structure and some unknown parameters derived from data. These models observe the engine dynamics with a reasonable level of accuracy, where some physical effects are, however, not directly described and still need hand-tuned correction maps due

to nonlinearity. In contrast with grey-box models, we here investigate a powerful tool, artificial Neural Networks (NN), which is one of the black-box modelling techniques and has been widely used in various engineering branches. NN have been proven to be useful for modelling nonlinear dynamic systems and can often achieve distinctly high accuracy [208].

In this section, we study different classes of NN with different configurations:

- Multi-Layer Perceptron (MLP) feedforward network
- Elman recurrent network
- Nonlinear AutoRegressive with eXogenous inputs (NARX) recurrent network

and apply them to engine dynamics modelling using purely input and output data without any a priori knowledge of its internal workings.

5.5.1 Multi-Layer Perceptron (MLP) Neural Network

We start with the classical type of NN known as Multi-Layer Perceptron (MLP). An MLP is a class of feedforward NN that consists of at least three layers of neurons: an input layer, one or more hidden layers, and an output layer. The input data $U(t) \in \mathbb{R}^p$ are propagated from the input layer to the output layer, through the hidden layers, to generate the output signals $\hat{Y}(t) \in \mathbb{R}^q$ that will track the reference output data $Y(t) \in \mathbb{R}^q$. Each layer of the MLP network is composed of some nonlinearly-activating nodes (neurons) that are fully connected and work in parallel in order to create a flow of information. Each neuron can be seen as a Multi-Input, Single-Output (MISO) computing unit where the output h is calculated by processing the weighted sum of the

inputs U with bias terms using a transfer function (i.e., an activation function ϕ). For MLP networks, especially in the machine learning community, the logistic sigmoid function $\sigma = 1/(1+e^{-x})$ and the hyperbolic tangent function $\tanh(x) = (e^{x}-e^{-x})/(e^{x}+e^{-x})$ are commonly chosen as the activation function. Other popular choices such as the Rectified Linear Unit (ReLU) are frequently used for deep learning problems. For the engine dynamics modelling problem, we select the hyperbolic tangent sigmoid function

$$\text{transig}(x) = 2/(1+e^{-2x}) - 1 \tag{5.24}$$

as the activation function. This is mathematically equivalent to $\tanh(x)$ but proved to be faster when training the network [209]. Thus, the output h of a neuron can be simply expressed as

$$h_k^l = \phi_k\left(\sum_{j=1}^{NoP} w_{kj}^l u_j + b_k^l\right) \tag{5.25}$$

where the subscript k and the superscript l denote the k-th neuron in the l-th hidden layer; the subscript j denote the j-th neuron in the previous layer; $\phi(x)$ is the activation function; NoP is the number of the neurons in the previous layer; w and b are the weight and bias of a neuron, respectively.

A computational structure such as an MLP network should be able to learn and generalise an input-output mapping from a set of training examples (the input and output data). The proper values of the network weights w and bias terms b can be found via a learning procedure where a cost function is minimised. Fig. 5.10 shows the conceptual architectural graph of a three-layer MLP. A detailed description of the NN approach is beyond the scope of the work. Readers are referred to [208] for a comprehensive analysis of artificial NN.

In this chapter, we assess the intake manifold pressure and engine torque as the main outputs for MLP network design. In some literature [133], it is noted that emissions such as NOx or CO are also chosen as the MLP network output. Here we focus on

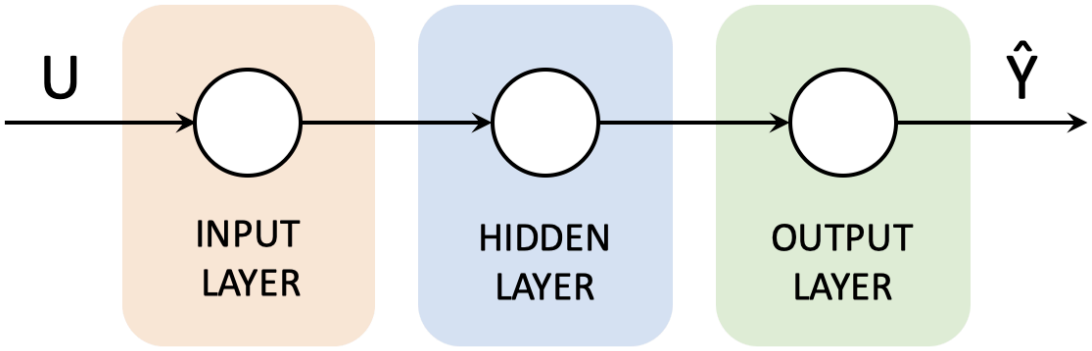

Figure 5.10: The feedforward architecture of a three-layer MLP network.

the pressure and torque as they are the primary variables in engine calibration and control. As shown in the MVEM model (Section 5.3.3), the engine torque is a highly nonlinear function of the engine speed, intake manifold pressure, spark advance, air-fuel ratio, fuel flow rate, etc. In order to investigate the potential of the NN to simulate the main features of the engine dynamics, we select the measurable input and output vectors for the MLP network as follows

$$U = \begin{bmatrix} \alpha \\ N \\ \lambda \\ \theta_{SA} \\ \dot{m}_{fi} \end{bmatrix} \begin{matrix} \text{throttle angle} \\ \text{engine speed} \\ \text{air-fuel ratio} \\ \text{spark advance angle} \\ \text{injected fuel rate} \end{matrix} \qquad p = 5 \tag{5.26}$$

$$Y = \begin{bmatrix} p_m \\ \tau_{ind} \end{bmatrix} \begin{matrix} \text{intake manifold pressure} \\ \text{indicated torque} \end{matrix} \qquad q = 2$$

Different number of neurons in the hidden layer (k) and number of hidden layers (l) can be adjusted to fulfil the purpose. These are often referred to as hyperparameters in the machine learning community. One needs to specify these values for faster training and less overfitting. Fig. 5.11 shows an MLP network with 7 neurons in the hidden

layer for the Wankel engine modelling.

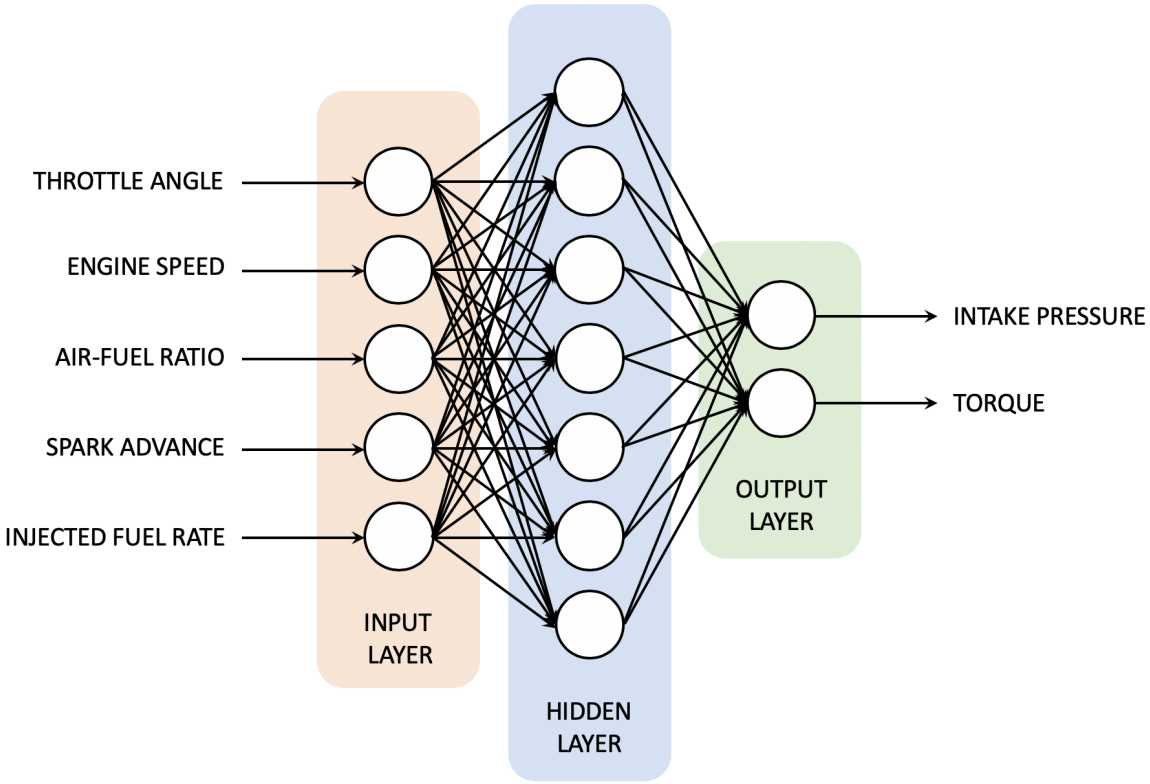

Figure 5.11: The feedforward MLP network configuration for the Wankel engine (with 7 neurons in the hidden layer).

5.5.2 Recurrent Neural Networks

The MLP described in the previous section is an NN with traditional feedforward architecture, where incoming data propagates in a single direction from the input layer to the output layer. Here we introduce a different type of NN named Recurrent Neural Networks (RNN) by considering feedback connections among neurons. The feedback can induce a dynamical effect into the computing units (neurons) by a local memory process. We investigate the RNN and its application on the engine modelling in terms of different topology.

5.5.3 Elman Recurrent Neural Network

One of the most basic architectures of RNN is the Elman network, also known as the Simple RNN. Very similar to the MLP network, the Elman network can be divided into an input layer, one or more hidden layers and an output layer. While the input and output layers are characterised by feedforward connections, the hidden layer contains recurrent connections embedded with time-delay elements. The current input and past network state are combined and processed by the neurons in the hidden layer. The output h of a neuron in the hidden layer can be written as

$$h_k^l(t) = \phi_k\Big(\sum_{j=1}^{NoP}[(w_{k_0 j}^l u_j(t) + w_{k_1 j}^l h_j(t-1) + \cdots + w_{k_n j}^l h_j(t-n_u) + b_k^l]\Big) \tag{5.27}$$

where $n_u$is the user-defined finite (often small) number of time delays. The other notations are the same as (5.25) in the MLP network. Fig. 5.12 shows the recurrent architecture of a three-layer Elman network.

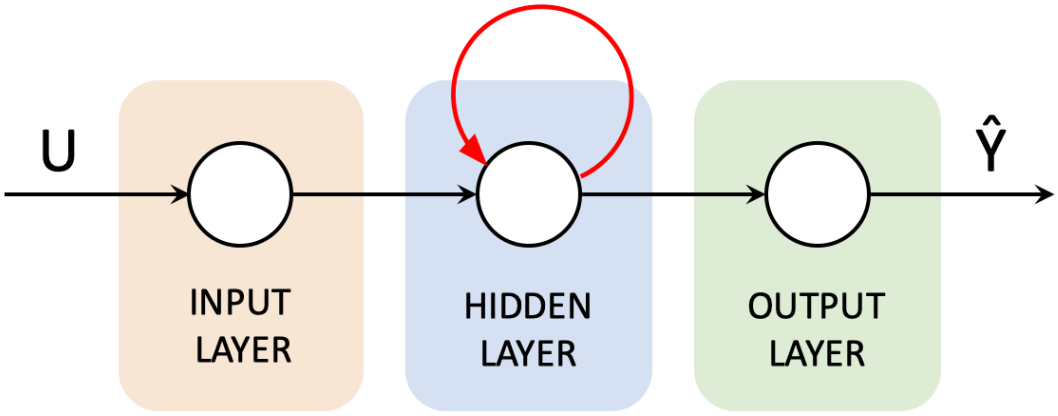

FIGURE 5.12. The recurrent architecture of a three-layer Elman network. The red arrow denotes the feedback connections among the neurons in the hidden layer.

5.5.4 Nonlinear AutoRegressive with eXogenous inputs (NARX) Recurrent Neural Network

Originally, Nonlinear AutoRegressive with eXogenous inputs (NARX) was a commonly-used method in time series modelling and analysis. An NARX model can be seen as an RNN with feedback connections enclosing several layers of the network. Differently from the Elman network, the recurrence in the NARX network is given by the feedback from the output as well. The defining equation for the NARX network can be determined for the output h of a neuron in the hidden layer as

$$
\begin{aligned}
h_k^l(t) = \phi_k\Big(\sum_{j=1}^{NoP}\big[(w_{k_0 j}^l u_j(t) + w_{k_1 j}^l h_j(t-1) + \cdots + w_{k_n j}^l h_j(t-n_u) + w_{k_1 j}^l y_j(t-1) + \\
\cdots + w_{k_n j}^l y_j(t-n_y) + b_k^l\big]\Big)
\end{aligned} \tag{5.28}
$$

where n_u and n_y are the user-defined finite (often small) number of time delays. The other notations are the same as (5.27) in the Elman network. Fig. 5.13 shows the recurrent architecture of a three-layer NARX network.

It is worth noting that the NARX network feeds back the delayed signals h and y in parallel. In the engine modelling case, since the engine toque as the actual output signal is measurable and available during the training of the network, one can leverage the actual output to create a series-parallel architecture [210], in which the actual output $Y(t)$ is used instead of feeding back the estimated output $\hat{Y}(t)$. Fig. 5.14 shows the use of the series-parallel architecture for the training of a NARX network.

5.6 Comparative Results

In this section, we present the model validation results for the SS model and the NN models. The results are compared and analysed in terms of the fidelity, the applicability, and the model calibration (or NN training) process.

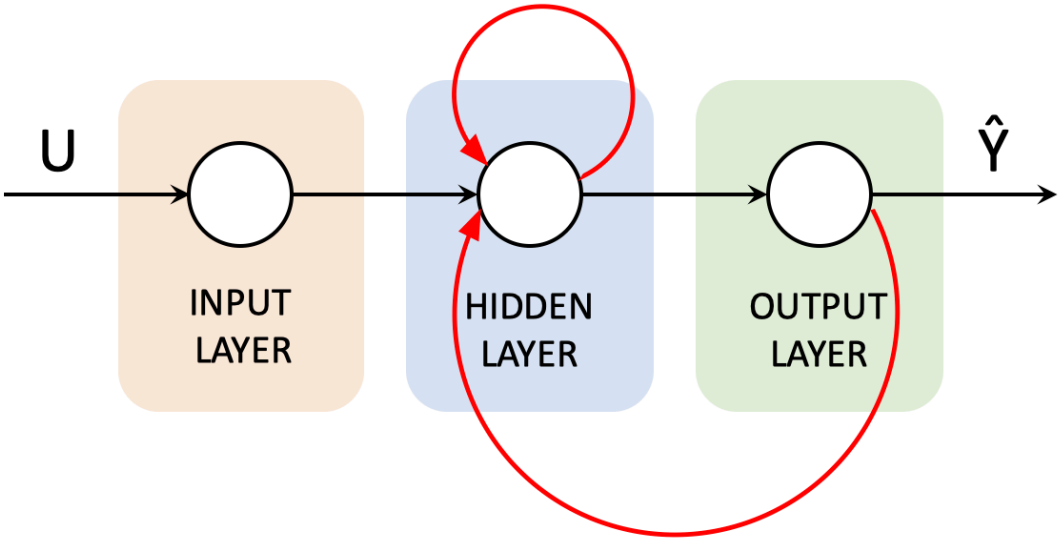

FIGURE 5.13. The recurrent architecture of a three-layer NARX network. The red arrows denote the feedback connections among the neurons in the hidden layer and from the output layer (compared with the Elman network in Fig. 5.12).

5.6.1 Linear SS Model

In order to verify the fidelity of the linear SS model, we check the torque response of the SS model against the same throttle angle profile for the MVEM. The throttle sweeps around 30 [°] with small excursion ±10 [°]. The torque and the intake manifold pressure responses from the two models are shown in Fig. 5.15. It is clear to see the linear behaviour of the SS model against the nonlinearity of the MVEM while both responses follow a similar trend.

It is evident that the linearised model is only valid in a neighbourhood of the nominal operating point. To design a global dynamic control, one can merge together different controls designed for a number of nominal operating points via simple gain scheduling or more sophisticated control techniques such as adaptive control [136].

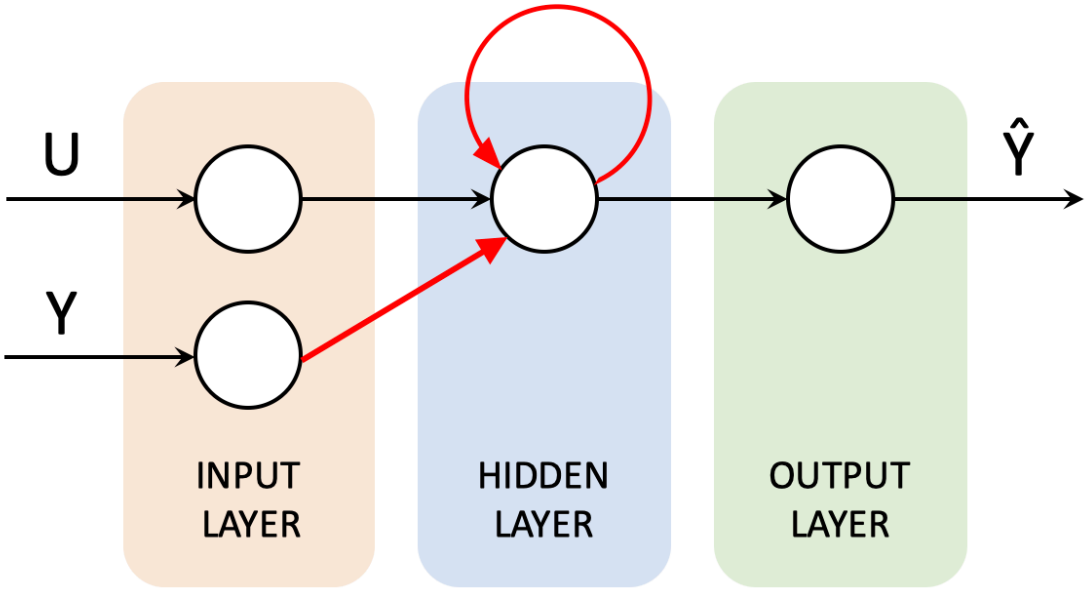

FIGURE 5.14. The recurrent series-parallel architecture of a three-layer NARX network (Training mode). The red arrows denote the feedback connections among the neurons in the hidden layer and from the actual outputs (compared with the parallel architecture in Fig. 5.13).

5.6.2 NN Models

We use the Levenberg-Marquardt backpropagation algorithm [211] for training the three NNs, namely the MLP network, the Elman network and the NARX network, by the data collected shown in section 5.2. The performance of an NN can be evaluated by using the Mean Squared Error (MSE) and regression analysis. The MSE is the average squared difference between the estimate and the target outputs and the correlation coefficient R is used for the regression analysis. Through multiple trainings with different hyperparameter settings (only one hidden layer for all NNs, i.e. $l = 1$), the MSE and the regression R are summarised in Table 5.3. There are many rule-of-thumb methods for determining the appropriate number of hidden neurons. Underfitting occurs when there are too few neurons in the hidden layer to adequately detect the data. On the other hand, too many neurons in the hidden layer may result

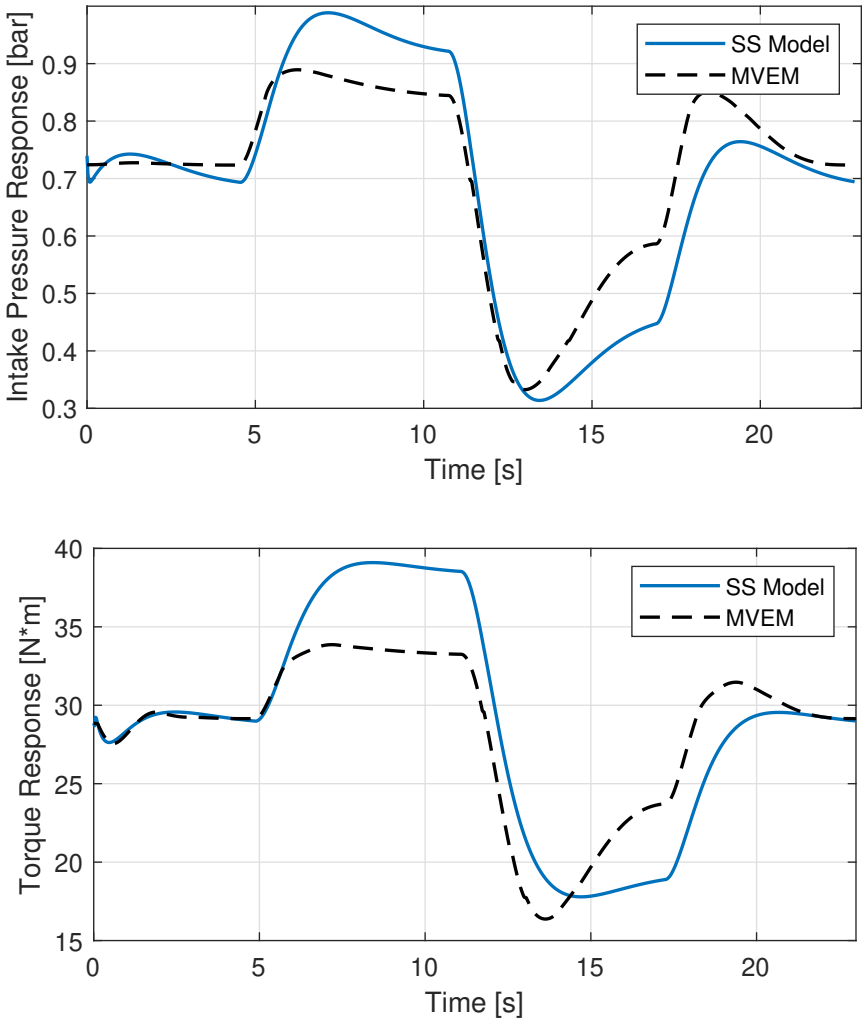

Figure 5.15: The intake manifold pressure and the torque responses of the linear SS model and the MVEM around the nominal operating point.

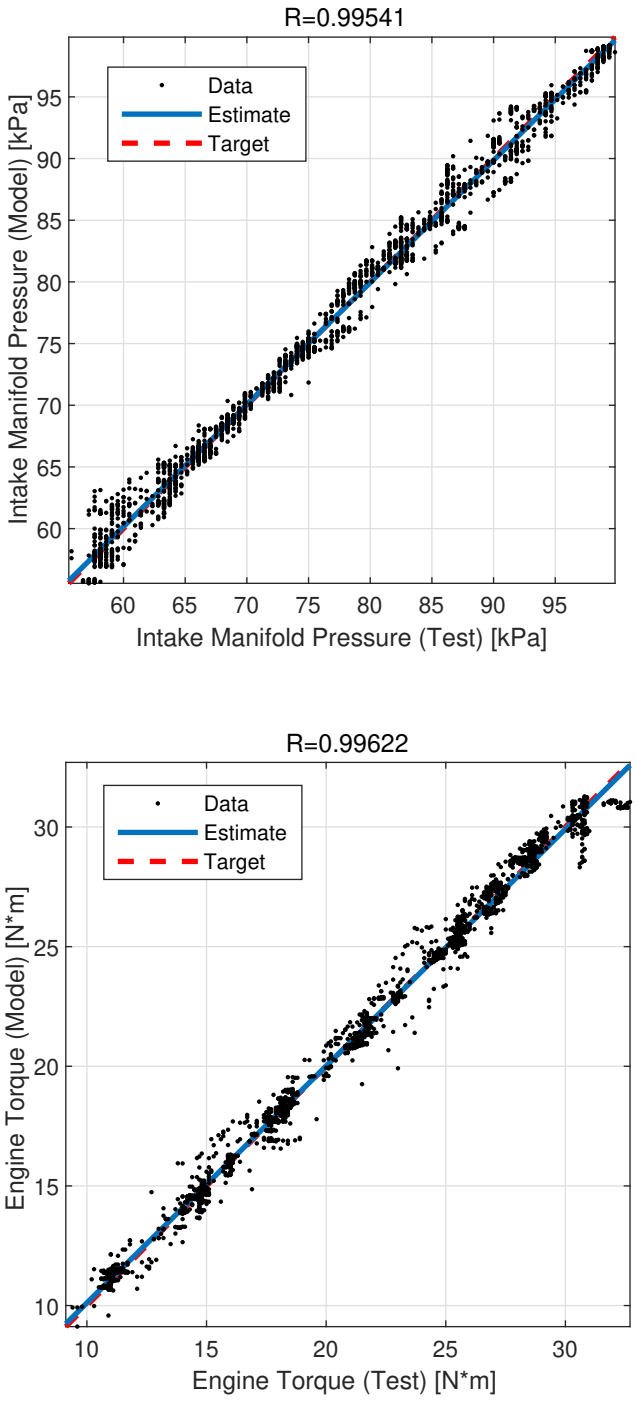

Figure 5.16: Regression analysis with respect to the intake manifold pressure and the engine torque for the MLP network with 3 neurons in the hidden layer.

Table 5.3: The MSE and the correlation coefficient R for the three types of NN.

NN	Output	# Hidden Neurons	# Time Delays	MSE	R
MLP	Intake Pressure	3	0	1.45144	0.99541
		7	0	0.21168	0.99930
		10	0	0.13156	0.99957
	Engine Torque	3	0	0.27029	0.99622
		7	0	0.15540	0.99782
		10	0	0.10385	0.99880
Elman	Intake Pressure	7	1	0.19126	0.99887
		7	2	0.18478	0.99938
		10	2	0.12929	0.99943
	Engine Torque	7	1	0.15738	0.99722
		7	2	0.11357	0.99850
		10	2	0.10174	0.99888
NARX	Intake Pressure	7	1;1	0.09476	0.99968
		7	2;2	0.08423	0.99978
		10	2;2	0.09077	0.99970
	Engine Torque	7	1;1	0.00440	0.99994
		7	2;2	0.00376	0.99995
		10	2;2	0.00369	0.99995

in overfitting and the increase of the training time. Some exemplar choices of the number of hidden neurons such as 3, 7, 10 are tested as shown in Table 5.3. Fig. 5.16 to 5.18 presents the regression analysis for the three NN with respect to the outputs, i.e. the intake manifold pressure and the engine torque. The comparative responses for the two outputs are shown in Fig. 5.19 and 5.20.

Overall, the three types of NN are all able to satisfactorily describe the engine dynamics, where the regression values R are all above 0.99. This result confirms the capability of the NN in nonlinear dynamic system modelling. In particular, the NARX network demonstrates to be the most accurate architecture for modelling the Wankel engine. The highest accuracy is achieved when the NARX network has 10 neurons in the hidden layer and 2 recurrent time delays. The torque estimation fits exceptionally

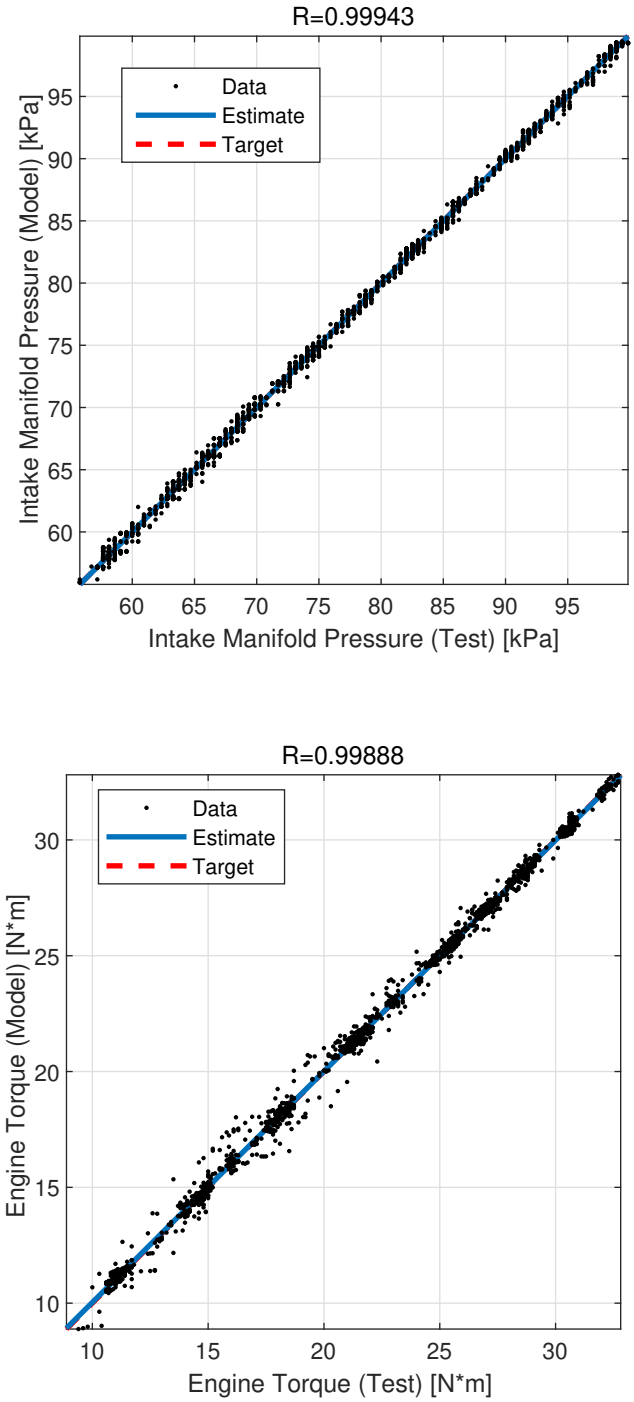

Figure 5.17: Regression analysis with respect to the intake manifold pressure and the engine torque for the Elman network with 10 neurons in the hidden layer and 2 time delays.

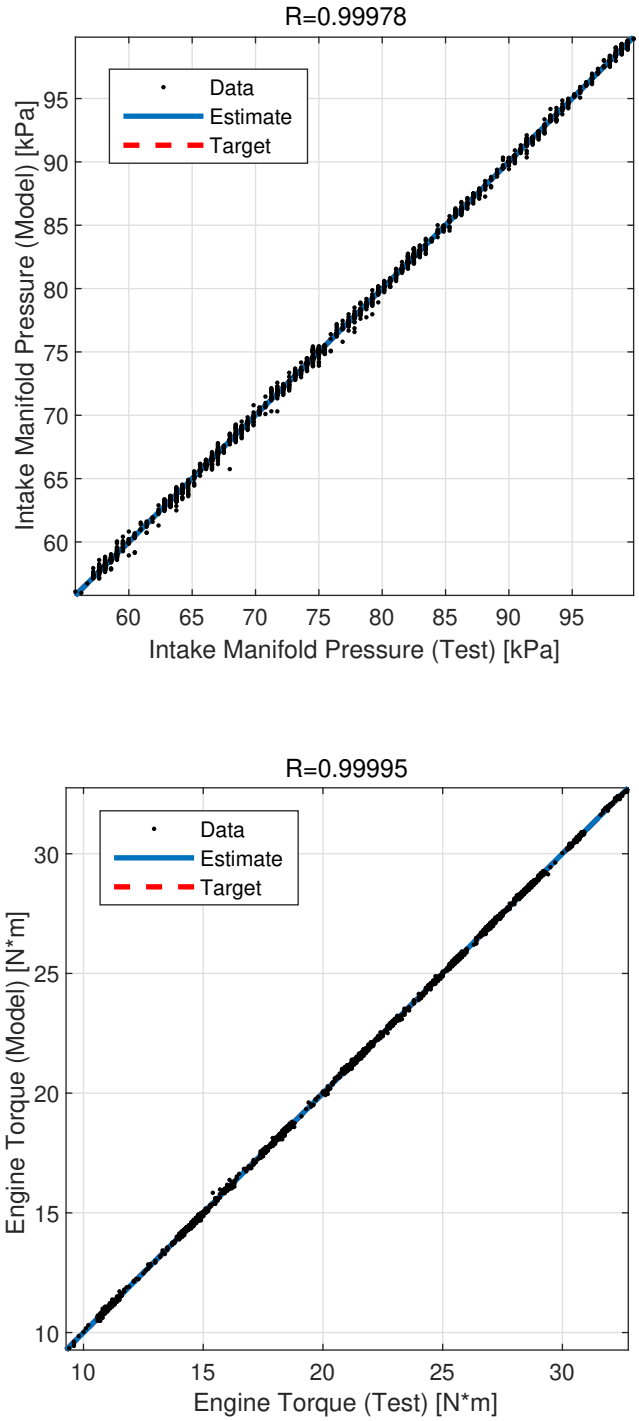

Figure 5.18: Regression analysis with respect to the intake manifold pressure and the engine torque for the NARX network with 7 neurons in the hidden layer and 2 time delays.

well with $R = 0.99995$. The Elman network can reach almost equal performance of the NARX network, especially when predicting the intake pressure. Compared to the two RNNs, the MLP network being a static mapping with only feedforward architecture struggles to achieve the equivalent accuracy and may not guarantee trustworthy dynamic response since no delays are used between the input and output. This is evident in Fig. 5.19, the intake manifold pressure response at around 3750s, where one finds that the MLP output transient is leading the target output transient. Such deviation from the realistic response can be ascribed to the lack of feedback mechanism in the MLP. However, it is noted that the training process for the MLP is faster than the other two RNNs with the same number of hidden neurons since there is no need computing for the time delays. It is interesting to notice that, for this engine modelling task, a network with a limited number of hidden neurons is less prone to overfit on the training data. However, a very low number of hidden neurons may lead to inferior modelling accuracy. Furthermore, increasing the number of hidden neurons introduces more complexity and does not guarantee the improvement of accuracy, especially for the RNN with more time delays. The performance is only marginally improved or even dropped slightly when the number of hidden neurons grows from 7 to 10 for both Elman and NARX network. The trade-off between the computational complexity and the model accuracy is needed when implementing the NN engine model in the control development process.

5.7 Model Synthesis and Conclusions

The three types of mathematical models presented in this paper are 1) the MVEM; 2) the SS model; and 3) the NN model. By using either (or both) physical knowledge or test data, these models are able to describe the Wankel engine dynamics with acceptable accuracy. They are all control-oriented models that have less computational demand and should be able to run faster than the available CFD models due to their simplicity.

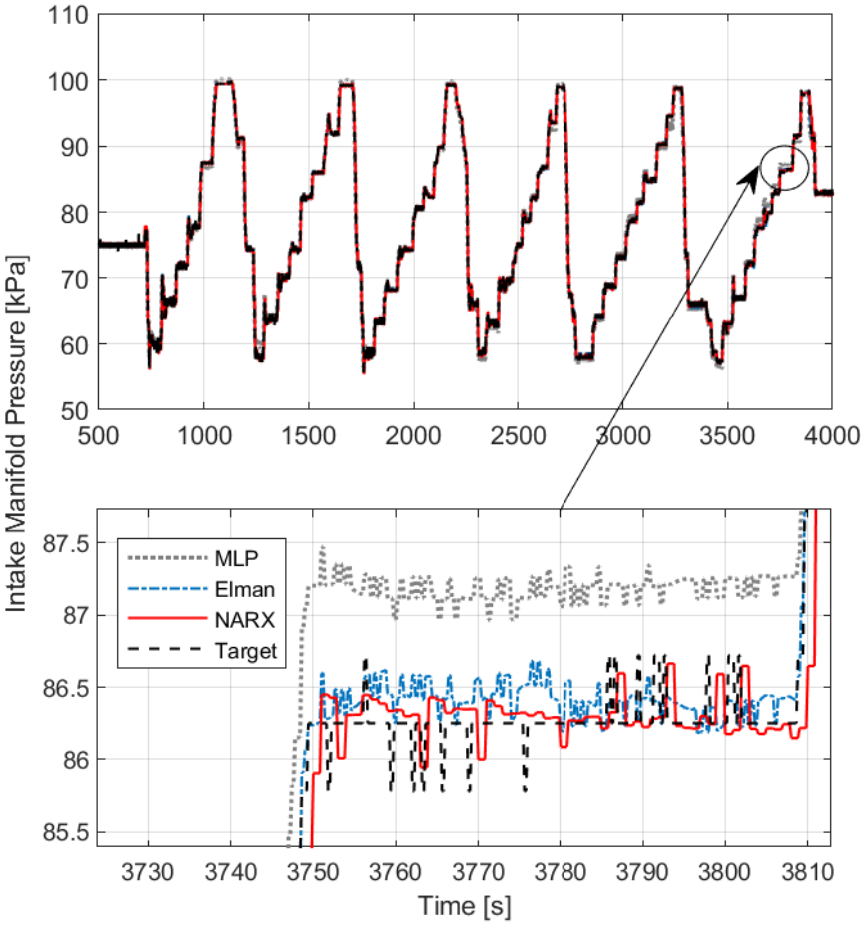

Figure 5.19: The intake manifold pressure responses for the MLP, Elman, and NARX networks compared with the measured engine test data.

For each model:

1) The MVEM model is the state-of-the-art approach for engine dynamics modelling and allows an in-depth study of engine physics and mechanisms. It consists of interconnected subsystems of the engine dynamics which are mostly in the form of nonlinear differential equations or empirical maps, which result in low-level fidelity but fast running speed.

2) The linear SS model characterises the engine dynamics as a set of first-order state

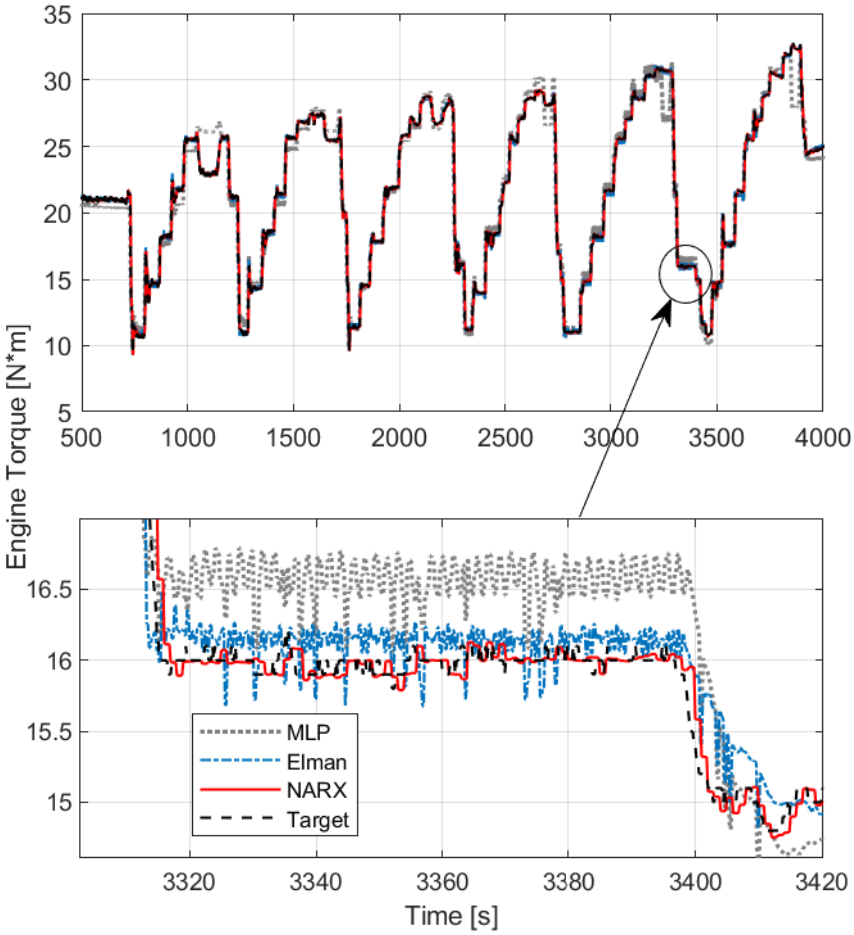

Figure 5.20: The torque responses for the MLP, Elman, and NARX networks compared with the measured engine test data.

equations. It has the simplest form yet the least accuracy since it is only valid in the neighbourhood around a nominal operating point. However, one can always merge together different controls designed for a number of nominal operating points via simple gain scheduling to design a global dynamic control. As a prominent characteristic of modern control theory, an SS model becomes very handy when designing an advanced control system, e.g., via optimal control, robust control, and intelligent control.

3) The NN models are essentially data-driven models via black-box system identifica-

tion. They are able to predict the dynamic behaviour of the Wankel engine by using the data collected from the engine tests and do not require *a priori* knowledge of the engine configuration or an understanding of underlying physics. Different network architecture can lead to different levels of performance, but the NN models overall tend to achieve higher accuracy than the MVEM and the SS model in terms of the regression analysis. In particular, the NARX network demonstrates the highest performance with acceptable complexity.

In the procedure of control development for the Wankel engine, one can use the SS model for the design phase of a control system for its simplicity and then use the NN model for the validation phase for its accuracy. Another benefit of the NN model can be derived by integrating it into the MVEM. The thermodynamic process of the combustion model is difficult to characterise and it relies on the static maps for the thermal efficiency in the MVEM (see (5.9)). Based on the results in this paper, one can easily create a NN model that outputs the engine torque using the measured input signals. This NN model could be used to replace the specific combustion model in the MVEM due to its better predicting capability. The focus of the next chapters will include idle speed control and air-fuel ratio control for the Wankel engine using the control-oriented model, specifically, the MVEM model, developed in this chapter. A novel adaptive optimal control strategy and its implementation on the ECU will be investigated in Chapter 6 and 8.

CHAPTER 6

OUTPUT FEEDBACK IDLE SPEED CONTROL VIA Q-LEARNING

Based on the dynamics modelling results from the previous chapter, we investigate the idle speed control problem for the Wankel rotary engine. The objective is develop an idle speed control system that is able to learn the optimal solution in real time using our adaptive optimal control. Since not all the engine variables are measurable using sensors, we integrate the controller with an extended Kalman filter (EKF) for the estimation of unknown variables, which results in an output feedback adaptive optimal idle speed control strategy.

6.1 Introduction

Idle speed control represents one of the most basic yet challenging automotive engine control problems, where the improvements in robustness and performance can directly result in better fuel economy, emissions, and drivability [212]. The idle speed needs to be regulated close to the set point. In a vehicle, the load disturbance due to the events such as power steering, transmission engagement, or low-speed manoeu-

vring may cause idle speed excursion. Advanced techniques that have been considered in the literature for idle speed control include H_∞ loop shaping [213], adaptive control [214], sliding mode control [215], and model predictive control [216].

Applying the idea of Q-learning and novel adaptive control techniques, in Chapter 3, a model-free adaptive optimal controller is derived in this thesis for unknown nonlinear input-affine systems. The controller performs in continuous time in a noniterative manner and online learns the optimal solution without *a priori* knowledge of system dynamics. However, from Chapter 3, the control design requires full state feedback, i.e., all the state variables need to be available or directly measurable. For practical problems, it is often the case that not all the states are measurable for the controller, where the output feedback control scheme becomes necessary. The options to solve this issue are to employ an observer or develop and use adaptive optimal output feedback control schemes. For instance, the paper of [217] presented the output feedback ADP for discrete-time linear systems and [218] developed one for continuous-time nonlinear systems using a sampled-data approach. In Chapter 4, we developed an optimal adaptive observer suitable to a nonlinear system. However, in this chapter, we have chosen an alternative optimal observer, an extended Kalman filter (EKF) to provide the optimal state observation needed for the adaptive optimal control scheme of Chapter 3. We demonstrate this for the idle speed control of a simulated engine system.

In this chapter, we design a novel adaptive optimal output feedback controller using continuous-time Q-learning and EKF for the idle speed regulation problem. The main contributions are summarised as follows.

1) Different from the existing approaches, the proposed reinforcement learning-based idle speed controller is completely model-free and able to learn the optimal solution (throttle profile) online in finite time using only the measurable outputs, namely, the intake manifold pressure, temperature, and the engine speed.

2) Reinforcement learning is applied to engine idle speed regulation, for the first time, in a continuous-time framework (in contrast to common iterative ADP algorithms).

3) The state feedback Q-learning combined with the EKF yields an optimal observer-based dynamic output feedback controller, of which the overall stability is guaranteed in the sense of Lyapunov.

We first present the engine model used for control design and validation and formulate the idle speed regulation into an infinite-horizon nonlinear optimal control problem. Then, we proposes a full state feedback controller based on continuous-time Q-learning and further extend the result to an output feedback controller using optimal observer techniques. Finally, we run the simulation of the proposed controller for the MVEM developed in the previous chapter.

6.2 Recap: Engine Model for Idle Speed Control

This section reviews the throttle body model and the fuel puddle model from the physics-based mean value engine model (MVEM) from Chapter 5 for the idle speed control problem.

6.2.1 Throttle Body Model

Assuming one-dimensional, steady, isentropic compressible flow of an ideal gas, the air mass flow rate $\dot{m}_{at}$ passing the throttle can be described as a linear function of the throttle angle α as

$$\dot{m}_{at} = K_\alpha \alpha \tag{6.1}$$

with the linearised flow rate sensitivity K_α ([128]). The throttle angle α is one of the major control inputs in the idle speed control system (while older engines often use an air bypass valve).

6.2.2 Fuel puddle model

The fuelling of the engine is controlled by a fuel injector fitted near the port on the intake manifold. Due to the port fuel injection (PFI) configuration, a fraction of the injected fuel is deposited on the manifold walls and becomes fuel puddles, which is refered to as the "wall-wetting" phenomenon. The final fuel flow rate $\dot{m}_f$ entering the housing is the sum of the fuel puddle flow rate $\dot{m}_{fpe}$ and the fuel vapour flow rate $\dot{m}_{fve}$ entering the combustion chamber

$$\dot{m}_f = \dot{m}_{fpe} + \dot{m}_{fve} = m_{fp}/\tau_p + m_{fv}/\tau_m \tag{6.2}$$

where τ_p and τ_m are the characteristic manifold time constants for the puddle m_{fp} and vapour m_{fv} fuel mass, respectively. Their dynamics can be taken as a set of two first-order processes with a time constant τ_f as

$$\begin{cases} \dot{m}_{fp} = \chi \dot{m}_{fi} - (1/\tau_f) m_{fp} - \dot{m}_{fpe} \\ \dot{m}_{fv} = (1-\chi)\dot{m}_{fi} + (1/\tau_f) m_{fp} - m_{fv}/\tau_m \end{cases} \tag{6.3}$$

where $\dot{m}_{fi}$ is the injected fuel flow rate (i.e., the control input in the air-fuel ratio control system) and χ ($0 \leq \chi < 1$) is the fraction of injected fuel that deposits on the manifold walls as fuel puddles (see [219] for more details).

6.3 State Feedback via Q-Learning

In this section, we propose an adaptive optimal state feedback controller for the idle speed regulation problem. The adaptive optimal control algorithm is based on our work ([143]) which extends the idea of Q-learning to completely unknown continuous-time nonlinear systems. The controller can online solve the nonlinear optimal control problems and is completely model-free.

6.3.1 Optimal Control Problem Formulation

The state variables can be chosen from the MVEM as

$$\mathcal{X}(t) = \begin{bmatrix} p_m(t) \\ T_m(t) \\ m_{fp}(t) \\ m_{fv}(t) \\ N(t) \end{bmatrix} \begin{matrix} \text{intake manifold pressure} \\ \text{intake manifold temperature} \\ \text{fuel puddle mass} \\ \text{fuel vapour mass} \\ \text{engine speed} \end{matrix} \tag{6.4}$$

For the idle speed control problem, the objective is to regulate the engine speed $N(t)$ around a certain low set point, namely, $N^0 = 3000$ (RPM) for the Wankel rotary engine. We can shift the coordinate of the equilibrium point to zero by translating the engine speed as $\mathcal{N}(t) = N(t) - N^0$. Similarly, by translating the other states to a nominal operating point, a new state vector x is defined such that

$$x(t) = \mathcal{X}(t) - \mathcal{X}^0 \tag{6.5}$$

where $\mathcal{X}^0$ is the nominal operating point around which the engine idles as shown in Table 6.1. By inspection of (6.1) and the MVEM, the throttle angle α is an affine input of the whole system. We can represent the MVEM as a continuous-time nonlinear time-invariant system in state space as

$$\dot{x}(t) = f(x(t)) + g(x(t))\alpha(t), \quad x(0) = x_0 \tag{6.6}$$

where the state vector $x(t)$ is defined as (6.4), the throttle angle $\alpha(t)$ is the control policy, and $f(x(t))$, $g(x(t))$ are the system drift and the input gain functions, respectively. It is reasonable to assume that the engine dynamics $f(x) + g(x)\alpha$ is Lipschitz continuous on a compact set $\Omega \in \mathbb{R}^5$ that contains the origin.

We define the infinite-horizon integral cost

$$V(x(t)) := \int_t^\infty r(x(\tau), \alpha(\tau)) \mathrm{d}\tau \tag{6.7}$$

with the utility $r(t) = S(x(t)) + R\alpha^2(t)$. The utility $r(t)$ is positive definite, i.e., $S(x(t)) > 0$ and $R > 0$ for $x \neq 0$. One can simply choose a quadratic utility term $S(x(t)) = x^\top S^0 x$ with a positive definite matrix $S^0 > 0$ for the idle speed regulation problem.

Table 6.1: Nominal idle operating point.

Engine state	Symbol	Value	Units
Intake manifold pressure	p_m^0	0.8	bar
Intake manifold temperature	T_m^0	25	^{o}C
Fuel puddle mass	m_{fp}^0	0.11	g
Fuel vapour mass	m_{fv}^0	0.25	g
Engine speed	N^0	3000	RPM

The optimal control problem is to minimise the value function (6.7) by choosing the optimal stabilising (admissible) control policy $\alpha^*(t)$. The optimal value function $V^*(x)$ can be determined as

$$V^*(x(t)) := \min_{\alpha} \int_t^{\infty} r(x(\tau), \alpha(\tau)) \mathrm{d}\tau \tag{6.8}$$

A general solution to the nonlinear optimal control problem can be formulated as a partial differential equation for the optimal value function $V^*(x)$. We define the Hamiltonian of the problem as

$$\mathcal{H}(x, \alpha, \nabla V_x) := r(x, \alpha) + (\nabla V_x)^{\mathsf{T}} (f(x) + g(x)\alpha) \tag{6.9}$$

with the gradient vector $\nabla V_x = \partial V / \partial x$. The optimal value function $V^*(x)$ in (6.8) satisfies the *Hamilton-Jacobi-Bellman* (HJB) equation

$$0 = \min_{\alpha} \mathcal{H}(x, \alpha, \nabla V_x^*) \tag{6.10}$$

It is noted that the throttle angle α should be in the range of $[0, 90^o]$. However, the idle speed control problem usually entails a local span of the operating condition near the low speed where the throttle angle is small, e.g., $\alpha \approx 30^o$, when all the states are around the nominal operating point in Table 6.1. The control action α is far below the upper limit and is therefore locally unconstrained. Therefore, the optimal control α^* can be found by setting $\partial \mathcal{H}(x, u, \nabla V_x^*) / \partial \alpha = 0$ so that

$$\alpha^* = -\frac{1}{2} R^{-1} g(x)^{\mathsf{T}} \nabla V_x^* \tag{6.11}$$

Inserting the optimal control (6.11) into (6.10) gives the HJB equation in terms of ∇V_x^* as

$$0 = S(x) + (\nabla V_x^*)^\top f(x) - \frac{1}{4}(\nabla V_x^*)^\top g(x) R^{-1} g(x)^\top \nabla V_x^* \tag{6.12}$$

In general, the HJB equation (6.9) is difficult to solve due to its nonlinearity and the requisite for *a priori* knowing the system drift dynamics $f(x)$ and input gain dynamics $g(x)$.

6.3.2 Parameterisation of Nonlinear Q-function

Similar to Chapter 4, an action-dependent version of value function $Q(x, \alpha)$ is to be created, such that $Q^*(x, \alpha^*) = V^*(x)$. For the continuous-time nonlinear input-affine system (6.6), the Q-function can be explicitly defined by adding the Hamiltonian (6.9) onto the optimal value (6.8) as

$$\begin{aligned} Q(x, \alpha) &:= V^*(x) + \mathcal{H}(x, \alpha, \nabla V_x^*) \\ &= \underbrace{V^*(x) + S(x) + (\nabla V_x^*)^\top f(x)}_{F_{xx}(x)} + \\ &\underbrace{(\nabla V_x^*)^\top g(x)\alpha}_{F_{x\alpha}(x,\alpha)} + \underbrace{R\alpha^2}_{F_{\alpha\alpha}(\alpha)} \end{aligned} \tag{6.13}$$

where $F_{xx}(x)$, $F_{x\alpha}(x, \alpha)$, and $F_{\alpha\alpha}(\alpha)$ are the lumped terms that can be approximated respectively via neural networks. Similar to Chapter 4, the Q-function is of the same value as the value function V() for the optimal control:

Lemma 6.1 *The Q-function defined in* (6.13) *is positive definite with the optimisation scheme* $Q^*(x, \alpha^*) = \min_\alpha Q(x, \alpha)$. *The optimal Q-function* $Q^*(x, \alpha^*)$ *has the same optimal value* $V^*(x)$ (6.8) *as for the value function* $V^\alpha(x)$ (6.7), *i.e.* $Q^*(x, \alpha^*) = V^*(x)$ *when applying the optimal control* α^*. ◇

Proof. See *Lemma 3.3* in Chapter 3 for detailed proof. □

6.3.3 Adaptive Critic for Q-function Approximation

We approximate the Q-function (6.13) using a critic neural network by

$$Q(x,\alpha) = W^\top \Phi(x,\alpha) + \varepsilon_Q(x,\alpha) + \varepsilon_x(x,t) \tag{6.14}$$

where $\Phi(x,\alpha) \in \mathbb{R}^n$ denotes the activation function vector with the number n of neurons in the hidden layer; $W \in \mathbb{R}^n$ is the weight vector, $\varepsilon_x(x,t)$ is some bounded error due to sensor noise or an observer; $\varepsilon_Q(x,\alpha)$ is the neural network approximation error; and $W^\top \Phi(x,\alpha)$ can be explicitly expressed according to the three components $F_{xx}(x)$, $F_{x\alpha}(x,\alpha)$, and $F_{\alpha\alpha}(\alpha)$ in (6.13) as

$$W^\top \Phi(x,\alpha) = \begin{bmatrix} W_{xx}^\top & W_{x\alpha}^\top & W_{\alpha\alpha}^\top \end{bmatrix} \begin{bmatrix} \Phi_{xx}(x) \\ \Phi_{x\alpha}(x)\alpha \\ \Phi_{\alpha\alpha}(\alpha) \end{bmatrix} \tag{6.15}$$

where $\Phi_{xx} \in \mathbb{R}^{n_{xx}}$, $\Phi_{x\alpha} \in \mathbb{R}^{n_{x\alpha}}$ and $\Phi_{\alpha\alpha} = \alpha^2$. The regressor $\Phi(x,\alpha)$ is selected to provide a complete independent basis such that $Q(x,\alpha)$ is uniformly bounded with $n = n_{xx} + n_{x\alpha} + 1$. Recall from the Weierstrass higher-order approximation theorem ([164]), the approximation error $\varepsilon_Q(x,\alpha)$ is bounded for a fixed n within a compact set Ω and as the number of neurons $N_{xx} \to \infty$ and $N_{x\alpha} \to \infty$, i.e., $n \to \infty$, we have $\varepsilon_Q(x,\alpha) \to 0$. The error term $\varepsilon_x(x,t)$ considers the effect of introducing the optimal observer states in $Q(x,\alpha)$ and in the control policy. The analysis of the output feedback control using an optimal observer will be explained later in Section 6.4.

One needs to derive the Bellman equation in terms of the Q-function to update the critic. By Bellman's principle of optimality ([41]), we have the following optimality equation

$$V^*(x(t-T)) = \int_{t-T}^{t} r(x(\tau),\alpha(\tau))\mathrm{d}\tau + V^*(x(t)) \tag{6.16}$$

The result from *Lemma 6.1* showed that $Q^*(x,\alpha^*) = V^*(x)$, which means we can rewrite (6.16) in terms of $Q^*(x,\alpha^*)$ as

$$
\begin{aligned}
&\overbrace{-\int_{t-T}^{t} r(x,\alpha)\mathrm{d}\tau}^{-\rho(x,\alpha)} = Q^*(x(t),\alpha^*(t)) \\
&\qquad\qquad - Q^*(x(t-T),\alpha^*(t-T)) \\
&= \underbrace{W^\top \Phi(x(t),\alpha^*(t)) - W^\top \Phi(x(t-T),\alpha^*(t-T))}_{W^\top \Delta\Phi(x,\alpha^*)} \\
&\quad + \varepsilon_{BQ} + \varepsilon_{Bx}
\end{aligned} \tag{6.17}
$$

with the integral reinforcement $\rho(x,\alpha)$, the difference $\Delta\Phi(t) = \Phi(x(t),\alpha^*(t)) - \Phi(x(t-T),\alpha^*(t-T))$, and the Bellman equation residual errors $\varepsilon_{BQ} = \varepsilon_Q(x(t),\alpha^*(t)) - \varepsilon_Q(x(t-T),\alpha^*(t-T))$ and $\varepsilon_{Bx} = \varepsilon_x(x(t)) - \varepsilon_x(x(t-T))$ being bounded for bounded ε_Q and ε_x. Define two auxiliary variables $\mathcal{P} \in \mathbb{R}^{n\times n}$ and $\mathcal{Q} \in \mathbb{R}^n$ by low-pass filtering the variables in (6.17) as

$$
\begin{cases}
\dot{\mathcal{P}} = -\ell\mathcal{P} + \Delta\Phi(t)\Delta\Phi(t)^\top, \quad \mathcal{P}(0) = 0 \\
\dot{\mathcal{Q}} = -\ell\mathcal{Q} + \Delta\Phi(t)\rho(x,\alpha), \quad \mathcal{Q}(0) = 0
\end{cases} \tag{6.18}
$$

with a filter parameter $\ell > 0$.

The adaptive critic neural network can be written as

$$
\hat{Q}(x,\alpha) = \hat{W}^\top \Phi(x,\alpha) \tag{6.19}
$$

where $\hat{W}$ and $\hat{Q}(x,\alpha)$ denote the current estimate of W and $Q(x,\alpha)$, respectively.

Now we design the adaptation law using the sliding mode technique to update $\hat{W}$ such that

$$
\dot{\hat{W}} = -\Gamma\mathcal{P}\frac{M}{||M||} \tag{6.20}
$$

where $M \in \mathbb{R}^n$ is defined as $M = \mathcal{P}\hat{W} + \mathcal{Q}$ and $\Gamma > 0$ is a diagonal adaptive learning gain to be tuned.

Lemma 6.2 *Given the adaptation law* (6.20)*, if* $\alpha(t)$*,* $\Delta\Phi(t)$*, and the system states* $x(t)$ *are persistently excited, the estimation error of weight* $\tilde{W} = W - \hat{W}$ *will converge to a compact set in finite time.* ◇

Proof. The proof follows similarly from *Lemma 3.4* in Chapter 3. It is shown that M can be expressed in terms of the weight error $\tilde{W}$ as $M = -\mathcal{P}\tilde{W} + \Lambda$ with the residual error $\Lambda = \int_0^t e^{-\ell(t-\tau)}\Delta\Phi(\tau)(\varepsilon_{BQ}(\tau) + \varepsilon_{Bx}(\tau))\mathrm{d}\tau$. The weight convergence is demonstrated by choosing a proper Lyapunov function

$$\mathcal{L}_1 = \frac{1}{2}(\mathcal{P}^{-1}M)^\top \Gamma^{-1}\mathcal{P}^{-1}M \tag{6.21}$$

and its time derivative $\dot{\mathcal{L}}_1 \leq -a||M||$ with a constant $a > 0$. □

6.3.4 Adaptive Optimal Control via Q-learning

We reconstruct the optimal control α^* from (6.11) based on the parameterisation of $Q(x, \alpha)$ (6.13) such that

$$\alpha^* = -\frac{1}{2}W_{\alpha\alpha}^{-1}W_{x\alpha}^\top \Phi_{x\alpha}(x) + \varepsilon_{Q\alpha} + \varepsilon_{x\alpha} \tag{6.22}$$

where $\varepsilon_{Q\alpha}$ and $\varepsilon_{x\alpha}$ are bounded approximation errors due to ε_Q and ε_x, $W_{x\alpha}^\top \Phi_{x\alpha}(x)$ accounts for the term $g(x)^\top \nabla V_x^*$, and $W_{\alpha\alpha}$ is essentially predefined R (see (6.13)). Although the value of R is available through the value function (6.7), we shall write the actor in the form of (6.23) for the sake of theoretical consistency. In practice, the initial weights of $W_{\alpha\alpha}$ can be chosen either randomly or as the same values in R. Therefore, one can determine the optimal control directly using the adaptive critic (6.19) if the weight $\hat{W}$ converges to the actual weight W. The control law (actor) will be

$$\alpha = -\frac{1}{2}\hat{W}_{\alpha\alpha}^{-1}\hat{W}_{x\alpha}^\top \Phi_{x\alpha}(x) \tag{6.23}$$

We summarise the result for this Q-learning algorithm as

Theorem 6.1 *Given the engine system* (6.6) *with the value function* (6.7) *and Q-function* (6.13)*, the adaptive critic neural network* (6.19) *with the adaptation law* (6.20) *and the actor*

(6.23) *form an adaptive optimal control so that the adaptive critic weight estimation error* $\tilde{W}$ *will converge to a compact set and the throttle angle (the actor)* α *will converge to a small bounded set around its optimal control solution* α^* *in finite time.*

Proof. The proof follows similarly from *Theorem 3.2* in Chapter 3. □

The proposed idle speed controller is a completely *model-free* algorithm that can approximately solve the optimal control problem *online* without the *a priori* knowledge of the system drift $f(x)$ and input gain $g(x)$. The finite-time convergence of the critic neural network weight is guaranteed. However, the control algorithm is data-driven and requires the complete knowledge of all the state variables $x(t)$.

6.4 Output Feedback Idle Speed Control

This section describes the design and analysis of the output feedback control for the idle speed regulation problem, which combines a reduced-order optimal observer with the previous Q-learning-based state feedback control.

6.4.1 Extended Kalman Filter

It is uncommon to directly use state feedback for a realistic system since not all the states are measurable in practice. By inspection of the engine states (6.4), the intake manifold pressure p_m, temperature T_m, and the engine speed N are commonly measurable through the manifold absolute pressure (MAP) sensor, the intake air temperature (IAT) sensor, and the tachometer, respectively. The other two states: the fuel puddle mass m_{fp} and the fuel vapour mass m_{fv} are not directly measurable in practice. If the system is observerable, one can design a reduced-order observer to online estimate only the unknown states. Kalman filters have been widely used as a linear quadratic estimation algorithm, which can be further extended to deal with the nonlinearities [220] and unknown parameters [221]. We design an extended Kalman filter (EKF) to estimate the fuel puddle model using the mass air flow (MAF) sensor and the

Lambda sensor (see also [207] for more details). This will be integrated as part of an adaptive optimal idle speed controller. Since only the fuel puddle variables m_{fp} and m_{fv} need to be estimated, we can use the EKF as a reduced-order optimal observer for our Q-learning-based state feedback idle speed controller. The EKF is a common type of estimator that is known for its robustness and stable performance. At the initial stage of the research, the EKF is a natural choice that is sufficient for the fuel puddles problem. One can also use the adaptive optimal control proposed in the later chapter for the same problem.

By inspection of the nonlinear fuel puddle model (6.2)(6.3) with the unknown parameters τ_f and χ, the parameters can be taken as extra states to be estimated. Moreover, it has been shown in [219][206] that the term $\dot{m}_{fpe}$ is negligible. Hence, the fuel puddle process (6.2)(6.3) can be written as

$$\begin{cases} \dot{\tau_f} = w_1 \\ \dot{\chi} = w_2 \\ \dot{m}_{fp} = \chi \dot{m}_{fi} - (1/\tau_f) m_{fp} + w_3 \\ \dot{m}_{fv} = (1-\chi)\dot{m}_{fi} + (1/\tau_f) m_{fp} - m_{fv}/\tau_m + w_4 \\ \dot{m}_f = m_{fv}/\tau_m + v \end{cases} \tag{6.24}$$

or in the form of augmented state equations as

$$\begin{cases} \dot{y} = f(y,u) + w \\ \dot{m}_f = h(y) + v \end{cases} \tag{6.25}$$

where $y = [\tau_f \ \chi \ m_{fp} \ m_{fv}]^T$ is the augmented state vector, $u = \dot{m}_{fi}$ is the system input (injected fuelling command), $z = \dot{m}_f$ is the measurement of the system output, $f(y,u)$ and $h(y)$ denotes the (non-)linear functions in (6.24). Practically, the measurement z can be obtained through dividing the reading of a mass air flow (MAF) sensor by the reading of a lambda sensor since $\dot{m}_f = \dot{m}_a/\lambda$. Moreover, $w = [w_1 \ w_2 \ w_3 \ w_4]^T \sim \mathcal{N}(0, \mathcal{W})$ and $v \sim \mathcal{N}(0, \mathcal{Y})$ are the zero mean multivariate Gaussian noises that account for the model inaccuracy and sensor noise with pre-defined covariance $\mathcal{W}$ and $\mathcal{Y}$.

Assumption 6.1 *Assume that the process noise w and the measurement noise v are bounded, i.e. $||w|| \leq \varpi$ and $|v| \leq \mu$ with $\varpi > 0$ and $\mu > 0$.*

It is not uncommon in practice that the noise is assumed to be truncated Gaussian so not only the bound but the distribution inside the bound is available. Similar assumptions can be found in [222][223][224] where the probabilistic and hard bound approaches are combined for set-membership identification.

For the system (6.25), an extended Kalman filter can be designed accordingly with the Kalman gain vector K as

$$\dot{\hat{y}} = f(\hat{y}, u) + K(z - h(\hat{y})) \tag{6.26}$$

where $\hat{y}$ is the estimate of the state vector y and K is the adaptive Kalman gain to be designed later.

From (6.25) and (6.26), the estimation error is defined as $\tilde{y} = y - \hat{y}$ and its derivative can be written as

$$\begin{aligned} \dot{\tilde{y}} &= \dot{y} - \dot{\hat{y}} \\ &= f(y,u) - f(\hat{y},u) - K(h(y) - h(\hat{y})) + w - Kv \end{aligned} \tag{6.27}$$

Since f and h are differentiable, the error dynamics can then be linearised around $x, \hat{x}$ such that

$$\dot{\tilde{y}} = (F - KH)\tilde{y} + o(||\tilde{y}||) + w - Kv \tag{6.28}$$

where $o(||\tilde{y}||)$ denotes the higher order terms of the approximation error, which have an upper bound $\delta > 0$. F and H are the Jacobian matrices of $f(y,u)$ and $h(y)$ with respect to y as

$$F = \frac{\partial f}{\partial y} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ m_{fp}/\tau_f^2 & u & -1/\tau_f & 0 \\ -m_{fp}/\tau_f^2 & -u & 1/\tau_f & -1/\tau_m \end{bmatrix} \tag{6.29}$$

$$H = \frac{\partial h}{\partial y} = \begin{bmatrix} 0 & 0 & 0 & 1/\tau_m \end{bmatrix} \tag{6.30}$$

The Kalman gain K can be online updated by solving the covariance prediction matrix P in the algebraic Riccati equation

$$\dot{P} = FP + PF^T - KHP + \mathcal{W} \tag{6.31}$$

such that

$$K = PH^T \mathcal{Y} - 1 \tag{6.32}$$

It can be proved that the solution P is bounded and positive definite via Theorem 3.4 in *Optimal Control* [30].

Theorem 6.2 For the augmented system (6.25) with the extended Kalman filter (6.26), the estimation error $\tilde{y}$ will exponentially converge towards a compact set around zero and thus $\hat{y} \to y$ holds provided that the noise/error bounds $\delta \to 0$, $\varpi \to 0$, $\mu \to 0$.

Proof: Defining the inverse of the positive definite matrix P as $Y = P^{-1}$, the algebraic Riccati equation (6.31) can be transformed into

$$-\dot{Y} = YF + F^T Y - YKH + Y\mathcal{W}Y \tag{6.33}$$

Then a Lyapunov function can be chosen as

$$V_k(t) = \frac{1}{2}\tilde{y}^T Y \tilde{y} \tag{6.34}$$

Its derivative can be calculated using (6.33) as

$$\begin{aligned} \dot{V}_k(t) &= \frac{1}{2}\dot{\tilde{y}}^T Y \tilde{y} + \frac{1}{2}\tilde{y}^T \dot{Y} \tilde{y} + \frac{1}{2}\tilde{y}^T Y \dot{\tilde{y}} \\ &= -\frac{1}{2}\tilde{y}^T (\dot{Y} + Y\mathcal{W}Y + YKH)\tilde{y} + \frac{1}{2}\tilde{y}^T \dot{Y} \tilde{y} \\ &\quad + \tilde{y}^T Y o(||\tilde{y}||) + \tilde{y}^T Y w - \tilde{y}^T YKv \\ &\leq -\frac{1}{2}\lambda_{\min}(Y\mathcal{W}Y + YKH)||\tilde{y}||^2 \\ &\quad + \frac{\lambda^2_{\max}(Y)}{\eta}||\tilde{y}||^2 + \frac{\lambda^2_{\max}(YK)}{2\eta}||\tilde{y}||^2 \\ &\quad + \frac{\eta}{2}\delta^2 + \frac{\eta}{2}\varpi^2 + \frac{\eta}{2}\mu^2 \\ &\leq -\alpha_1 V_k(t) + \beta \end{aligned} \tag{6.35}$$

where $\lambda_{\min}(\bullet)$, $\lambda_{\max}(\bullet)$ denote the minimum, maximum eigenvalues of a matrix, $\alpha_1 = \lambda_{\min}(Y\mathcal{W}Y + YKH)/\lambda_{\max}(Y) - [2\lambda^2_{\max}(Y) + \lambda^2_{\max}(YK)]/2\eta\lambda_{\max}(Y)$ and $\beta = \eta(\delta^2 + \varpi^2 + \mu^2)/2$ are positive constants for a properly chosen constant $\eta > [2\lambda^2_{\max}(Y) + \lambda^2_{\max}(YK)]/2$ $\lambda_{\min}(Y\mathcal{W}Y + YKH)$. This implies that $V_k(t) \leq V(0)e^{-\alpha_1 t} + \beta/\alpha_1$ holds and the estimation error $\tilde{x}$ will exponentially converge towards a compact set defined by $\Omega_1 := \{\tilde{x} \mid ||\tilde{x}|| \leq \sqrt{\eta(\delta^2 + \varpi^2 + \mu^2)/\alpha_1\lambda_{\min}(Y)}\}$. Clearly, $\lim_{t\to\infty} \tilde{y} = 0$ holds for $\beta \to 0$, i.e. $\delta \to 0$, $\varpi \to 0$, $\mu \to 0$. ◇

Hence, the fuel puddle dynamics as well as the unknown parameters τ_f and χ can be online estimated via extended Kalman filter (6.26) using only MAF and lambda sensors.

6.4.2 Output Feedback Synthesis

The proposed Q-learning-based state feedback idle speed controller (*Theorem 6.1*) requires the complete knowledge of the system states. A reduced-order optimal state observer (*Theorem 6.2*) is designed to estimate the unknown states. The combination of the two naturally leads to an adaptive optimal output feedback idle speed controller. Fig. 6.1 presents a schematic diagram of the proposed Q-learning-based idle speed control system. We summarise the main result of this chapter in the following theorem.

Theorem 6.3 *Given the engine system* (6.6) *with the stochastic fuel puddle process* (6.25) *and the prescribed value function* (6.7) *and Q-function* (6.13)*, the adaptive critic neural network* (6.19) *with the adaptation law* (6.20) *and the actor* (6.23) *and the EKF* (6.26) *form an adaptive optimal output feedback control so that the throttle angle α will converge to a small bounded set near its optimal control solution α^* in finite time, i.e., the idle speed will be regulated near the set point with minimum control effort subject to the prescribed value function* (6.7)*.*

Proof. Since the employed state $\hat{x}$ for estimating the Q-function is partially observed from the EKF (6.26), i.e., the states x_3, x_4 are obtained as $\hat{y}_3, \hat{y}_4$, the Q-functions $Q(x, \alpha(x))$,

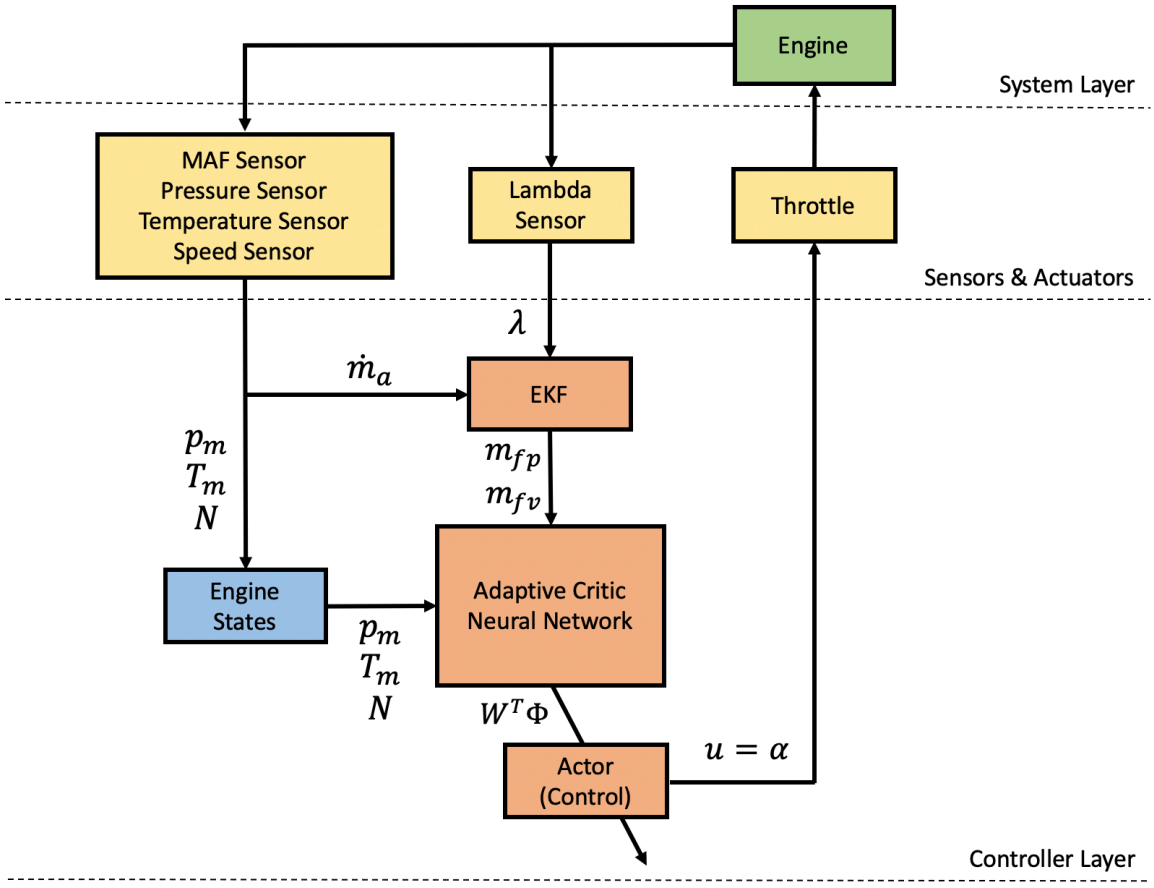

FIGURE 6.1. A schematic diagram of the proposed output feedback Q-learning-based idle speed control system.

$Q(\hat{x}, \alpha(\hat{x}))$ are continuous in x, $\hat{x}$ as they are physical states of the engine, we can explicitly write

$$\begin{aligned} Q(\hat{x}, \alpha(\hat{x})) &= Q(x, \alpha(x)) + (Q(\hat{x}, \alpha(\hat{x})) - Q(x, \alpha(x))) \\ &= Q(x, \alpha(x)) + \varepsilon_x(x, t) \end{aligned} \tag{6.36}$$

Then we have $Q^*(\hat{x}, \alpha(\hat{x})) = Q^*(x, \alpha(x)) + \varepsilon_x(x, t)$, where the bounded error $\varepsilon_x(x, t)$ comprises the effect of introducing the optimal observer into the output feedback control. One can use $Q^*(x, \alpha)$ in a Lyapunov function given the exponential convergence of the EKF (6.26), i.e. $\varepsilon_x(x, t) \to 0$ as $t \to \infty$.

We design the overall Lyapunov function with respect to the EKF estimation error $\tilde{y}$, the optimal integral cost Q^*, the adaptive critic weight estimation error $\mathcal{L}_1$ (6.21), and the neural network residual error Λ (see the proof of *Lemma 6.2*) as

$$\mathcal{L} = \frac{k_1}{2}\tilde{y}^\mathsf{T} P^{-1}\tilde{y} + k_2 Q^*(x,\alpha) + k_3\mathcal{L}_1 + \frac{k_4}{2}\Lambda^\mathsf{T}\Lambda \tag{6.37}$$

with positive constants k_1, k_2, k_3, and k_4.

We investigate the Lyapunov function $\mathcal{L}$ in a compact set $\tilde{\Omega} \subset \mathbb{R}^4 \times \mathbb{R}^n \times \mathbb{R}^5 \times \mathbb{R} \times \mathbb{R}^n$ in tuple $(\tilde{y}, M, x, \alpha, \Lambda)$ that contains the origin and $\tilde{\Omega} \subset \Omega$. Any initial value of $(\tilde{y}, M, x, \alpha, \Lambda)$ is assumed to be within the interior $\tilde{\Omega}$. Thus, for any initial trajectory, the state x and the control α remain bounded for at least finite time $t \in [0, T]$. Based on (6.13), differentiating the term $k_2 Q^*(x,\alpha)$ in (6.37) will involve $\dot{Q}^*(x,\alpha) = \dot{V}^* + \dot{\mathcal{H}}(x,\alpha,\nabla V_x^*)$. For the Lagrange multiplier $\lambda = \nabla V_x^*$, differentiating the Hamiltonian gives

$$\dot{\mathcal{H}}(x,\alpha,\nabla V_x^*) = \partial\mathcal{H}/\partial t + (\nabla\mathcal{H}_\alpha)^\mathsf{T}\dot{\alpha} + (\nabla\mathcal{H}_x + \dot{\lambda})^\mathsf{T}\dot{x} \tag{6.38}$$

According to Lagrange's theory ([30]), from the costate equation and stationarity condition, the derivative of the Lagrange multiplier λ satisfies $\dot{\lambda} = -\nabla\mathcal{H}_x$ and $\nabla\mathcal{H}_\alpha = 0$. For a time-invariant system (6.6) and value function (6.7), the Hamiltonian $\mathcal{H}(x,\alpha,\nabla V_x^*)$ is not an explicit function of t, i.e. $\dot{\mathcal{H}} = \partial\mathcal{H}/\partial t = 0$. We have proved exponential convergence of the optimal observer in *Theorem 6.2*, the control action based on the partially-estimated states $\hat{x}$ (where x_3, x_4 are replaced by $\hat{y}_3, \hat{y}_4$) will result in an exponentially bounded error of the optimal control, which is contained in $\varepsilon_{x\alpha}$ in (6.22), i.e., $\alpha^* = \alpha + \varepsilon_{Q\alpha} + \varepsilon_{x\alpha}$. Hence, inserting the Hamiltonian $\mathcal{H}$ (6.9), the derivative of the value function is given as

$$\begin{aligned}\dot{V}^* &= (\nabla V_x^*)^\mathsf{T}(f + g\alpha^* + g\varepsilon_{Q\alpha} + g\varepsilon_{B\alpha}) \\ &= -r(x,\alpha^*) + (\nabla V_x^*)^\mathsf{T} g(\varepsilon_{Q\alpha} + \varepsilon_{B\alpha})\end{aligned} \tag{6.39}$$

The time derivative of $\mathcal{L}$ can be derived as

$$
\begin{aligned}
\dot{\mathcal{L}} &= k_1 \tilde{y}^T P^{-1} \dot{\tilde{y}} + \frac{k_1}{2} \tilde{y}^T P^{-1} \dot{P} P^{-1} \tilde{y} + k_2 [\dot{V}^* + \dot{\mathcal{H}}] + k_3 \dot{\mathcal{L}}_1 + k_4 \Lambda^\top \dot{\Lambda} \\
&= -k_1 \tilde{y}^T (P^{-1} \mathcal{W} P^{-1} + H^\top \mathcal{Y}^{-1} H) \tilde{y} \\
&\quad + k_1 \tilde{y}^T P^{-1} o(||\tilde{y}||) + k_1 \tilde{y}^T P^{-1} w - k_1 \tilde{y}^T H^\top \mathcal{Y}^{-1} v \\
&\quad + k_2 (-r(x, \alpha^*) + (\nabla V_x^*)^\top g(\varepsilon_{Q\alpha} + \varepsilon_{x\alpha})) + k_3 \dot{\mathcal{L}}_1 + k_4 \Lambda^\top (-\ell \Lambda + \Delta\Phi \varepsilon_{BQ} + \Delta\Phi \varepsilon_{Bx}) \\
&\leq -\frac{k_1}{2} \lambda_{\min}(P^{-1} \mathcal{W} P^{-1} + H^\top \mathcal{Y}^{-1} H) ||\tilde{y}||^2 \\
&\quad - k_2 S(x) - k_2 R ||\alpha||^2 - k_3 a ||M|| - (k_4 \ell - \frac{k_4 \eta_2}{2}) ||\Lambda||^2 \\
&\quad + \frac{k_1 \lambda_{\max}^2(P^{-1})}{\eta_1} ||\tilde{y}||^2 + \frac{k_1 \lambda_{\max}^2(H^\top \mathcal{Y}^{-1})}{2\eta_1} ||\tilde{y}||^2 \\
&\quad + \frac{k_1 \eta_1}{2} \delta^2 + \frac{k_1 \eta_1}{2} \varpi^2 + \frac{k_1 \eta_1}{2} \mu^2 + k_2 ||g \nabla V_x^*|| (||\varepsilon_{Q\alpha}|| + ||\varepsilon_{x\alpha}||) \\
&\quad + \frac{k_4}{2\eta_2} ||\Delta\Phi|| (||\varepsilon_{BQ}|| + ||\varepsilon_{Bx}||) \\
&\leq -a_1 ||\tilde{y}||^2 - a_2 S(x) - a_2 ||\alpha||^2 - a_3 ||M|| - a_4 ||\Lambda||^2 + b
\end{aligned}
\tag{6.40}
$$

where $\lambda_{\min}$ and $\lambda_{\max}$ denote the minimum and maximum absolute eigenvalues of a matrix; $a_1 = k_1 \lambda_{\min}(P^{-1} \mathcal{W} P^{-1} + H^\top \mathcal{Y}^{-1} H)/2 - k_1 \lambda_{\max}^2(P^{-1})/\eta_1 - k_1 \lambda_{\max}^2(H^\top \mathcal{Y}^{-1})/2\eta_1$, $a_2 = k_2, a_3 = k_2 R, a_3 = k_3 a, a_4 = k_4 \ell - k_4 \eta_2/2$ are positive constants for the properly chosen constants η_1 and η_2; η_1 and η_2 are constants in Young's inequality, e.g., a constant η in $ab \leq \frac{\eta}{2} a^2 + \frac{1}{2\eta} b^2$, with $\eta_1 > [2\lambda_{\max}^2(P^{-1})/\eta_1 + \lambda_{\max}^2(H^\top \mathcal{Y}^{-1})/2\eta_1]/\lambda_{\min}(P^{-1} \mathcal{W} P^{-1} + H^\top \mathcal{Y}^{-1} H)$ and $\eta_2 < 2\ell$; $b = \frac{k_1 \eta_1}{2} \delta^2 + \frac{k_1 \eta_1}{2} \varpi^2 + \frac{k_1 \eta_1}{2} \mu^2 + k_2 ||g \nabla V_x^*|| (||\varepsilon_{Q\alpha}|| + ||\varepsilon_{x\alpha}||) + \frac{k_4}{2\eta_2} ||\Delta\Phi|| (||\varepsilon_{BQ}|| + ||\varepsilon_{Bx}||)$ is the lumped residue.

Thus, the first five terms in the last inequality of (6.40) form a negative definite function in $\tilde{\Omega}$ so that the set of ultimate boundedness Ω_α exists and it depends on the size of b, i.e. a smaller value of b will decrease the size of Ω_α. Assuming that n has been chosen large enough, we have small $||\varepsilon_{Q\alpha}||, ||\varepsilon_{BQ}||$. Moreover, $||\varepsilon_{x\alpha}||, ||\varepsilon_{Bx}||$ will be small given the exponential convergence of ε_x caused by the EKF (6.26). Hence, it is possible to obtain b to be sufficiently small so that $\Omega_\alpha \subset \tilde{\Omega}$. Hence, it is impossible for any trajectory to leave $\tilde{\Omega}$, i.e. it is an invariant set, i.e. the states $x(t)$ remain bounded and subsequently also the functions of $x(t)$ and $\alpha(t)$: approximation error

$\varepsilon_Q(x,\alpha)$, $\Phi(x,\alpha)$ are bounded functions over a compact set. From Lyapunov's theorem and *Lemma 6.2*, $\mathcal{L}$ is uniformly ultimately bounded. It follows the actor α will converge to a small bounded set near its optimal solution α^*. □

6.5 Simulations

An MVEM of a Wankel rotary engine is created in Matlab/Simulink, where the model parameters are calibrated based on the experimental data sets (see Chapter 5 and [207], [2] for more detail). For the idle speed control problem, we choose the value function as (6.7) with $S^0 = diag[1\ \ 1\ \ 1\ \ 1\ \ 4]$ and $R = 1$. The activation function $\Phi(x,\alpha)$ of the adaptive critic neural network (6.15) is selected as $\Phi(x,\alpha) = [p_m^2\ \ p_m T_m\ \ T_m^2\ \ m_{fp}\ \ m_{fv}\ \ p_m N\ \ N^2\ p_m\alpha\ \ T_m\alpha\ \ m_{fp}\alpha\ \ m_{fv}\alpha\ \ N\alpha\ \ \alpha^2]^\mathsf{T}$ with the number of neurons $n = 13$. We initialise the state $x(0) = [1\ \ 1\ \ 1\ \ 1\ \ 1]^\mathsf{T}$ and the weight $\hat{w}(0) = [0.5\ \ 0.5\ \ 0.5\ \ 0.5\ \ 0.5\ \ 0.5\ \ 0.5\ \ 0.5\ \ 0.5\ \ 0.5\ \ 0.5\ \ 0.5\ \ 1]^\mathsf{T}$. The tuning parameters are chosen as such: the sample period $T = 2$s, the filter parameter $\ell = 1$, the adaptive learning gain $\Gamma = 7$.

We first inject the exploration noise onto the throttle angle for the learning period to ensure the persistent excitation of the signals. The engine load is set to be constant when learning. Fig. 6.2 presents the engine trajectories with the exploration noise for a period of 30 s. The state variables are normalised into the value range of [0,1]. It is shown in Fig. 6.3 that the adaptive critic weights $\hat{W}$ converge before the exploration noise is removed from 700s.

In order to validate the performance of the resulting controller after learning, we simulate a load disturbance (caused for instance by power steering or transmission engagement) at 900s and 1000s. The results are presented in Fig. 6.4, where the engine speed response under the resulting controller is plotted against that when there is no control action. The controller can effectively reject the disturbance in either case of an increase or decrease in the engine load.

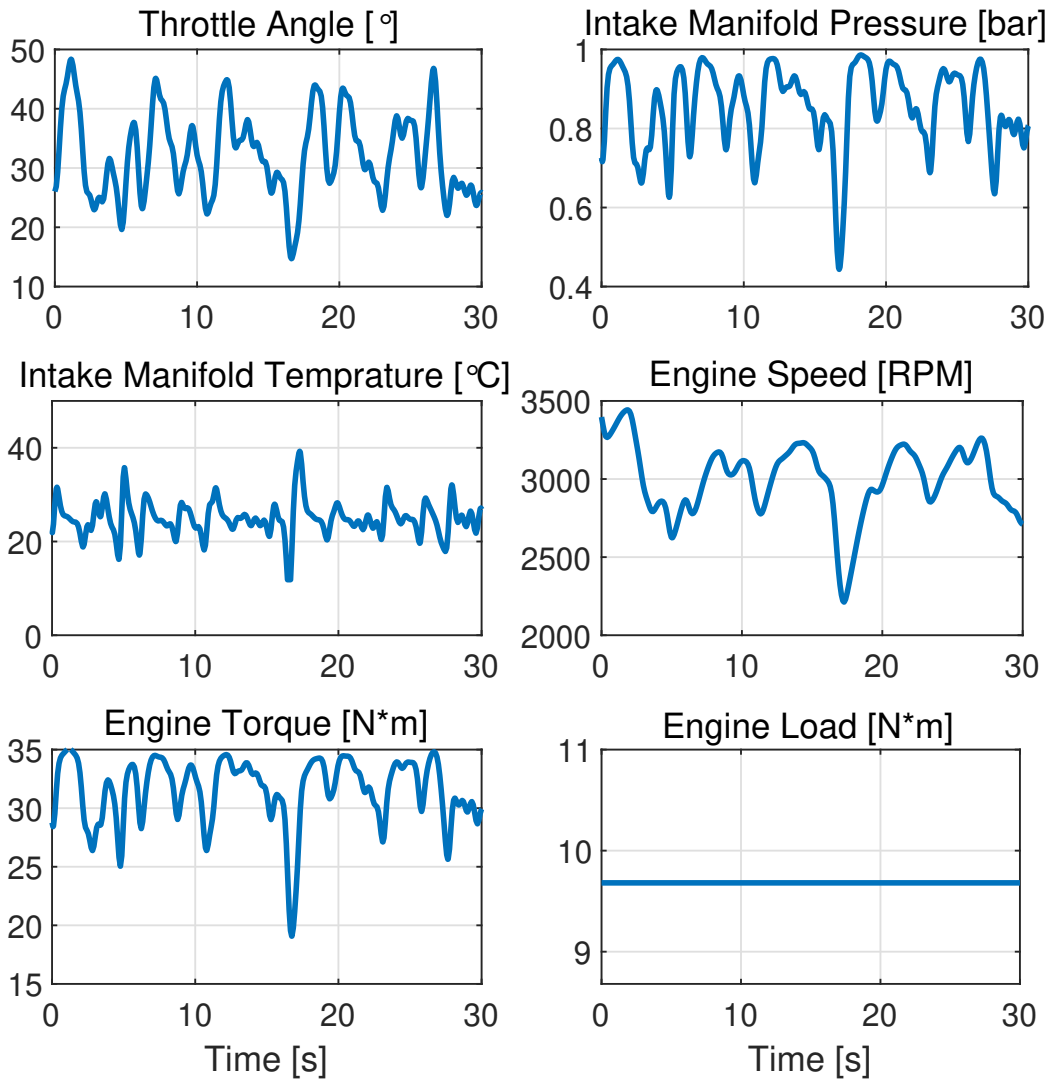

Figure 6.2: Engine trajectories with the exploration noise.

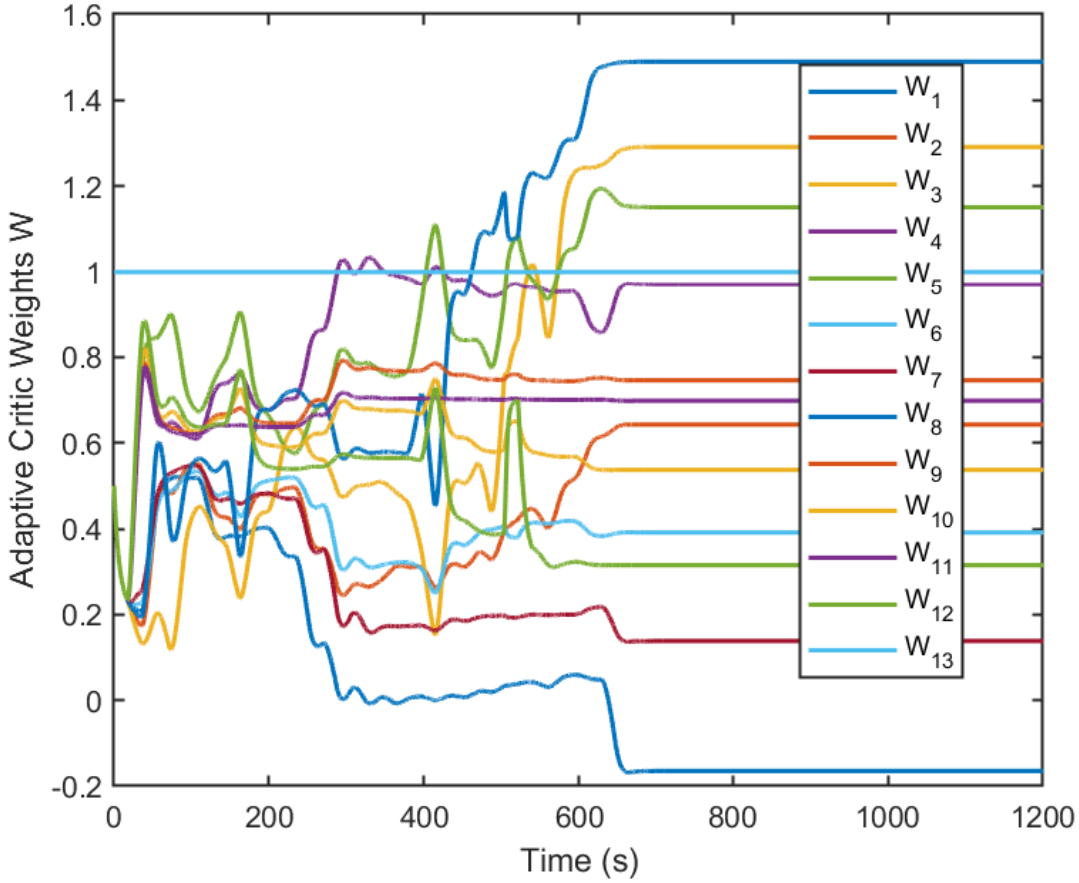

Figure 6.3: Adaptive critic weights convergence.

6.6 Conclusions

In this chapter, we have proposed an adaptive optimal output feedback controller for the idle speed regulation problem using reinforcement learning principles, namely, Q-learning. The Q-learning-based controller is data-driven and completely model-free. Via an EKF, the feedback control requires only the measurable outputs (intake manifold pressure and temperature, engine speed), i.e., not all the state variables are needed. The convergence of the estimation errors and the value function is proved in the sense of Lyapunov stability. The simulation on a Wankel engine model showed the proposed controller, after learning, can effectively reject load disturbance and reg-

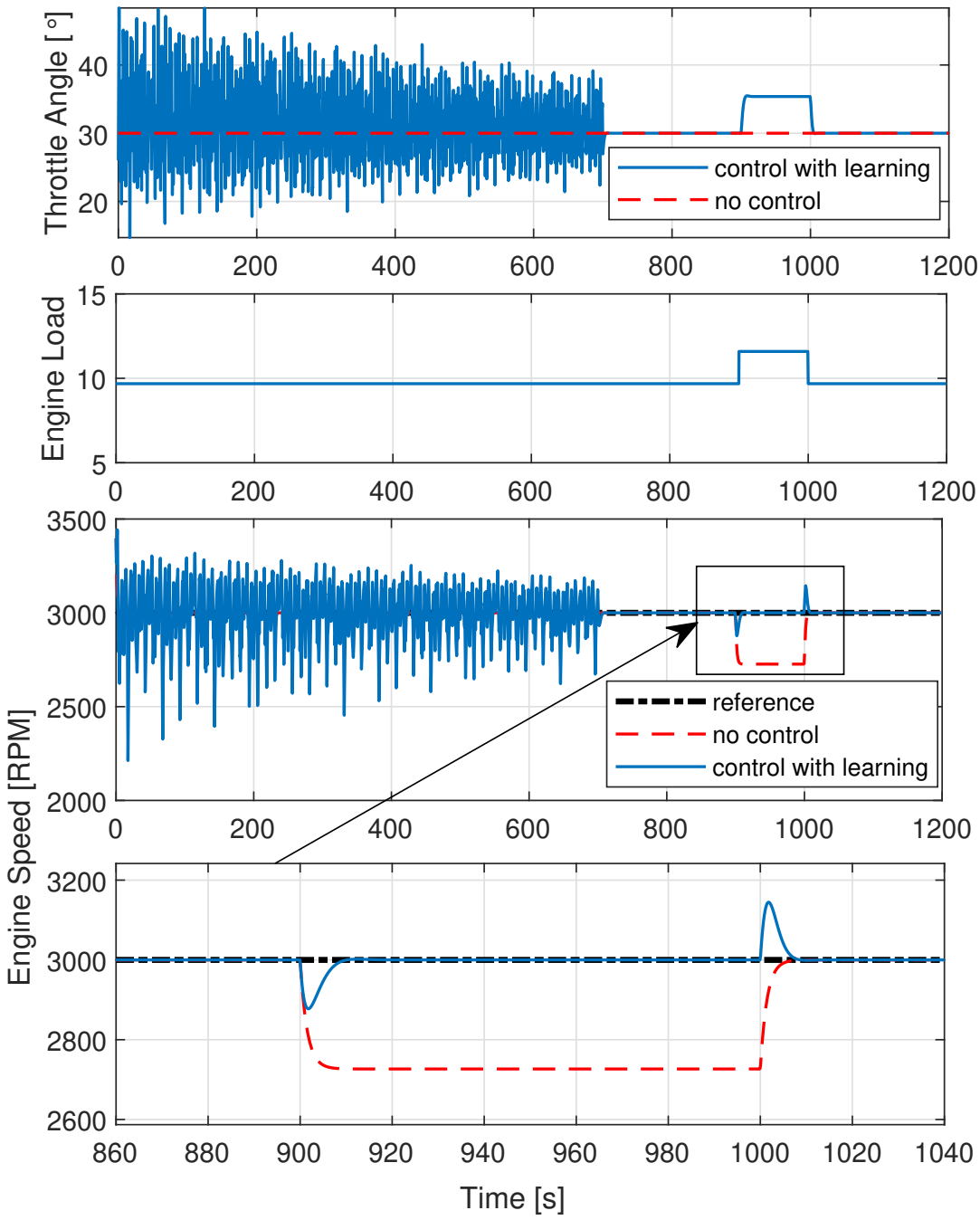

Figure 6.4: Simulation results of the learning-based controller.

ulate the engine idle speed around a desired point. Due to a hardware limitation that the available AIE 225CS Wankel in the test-rig is not equipped with an electronic throttle control unit, a practical engine test is not included in this work. Future work will focus on the practical validation of the proposed controller via engine tests.

CHAPTER 7

AIR-FUEL RATIO CONTROL VIA NONLINEAR OBSERVERS: THEORY AND EXPERIMENTAL VALIDATION*

Emission regulations has been more and more stringent around the world since the 1970s. The shortcoming of high emissions has severely limited the application of Wankel rotary engines in the automotive industry. The fuel puddles due to port fuel injection (PFI) and the leakage between combustion chambers are significant sources of efficiency loss and emissions. For most spark ignition engines in production, the emission strongly depends on the air-fuel ratio (AFR) controller in cooperation with a three-way catalytic (TWC) converter. In this chapter, we focus on the AFR control problem for a Wankel rotary engine. Before jumping to the proposed adaptive optimal control approach, this work will deal with the model uncertainty and nonlinearity using a new observer-based AFR control framework.

*The content of this chapter is adapted from the author's own work [207], where some materials have been re-used.

7.1 Introduction

The main cause of the high emissions of Wankel rotary engines is given by its design. On the one hand, the significantly different temperatures in each combustion chamber often lead to imperfect sealing, which accounts for leakage and unburned fuel mixture [4]. On the other hand, it is inevitable for port fuel injection (PFI) that a considerable portion of fuel will be trapped at the intake manifold wall as fuel puddles, which is also known as the "wall-wetting" phenomenon. One way to overcome this is implementing direct fuel injection (DFI) into the combustion chambers, which has proved successful for reciprocating compression ignition engines and then in reciprocating spark ignition engines [225]. However, the implementation of DFI is likely to increase the cost and the complexity of engine configurations. Alternatively, one can design an AFR controller using observers to compensate for the effect of fuel puddle dynamics and for the rapid change of air-filling dynamics.

The common treatment for engine emissions is to convert pollutant exhaust CO, NO_x, into innocuous ones: N_2, H_2O, and CO_2, using three-way catalytic (TWC) converters. However, as shown in Fig. 2.6, the conversion efficiency of TWC is fairly sensitive to AFR, which is required to be regulated around the stoichiometric value (e.g. 14.7 for petrol) [136]. Moreover, combustion with a stoichiometric AFR is essential to achieve the optimal thermal efficiency and dynamic performance. Therefore, it is of great importance to design a well-performing AFR controller for Wankel engines so as to improve emissions, thermal efficiency and fuel economy. For most spark ignition engines in production, the widely-used control strategy is still PID control based on lookup tables, with which it could be difficult to meet the emission requirements in the presence of complex dynamics and rapidly-changing operational scenarios of Wankel engines. Practically, the compilation of the lookup tables also requires significant effort in engine calibration tests and is usually time-consuming [137].

This motivates the research on advanced AFR control design such as optimal con-

trol [138], robust control [139][140], adaptive control [136][141], and more recently, observer-based control [137][142]. An optimal AFR controller was designed in [138] considering the cyclic variations of residual gas. However, it needs the knowledge of in-cylinder pressure which requires expensive sensors that are not suited to commercial applications. Then, robust techniques such as H_∞ control [139] and sliding mode control [140] were proposed to regulate the AFR in the presence of external disturbance. In order to deal with parameter uncertainties, adaptive approaches were presented to address air-filling dynamics in [136] and time delay dynamics in [141]. However, the complexity of the adaptive controller limits their practical implementation. This prompts further work on AFR control using simple, easily implemented observers. In [142], a sliding mode AFR controller was proposed using observers to reduce chattering. Later on, various popular observer techniques were investigated in [137], which show great potential in application with design simplicity. However, the effect of fuel puddle dynamics was not specifically studied in [137]. There are two parameters, fuel puddle fraction and the time constant for the puddle evaporation, which are assumed to be known for AFR control, but which are not measurable in practice. We will use novel nonlinear observer techniques to estimate the unknown dynamics and account for disturbances.

In this chapter, in order to design a simple yet robust AFR controller for Wankel engines, a mean-value engine model is developed. Moreover, as in Chapter 6, this work incorporates the idea of an extended Kalman filter (EKF) [219] to account for the effect of fuel puddle dynamics. By reformulating the AFR regulation into a fuel flow tracking problem, various popular observer techniques [137] are investigated and employed in the AFR control design, which leads to a new observer-based AFR control framework. Comparative simulations present the improved transient and steady-state responses.

7.2 Problem Formulation

It has been shown in our previous work [136] that the AFR regulation problem can be reformulated into a fuel flow tracking problem for the design purpose. Then the feedback control error e used in the AFR controller can be defined as

$$e = \dot{m}_{fd} - \dot{m}_f = \frac{1}{\lambda_d}\dot{m}_a - \dot{m}_f \tag{7.1}$$

where $\dot{m}_{fd}$ and λ_d are the desired fuel mass flow rate and AFR. Thus, its derivative is calculated as

$$\dot{e} = \frac{1}{\lambda_d}\ddot{m}_a - \ddot{m}_f \tag{7.2}$$

Clearly, $\ddot{m}_a$ is the derivative of complex nonlinear air-filling dynamics (5.5) and $\ddot{m}_f$ is the derivative of (5.6) with fuel puddle dynamics (5.7) as in Chapter 5. The measurement of $\ddot{m}_a$ and $\ddot{m}_f$ is practically infeasible. However, it is conceivable to estimate them using nonlinear observers.

In order to understand the dynamics of the control error, substituting the air-filling dynamics (5.5) and the fuel puddle dynamics (5.6)(5.7) from Chapter 5 into (7.2) gives

$$\dot{e} = M - u_d \tag{7.3}$$

where $M = \frac{V_d}{120R\lambda_d}\frac{d(\eta_{vol}p_m n/T_m)}{dt} - \frac{m_{fp}}{\tau\tau_m} + \frac{m_{fv}}{\tau_m^2}$ is lumped unknown dynamics to be observed and $u_d = (1-\chi)u/\tau_m$ is a linear function of the control input u with the unknown parameter χ.

7.3 Nonlinear Observers Design

This section investigates popular observer techniques from [137] to estimate the dynamics of $\ddot{m}_a$ and $\ddot{m}_f$, which will be used in the AFR control design. The extended Kalman filter is used for the fuel puddle dynamics whereas the differentiation observer and the unknown input observer are used for air-filling dynamics.

7.3.1 Differentiation Observer

An intuitive way to observe the unknown lumped dynamics M is defined in terms of the "the dirty derivative" of e using the low-pass filter operation $(\bullet)_f = [\bullet]/(ks+1)$, which can be given as

$$\frac{s}{ks+1}e = [\frac{1}{k} - \frac{1}{k(ks+1)}]e \tag{7.4}$$

where $k > 0$ is the design filter parameter. Then the differentiation observer is designed as

$$\hat{M} = \dot{e}_f + \hat{u}_d \tag{7.5}$$

with $\hat{u}_d = (1-\hat{\chi})u/\tau_m$, where $\hat{\chi}$ is the estimate of χ using the extended Kalman filter (6.26). From (7.3)(7.4)(7.5), the estimation error can be written as

$$\tilde{M} = M - \hat{M} = \frac{ks^2}{ks+1}e + \tilde{u}_d \tag{7.6}$$

with $\tilde{u}_d = u_d - \hat{u}_d$.

Assumption 7.1 *The second derivative of e and the first derivative of $\tilde{x}$ are assumed to be bounded, i.e. $sup_{t\geq 0}|\ddot{e}(t)| \leq \psi$ and $sup_{t\geq 0}||\dot{\tilde{x}}(t)|| \leq \zeta$ with $\psi > 0$ and $\zeta > 0$.*

Proposition 7.1 *Under Assumption 7.2, for the error dynamics* (7.3) *with the differentiation observer* (7.5) *and the extended Kalman filter* (6.26)*, the estimation errors $\tilde{M}$, $\tilde{x}$ will exponentially converge to a compact set around zero.* ◇

Proof: Presenting (7.6) in the time domain gives

$$\dot{\tilde{M}} = -\tilde{M}/k + \ddot{e} + \dot{\tilde{u}}_d + \tilde{u}_d/k \tag{7.7}$$

By choosing a Lyapunov function $V_1(t) = \tilde{M}^2/2 + k_1 V_k$ with $k_1 > 0$, it follows that

$$\dot{V}_1(t) = \tilde{M}\dot{\tilde{M}} + k_1\dot{V}_k \leq -a_1(\sigma_1)V_1 + \beta_1(\sigma_1) \tag{7.8}$$

where $a_1(\sigma_1) = \min\{2/k - 3/\sigma_1,\ (a(\sigma_1)k_1 - \sigma_1/k^2)/\lambda_{\max}(Y)\}$ and $\beta_1(\sigma_1) = \sigma_1(\psi^2+\zeta^2)/2 + k_1\beta(\sigma_1)$ are positive constants if the constants σ_1 and k_1 are properly selected as $\sigma_1 >$

$3k/2$ and $k_1 > \sigma_1/a(\sigma_1)k^2$. This implies $V_1(t) \leq V_1(0)e^{-a_1(\sigma_1)t} + \beta_1(\sigma_1)/a_1(\sigma_1)$ holds and thus $\tilde{M}$, $\tilde{x}$ will exponentially converge towards $\Omega_2 := \{\tilde{M} \mid |\tilde{M}| \leq \sqrt{2\beta_1(\sigma_1)/a_1(\sigma_1)}\}$ and $\Omega_3 := \{\tilde{x} \mid ||\tilde{x}|| \leq \sqrt{2\beta_1(\sigma_1)/a_1(\sigma_1)\lambda_{\min}(Y)}\}$, respectively. □

7.3.2 Unknown Input Observer

The unknown input observer [137] is designed based on the idea of using the low-pass filter operation $(\bullet)_f = [\bullet]/(ks+1)$ on both sides of (7.3) such that

$$\dot{e}_f = M_f - u_{df} \tag{7.9}$$

Then the unknown input observer can be designed as

$$\hat{M} = M_f = \frac{e - e_f}{k} + \hat{u}_{df} \tag{7.10}$$

where $e_f, \hat{u}_{df}$ are the filtered version of $e, \hat{u}_d$ as

$$\begin{cases} k\dot{e}_f + e_f = e, & e_f(0) = 0 \\ k\dot{\hat{u}}_{df} + \hat{u}_{df} = \hat{u}_d, & \hat{u}_{df}(0) = 0 \end{cases} \tag{7.11}$$

with the design filter parameter $k > 0$. From (8.3)(7.4)(7.9)(7.10), the estimation error can be described as

$$\tilde{M} = M - \hat{M} = \frac{ks}{ks+1}M + \frac{1}{ks+1}\tilde{u}_d \tag{7.12}$$

Assumption 7.2 *It is practically feasible to assume that the derivative of the lumped unknown term M is bounded, i.e. $\sup_{t\geq 0}|\dot{M}(t)| \leq \xi$ with $\xi > 0$.*

Proposition 7.2 *Under Assumption 7.2, for the error dynamics* (7.3) *with the unknown input observer* (7.10) *and the extended Kalman filter* (6.26)*, the estimation errors $\tilde{M}$, $\tilde{x}$ will exponentially converge to a compact set around zero.* ◇

Proof: Presenting (7.12) in the time domain gives

$$\dot{\tilde{M}} = -\tilde{M}/k + \dot{M} + \tilde{u}_d/k \tag{7.13}$$

By choosing a Lyapunov function $V_2(t) = \tilde{M}^2/2 + k_2 V_k$ with $k_2 > 0$, it follows that

$$\dot{V}_2(t) = \tilde{M}\dot{\tilde{M}} + k_2\dot{V}_k \leq -a_2(\sigma_2)V_2 + \beta_2(\sigma_2) \tag{7.14}$$

where $a_2(\sigma_2) = \min\{2/k - 2/\sigma_2,\ (a(\sigma_2)k_2 - \sigma_2/k^2)/\lambda_{\max}(Y)\}$ and $\beta_2(\sigma_2) = \sigma_2\xi^2/2 + k_2\beta(\sigma_2)$ are positive constants if σ_2, k_2 are properly selected as $\sigma_2 > k$, $k_2 > \sigma_2/a(\sigma_2)k^2$. This implies $V_2(t) \leq V_2(0)e^{-a_2(\sigma_2)t} + \beta_2(\sigma_2)/a_2(\sigma_2)$ holds and $\tilde{M}$, $\tilde{x}$ will exponentially converge towards $\Omega_4 := \{\tilde{M} \mid |\tilde{M}| \leq \sqrt{2\beta_2(\sigma_2)/a_2(\sigma_2)}\}$ and $\Omega_5 := \{\tilde{x} \mid ||\tilde{x}|| \leq \sqrt{2\beta_2(\sigma_2)/a_2(\sigma_2)\lambda_{\min}(Y)}\}$, respectively. ◇

Remark 7.1 *The filter operation* $(\bullet)_f$ *is applied to* e *for both observers* (7.5) *and* (7.10). *However,* $\hat{u}_d$ *is also filtered in* (7.10) *while in* (7.5) $\hat{u}_d$ *is directly coupled with* $\hat{M}$. *Furthermore, by inspection of* (7.6) *and* (7.12), *the estimation error* $\tilde{M}$ *can be minimised by setting the filter parameter* k *sufficiently small.* ◇

Remark 7.2 *Assumption 7.2 for the unknown input observer* (7.10) *is weaker than Assumption 7.1 for the differentiation observer* (7.5). *For both observers, the upper bound of the estimation error* $\tilde{M}$ *depends on the convergence bound* $\beta(\bullet)$ *of the extended Kalman filter* (6.26). *For* (7.5), *it also depends on the upper bound* ψ *of the second derivative of* e *and the upper bound* ζ *of the first derivative of* $\tilde{x}$. *In contrast, for* (7.10), *it depends only on the upper bound* ξ *of the first derivative of* M *apart from* $\beta(\bullet)$, *which is a weaker condition in practice.* ◇

7.4 AFR Control Design

Based on the observers above, the injected fuel mass flow rate for AFR control can be designed as

$$u = \dot{m}_{fi} = \tau_m(k_p e + \hat{M})/(1 - \hat{\chi}) \tag{7.15}$$

where $k_p > 0$ is the feedback gain to be tuned, $\hat{\chi}$ and $\hat{M}$ are the estimates of χ and M via the extended Kalman filter (6.26) and the unknown input observer (7.10), respectively. Fig. 7.1 presents a schematic diagram of the proposed observer-based AFR control system.

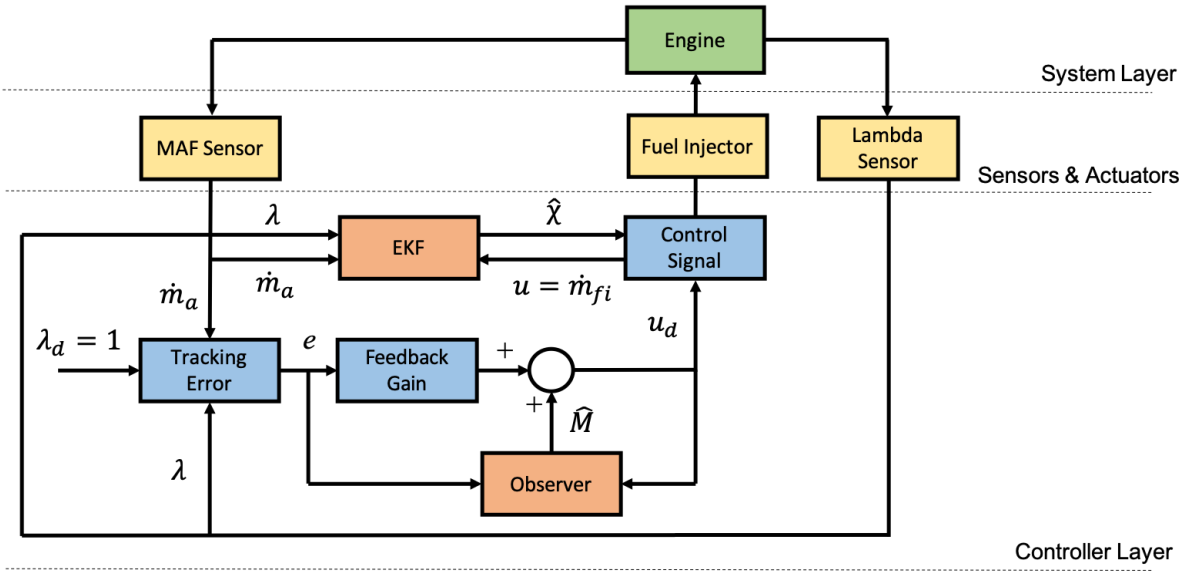

FIGURE 7.1. A schematic diagram of the proposed observer-based AFR control system. The observer represents the differentiation observer or the unknown input observer.

Theorem 7.1: *For the Wankel engine model given in Chapter 5 and the error dynamics* (7.3)*, the AFR control* (7.15) *based on the extended Kalman filter* (6.26) *and the unknown input observer* (7.10) *will lead to the exponential convergence of the error* e *and the estimation errors* $\tilde{M}$, $\tilde{x}$ *towards a small compact set around zero.*

Proof: Substituting (7.15) into $u_d = (1-\chi)u/\tau_m$ gives

$$\begin{aligned} u_d &= (1-\chi)(k_p e + \hat{M})/(1-\hat{\chi}) \\ &= \tilde{\chi}(k_p e + \hat{M})/(1-\hat{\chi}) + (k_p e + \hat{M}) \end{aligned} \tag{7.16}$$

with $\tilde{\chi} = \chi - \hat{\chi}$. Then the closed-loop error dynamics can be calculated by substituting (7.16) into (7.3) as

$$\dot{e} = -k_p e + \tilde{M} - \tilde{\chi}(k_p e + \hat{M})/(1-\hat{\chi}) \tag{7.17}$$

Selecting a Lyapunov function as

$$W(t) = e^2/2 + k_3\tilde{M}^2/2 + k_4\tilde{x}^T Y \tilde{x}/2 \tag{7.18}$$

with $k_3 > 0$, $k_4 > 0$ to be chosen, its first derivative can be determined using (**??**)(7.14)(7.17) as

$$\begin{aligned}\dot{W}(t) &= e\dot{e} + k_3\dot{V}_2 + k_4\dot{V}_k \\ &\leq -[k_p - \epsilon(k_p^2 e^2 + \hat{M}^2)/2(1-\hat{\chi})^2 - \epsilon/2]e^2 \\ &\quad - (k_3/k - k_3/\epsilon - 1/2\epsilon)\tilde{M}^2 \\ &\quad - [(k_2k_3 + k_4)a(\epsilon) - \epsilon k_3/2k^2 - 2/\epsilon]||\tilde{x}||^2/2 \\ &\quad + k_3\beta_2(\epsilon) + k_4\beta(\epsilon) \\ &\leq -bW(t) + \Delta\end{aligned} \tag{7.19}$$

where $b = \min\{2k_p - \epsilon(k_p^2 e^2 + \hat{M}^2)/(1-\hat{\chi})^2 - \epsilon,\ 2(k_1/k - k_3/\epsilon - 1/2\epsilon),\ (k_2k_3a(\epsilon) + k_4a(\epsilon) - \epsilon k_3/2k^2 - 2/\epsilon)/\lambda_{\max}(Y)\}$ and $\Delta = k_3\beta_2(\epsilon) + k_2\beta(\epsilon)$ are positive constants if k_p, k_3, k_4 are properly selected as $k_p > \epsilon(k_p^2 e^2 + \hat{M}^2)/2(1-\hat{\chi})^2 - \epsilon/2$, $k_3 > k/2(\epsilon - k)$, $k_4 > \epsilon k_3/2a(\epsilon)k^2 + 2/a(\epsilon)\epsilon - k_2k_3a(\epsilon)$ with a constant $\epsilon > 0$. This implies that $W(t) \leq W(0)e^{-bt} + \Delta/b$ holds and then e, $\tilde{M}$ will exponentially converge towards $\Omega_6 := \{\Psi \mid |\Psi| \leq \sqrt{2\Delta/b}\}$ and $\tilde{x}$ will exponentially converge towards $\Omega_7 := \{\tilde{x} \mid ||\tilde{x}|| \leq \sqrt{2\Delta/b\lambda_{\min}(Y)}\}$. □

Remark 7.3 *Similar to Theorem 7.1, the control* (7.15) *with the differentiation observer* (7.5) *can be addressed in the same manner via Proposition 7.1.* ◇

In practice, the control error e in (7.15) can be obtained from (7.1), where $\dot{m}_a$ is measured by a MAF sensor and $\dot{m}_f$ is calculated using MAF and lambda sensors as $\dot{m}_f = \dot{m}_a/\lambda$. By Theorem 7.1, the fuel mass flow rate $\dot{m}_f$ can track the desired reference $\dot{m}_{fd}$ and thus the AFR λ is regulated around the stoichiometric value λ_d.

7.5 Simulations

An MVEM of a Wankel rotary engine is created in Matlab/Simulink, where the model parameters are calibrated based on the experimental data sets (see Chapter 5 and [207], [2] for more detail). The throttle angle is controlled to operate the engine with proper acceleration and deceleration.

7.5.1 Fuel Puddle Estimation

The fuel puddle dynamics (**??**) are estimated using the extended Kalman filter (6.26). Sufficient Gaussian noises are added into the measurement of AFR λ and air mass flow rate $\dot{m}_a$ to account for the effect of sensor noise. Fig. 7.2 presents the simulation results of the estimated and measured AFR. The extended Kalman filter performs satisfactory estimation for both transients and steady states. Moreover, the unknown parameter τ_f and χ can quickly converge to the true value $\tau_f = 2.5$s and $\chi = 0.6$, which is shown in Fig. 7.3.

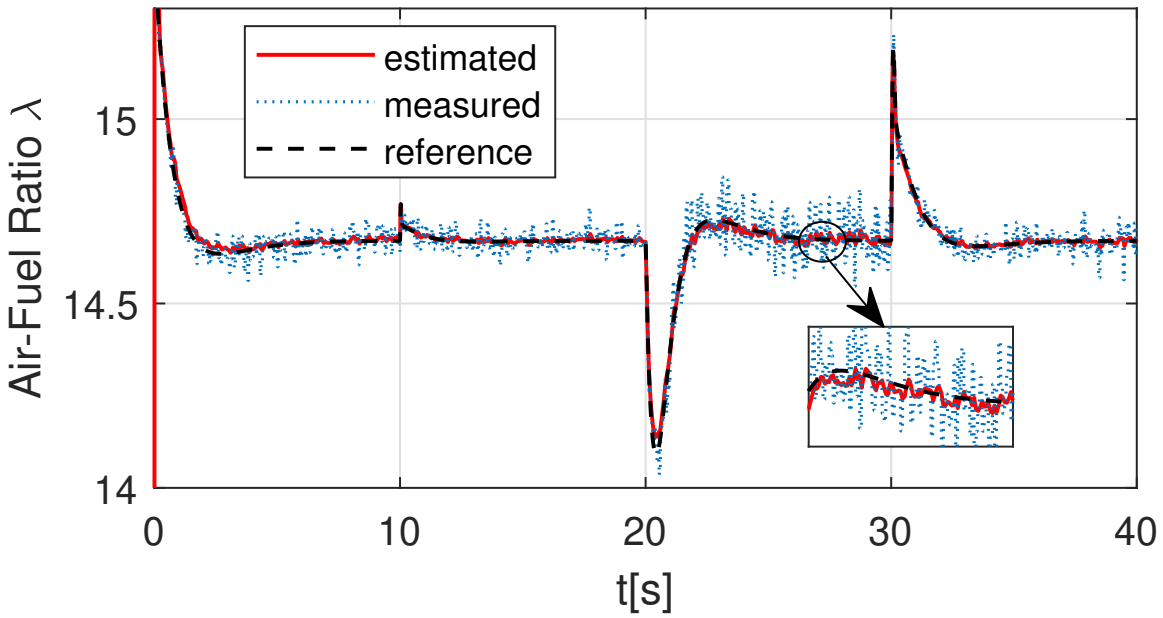

Figure 7.2: Comparison between the estimated and measured AFR.

7.5.2 Air-fuel Ratio Control

The proposed AFR control based on the two observers (7.5)(7.10) are compared with a fixed-gain PID controller, of which the simulation results are provided in Fig. 7.4. It is obvious that all the controllers are able to regulate the AFR at the stoichiometric value $\lambda_d = 14.67$ in steady states. For transients (e.g. $t = 10,\ 20,\ 30$s when engine

accelerates/decelerates), both observer-based controllers (b)(c) to some extent reduce the transient errors compared to the PID controller (a) in Fig. 7.4. However, the controller based on the differentiation observer (7.5) appears to be slower at transients. It is clear that the controller based on the unknown input observer (7.10) achieves better robustness against disturbances.

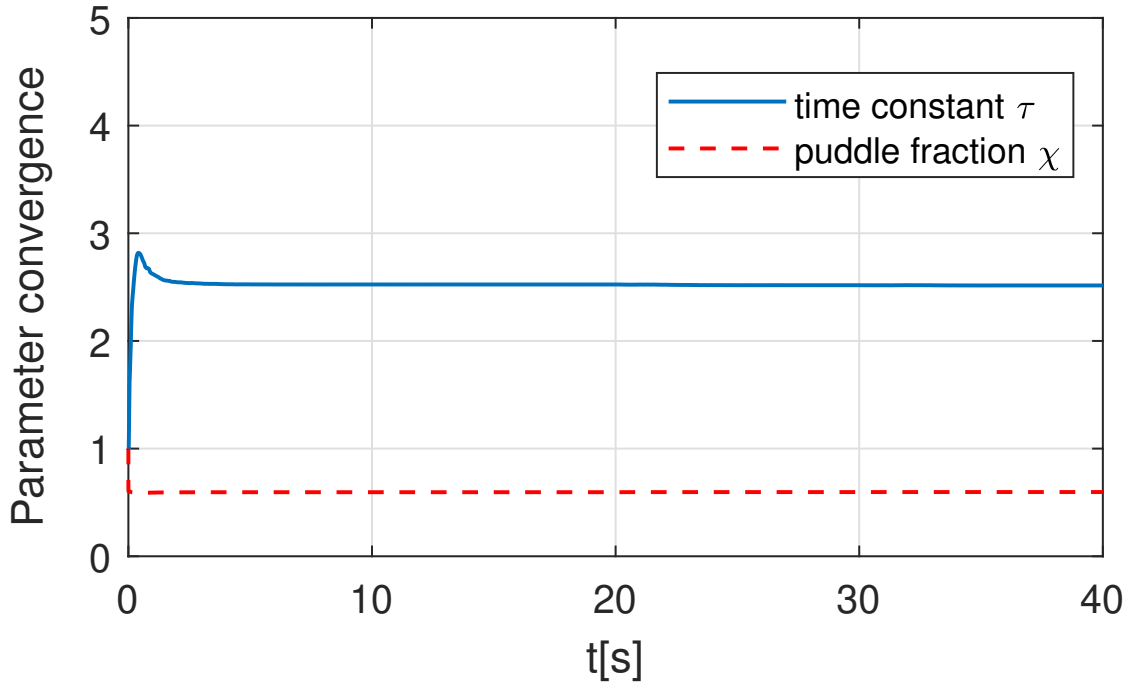

Figure 7.3: Convergence of the estimated fuel puddle parameters.

7.6 Engine Test Set-up

The Wankel engine under investigation for dynamic modelling is a 225CS rotary engine, produced by Advanced Innovative Engineering (AIE) UK Ltd. It is a single-rotor, peripheral-port-injected, twin-spark engine and was previously configured to have a nominal peak power output of 30kW for aerospace use on drones. As mentioned already, due to its high specific power output, there is a recent interest in us-

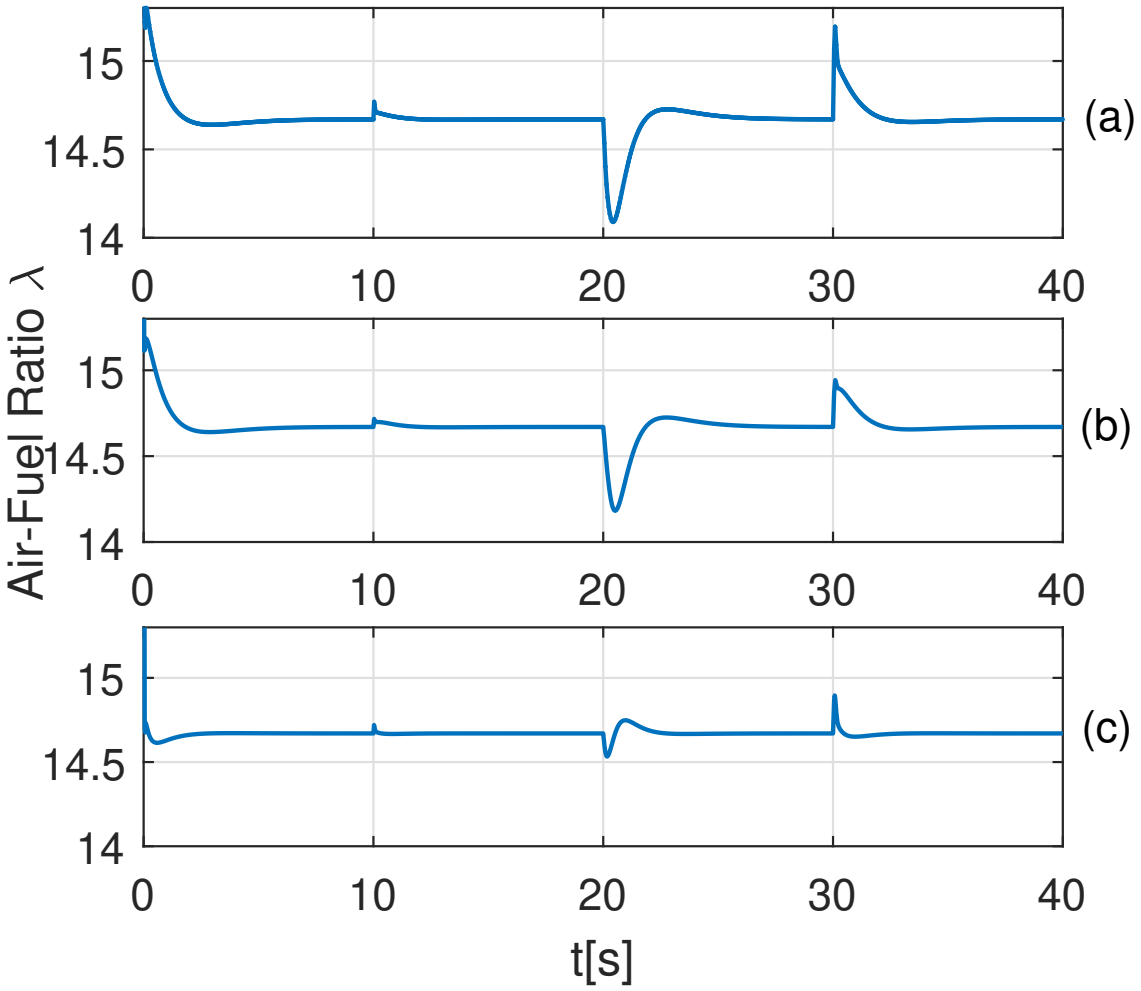

Figure 7.4: Comparison of the AFR responses based on (a) PID control, (b) differentiation observer (7.5), and (c) unknown input observer (7.10).

ing it as a range extender for HEV. Fig. 7.5 shows a picture of the AIE 225CS Wankel rotary engine test set-up.

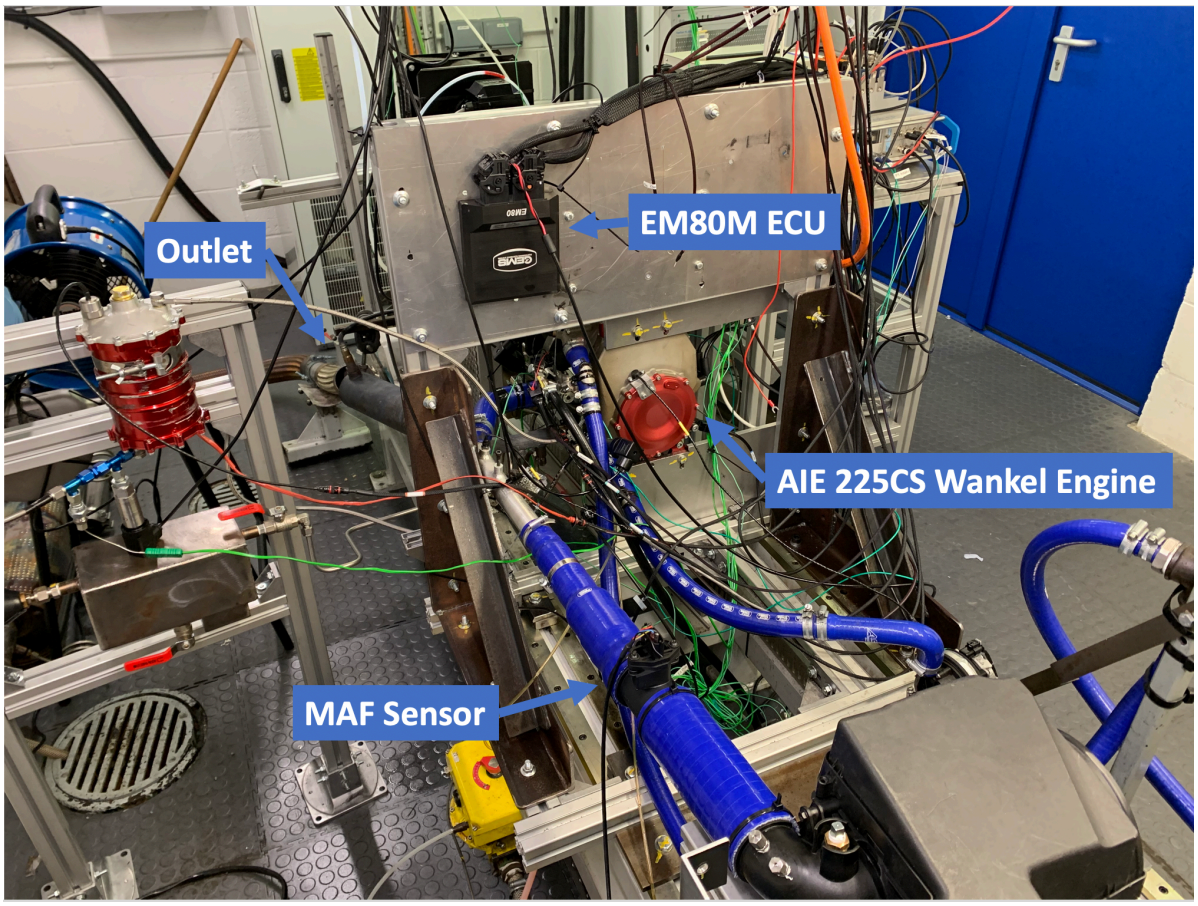

Figure 7.5: A picture of the AIE 225CS Wankel rotary engine test set-up.

The engine experiments are carried out in an engine test cell at the Institute for Advanced Automotive Propulsion Systems (IAAPS) at the University of Bath. The test cell is equipped with an AC dynamometer for the assessment of the engine performance. The maximum nominal power and speed allowed from the AC motor is around 50 kW and 8500 RPM, respectively. The test facilities include a Sierra CP Test Automation System with the proprietary CADET software, which enables the dynamometer control and data acquisition via the transducers installed on the engine. The system can also collect data via the automotive standard Controller Area Network (CAN) bus, which can be compared online with the data from the engine

control unit (ECU). The ECU used by AIE is an EM80 model produced by the project partner General Engine Management Systems (GEMS UK) Ltd (GEMS). In order to implement our new controller, we replace the EM80 model with a new model EM80M. The EM80M allows custom code generated from a Matlab/Simulink model to be incorporated into the co-processor code. The co-processor then runs the model code and interacts with the main processor. The ECU is online configurable when connected to the GEMS GWv4 proprietary software so that the user is able to control and monitor the engine parameters such as air-fuel ratio control, and fly-by-wire throttle control.

It is worth mentioning that the MAF sensor used in the engine tests is a Bosch HFM5. As shown in Fig. 7.5, the MAF sensor is connected using a long straight cylinder pipe after an air filter so that an accurate measurement can be achieved from steady air flow.

The engine test-rig described above was used as the experimental platform to validate the efficacy of the proposed AFR control, which is the extended Kalman filter combined with the unknown input observer. The proposed AFR controller is coded using Simulink, which is then compiled in C code and flashed onto the ECU. In the test, a gain scheduling like PID control is pre-defined using look-up tables during calibration of the engine, which regulates the AFR control system. The PID controller was well tuned by the manufacturer and GEMS. Hence, we can take this PID control as the baseline control and carry out tests to show the effect of the proposed observers-based control.

7.7 Experimental Results and Discussion

In our experiments, the filter constant of the estimator is set as $k = 0.1$ to trade-off the robustness against noise and the control responses under different engine operating regimes. The engine speed is controlled through an available speed control. Fig. 7.6 shows the Brake-specific fuel consumption (BSFC) map of the 225CS Wankel engine.

The auxiliary power unit (APU) is run on the standard New European Drive Cycle (NEDC) with the APU speed and power demand based on the requirement to maintain battery state of charge of a battery electric vehicle, which was provided by Tata Motors European Technical Centre. The NEDC cycle along with the APU power and speed demand over the full period of 1,200 [s] is shown in Fig. 7.7. It is seen that the engine speed varies from 0 to 6,000 [RPM], where in some particular periods, the engine is switched off (based on the APU power demand and battery state of charge). In the test, the ideal AFR demand from the ECU is always $\lambda_d = 1$.

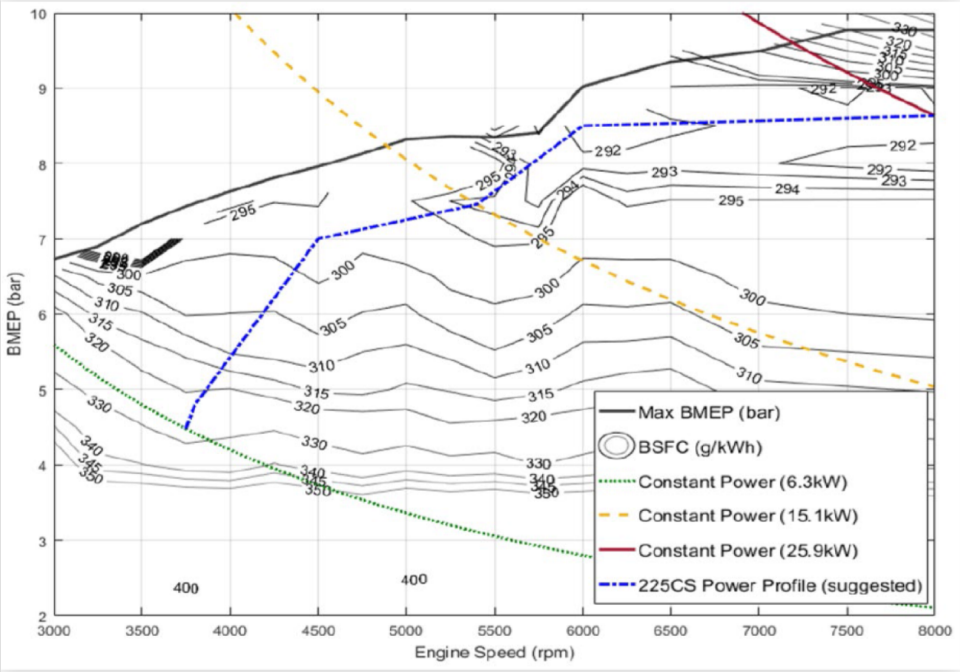

FIGURE 7.6. BSFC map of the 225CS Wankel engine.

Figs. 7.8 - 7.10 present the comparative responses of the engine variables: air mass flow rate, fuel mass flow rate, and exhaust temperature with the proposed control and the gain scheduling like baseline PID control for the full period 1,200 s. This implies that the engine is operated safely and smoothly with both controls. However, there are minor differences at the time instances when the engine changes its speed, which indicates that PID control has more oscillations than the proposed control. This validates the effectiveness of the observer. Fig. 7.11 shows the AFR lambda responses

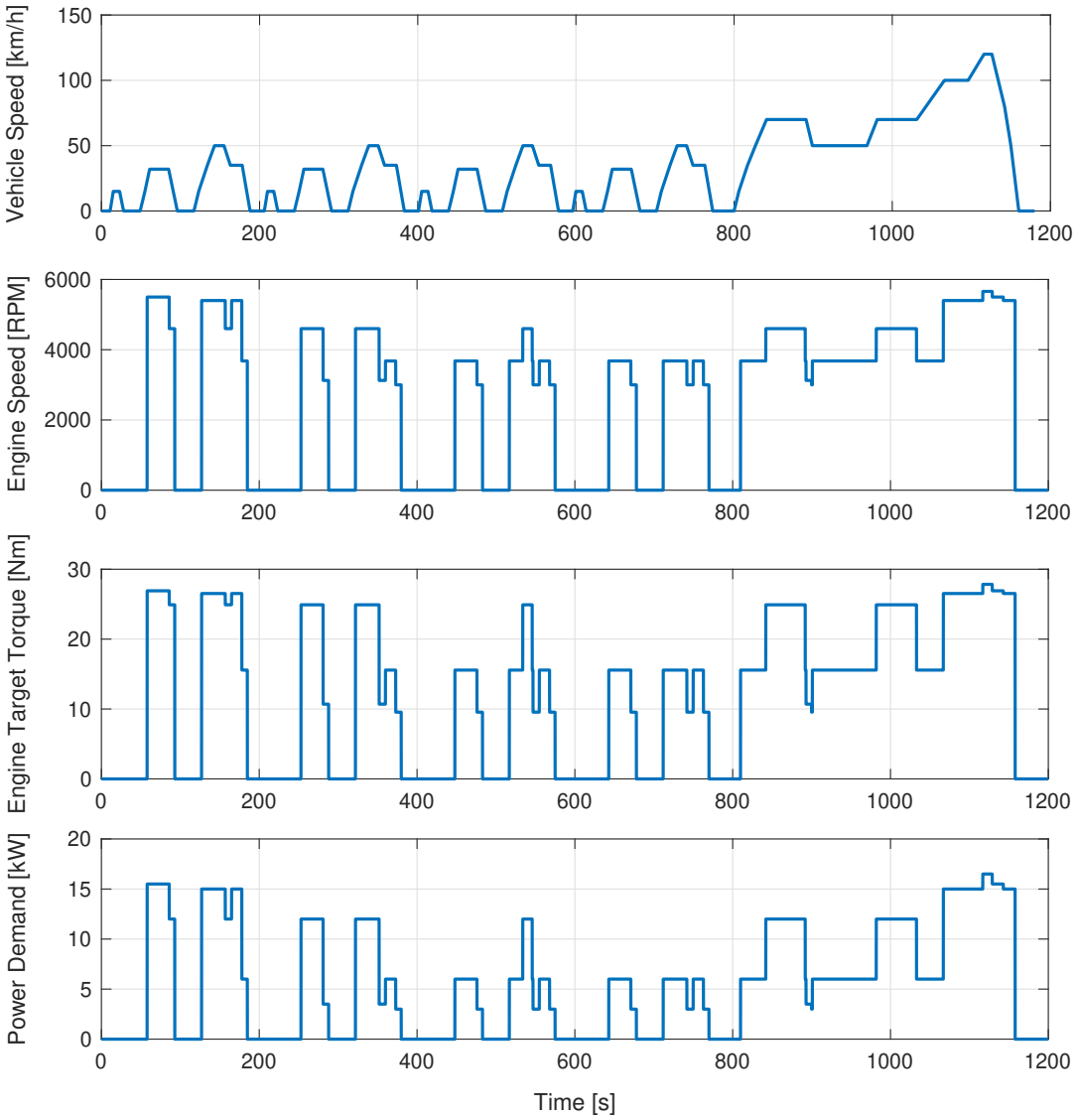

FIGURE 7.7. NEDC drive cycle profile with the Wankel engine speed, torque, and power demand converted from the APU requirement provided by Tata Motors European Technical Centre.

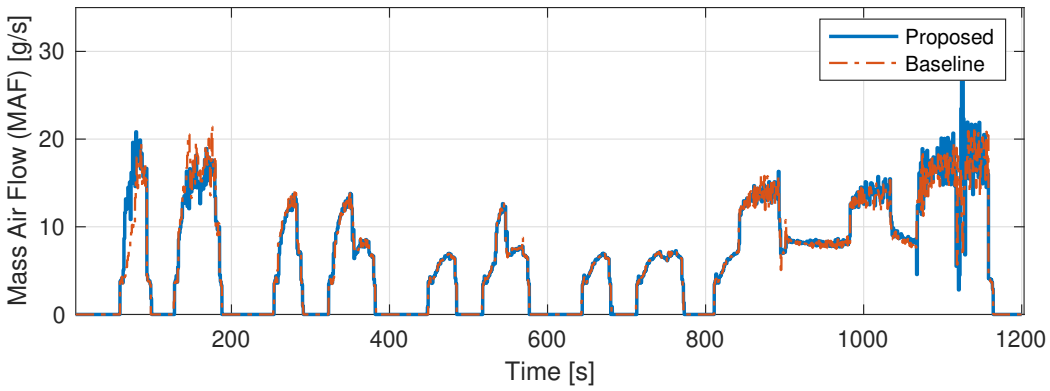

FIGURE 7.8. MAF trajectories of the baseline and proposed control under NEDC drive cycle.

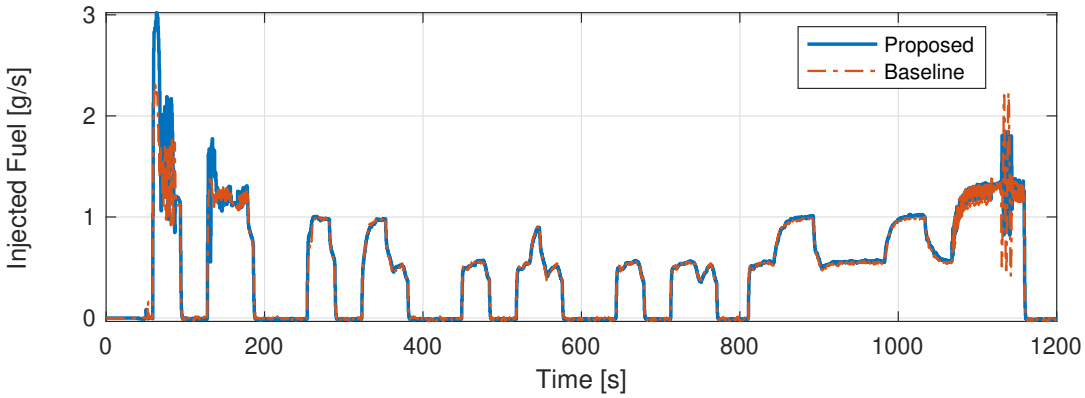

FIGURE 7.9. Fuel flow rate trajectories of the baseline and proposed control under the NEDC drive cycle.

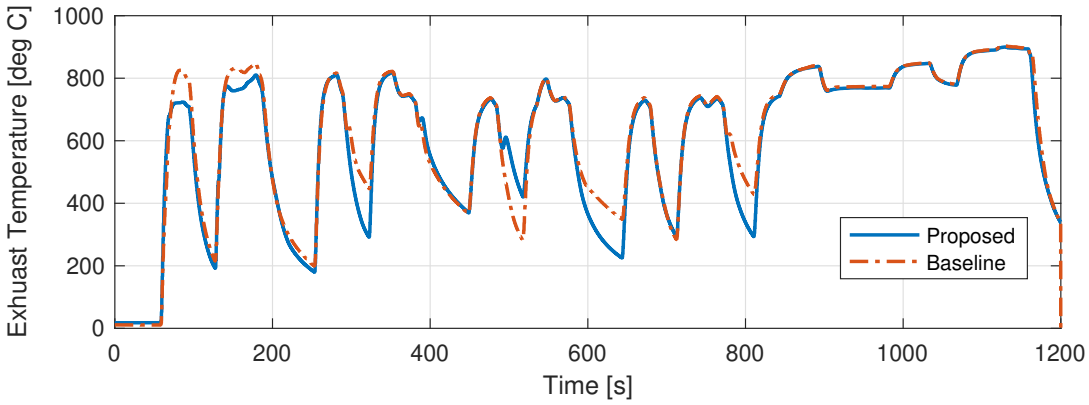

FIGURE 7.10. Exhaust temperature trajectories of the baseline and proposed control under the NEDC drive cycle.

of the two controllers. It should be noted that the lambda reading in the stop periods (when the lambda value is far from $\lambda_d = 1$) is meaningless as the controller is not activated when engine stops running. The lambda trajectories in the valid periods between the two controllers are difficult to distinguish in Fig. 7.11 as both are very close to $\lambda_d = 1$, where the proposed controller seems to have a slightly better AFR response. However, the difference of emission results is significant. Fig. 7.12 demonstrates the emission responses of the baseline and proposed control under the NEDC drive cycle. The real-time and cumulative emission responses of NOx, CO, and total hydrocarbon emissions (THCs) measured by the Horiba Motor Exhaust Gas Analyser (MEXA). One can find that significantly-reduced emissions can be achieved using the proposed control. For example, the NOx is reduced considerably using the proposed control before 200 s as shown in Fig. 7.12. This can be explained by Fig. 7.9 as higher fuel flow rate is achieved in that period so less lean combustion results in less NOx emission. In fact, emissions of CO, NOx, and THC vary with different engines, and the AFR of the mixture in the combustion chamber has the greatest influence on the untreated emissions. An engine that is operated at or very close to the ideal AFR enables both NOx reduction and CO, THC oxidation in a single catalyst bed. Hence, for a catalyst to be efficient, a very tight control of AFR is necessary, where high conver-

sion efficiencies for all three pollutants can be achieved.

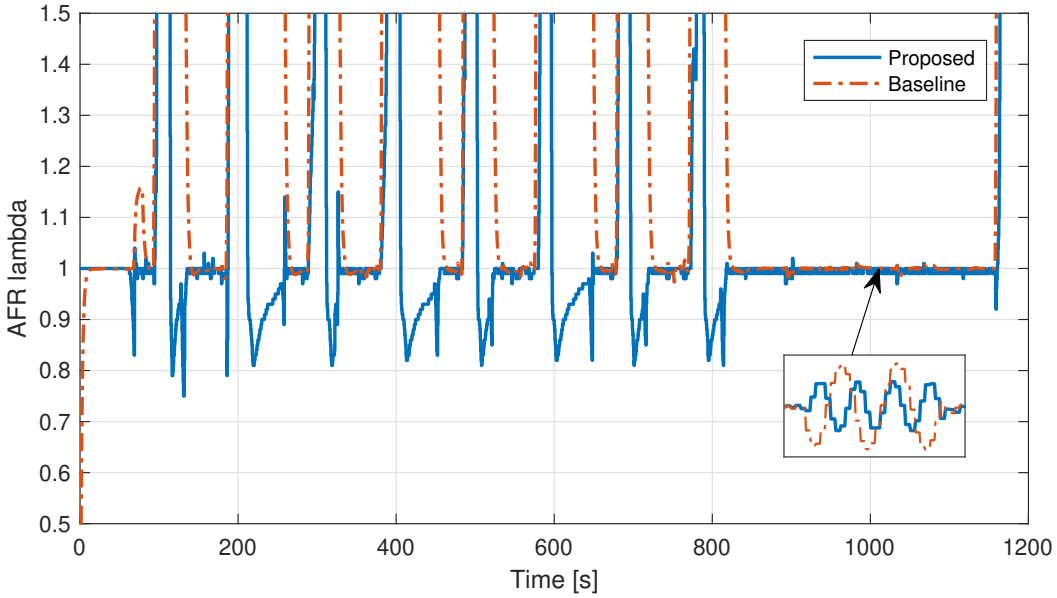

FIGURE 7.11. AFR lambda trajectories of the baseline and proposed control under the NEDC drive cycle.

Table 7.1: AFR error statistics.

AFR error	Baseline PID control	Proposed control
MSE	0.0020	0.0011
MAE	0.0376	0.0270
SD	0.0445	0.0333

To further show the effectiveness of the proposed control under fast varying engine operation speed, we also carried out dynamic tests, where a manually created engine speed chirp profile is used as the engine speed control command (with no engine stops). Compared with the NEDC speed profile in Fig. 7.7, the engine speed in Fig. 7.13 has faster variations, i.e. the speed oscillates within the range of 3,000 to 4,000 rpm with increasing frequency. This manually created engine speed evolution aims to test the AFR control transient response under fast engine dynamics variations and

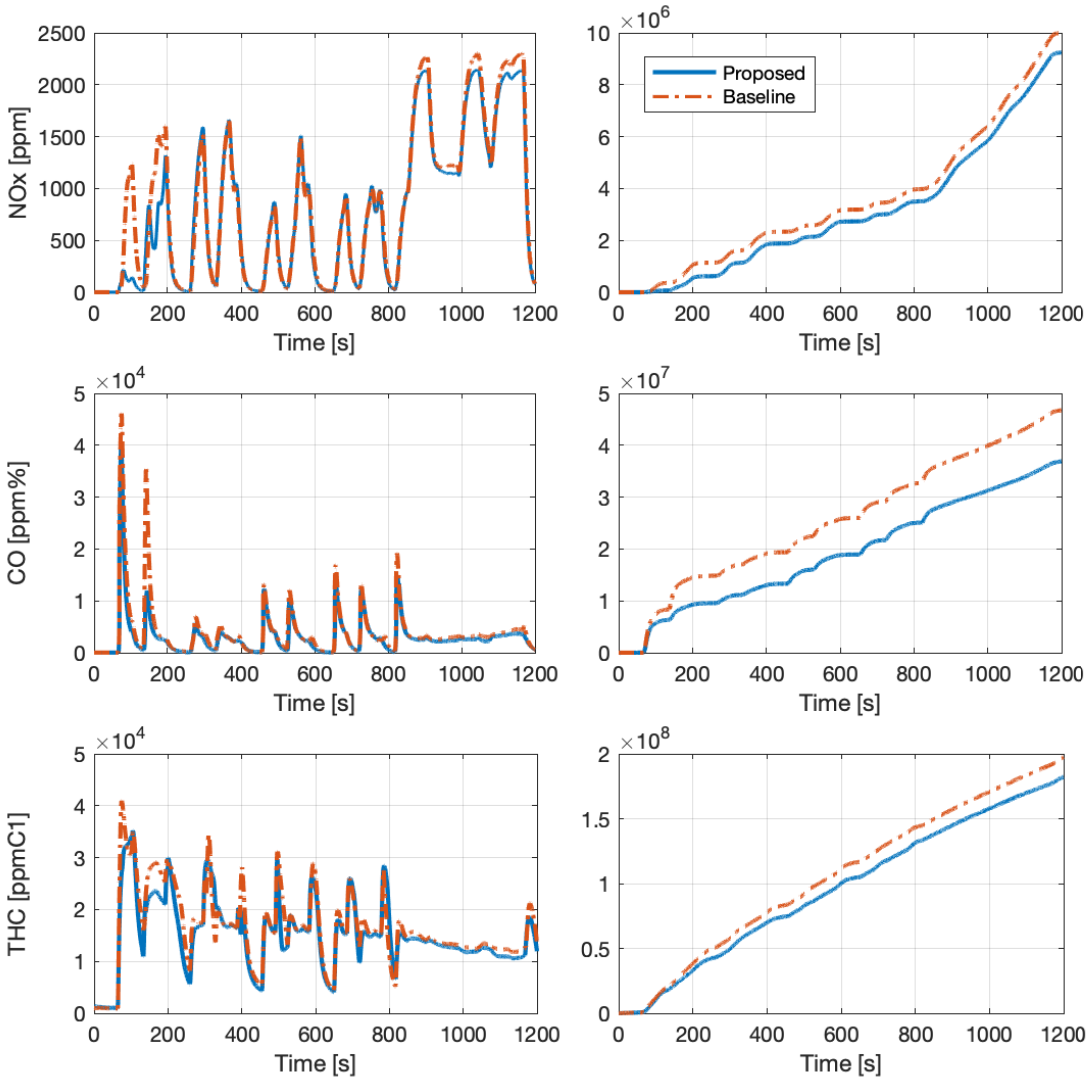

FIGURE 7.12. Emission responses of the baseline and proposed control under the NEDC drive cycle: real time (left) and cumulative (right) responses.

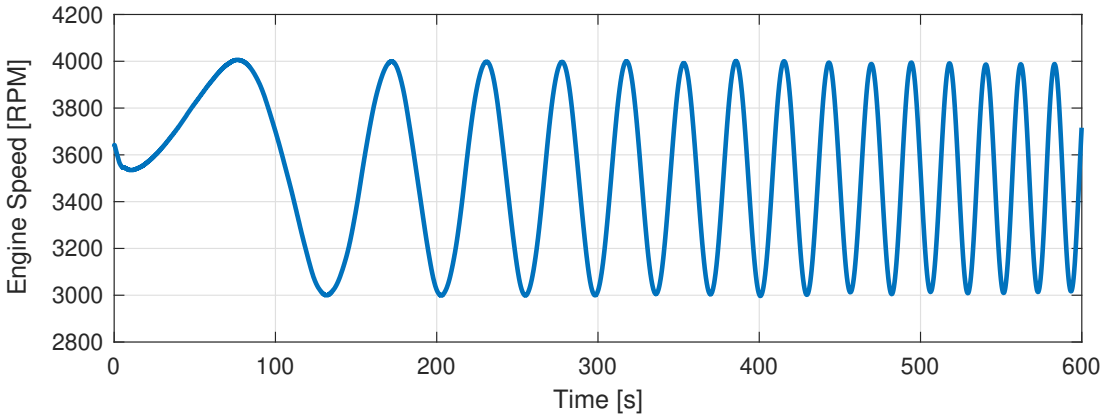

FIGURE 7.13. Engine speed chirp profile.

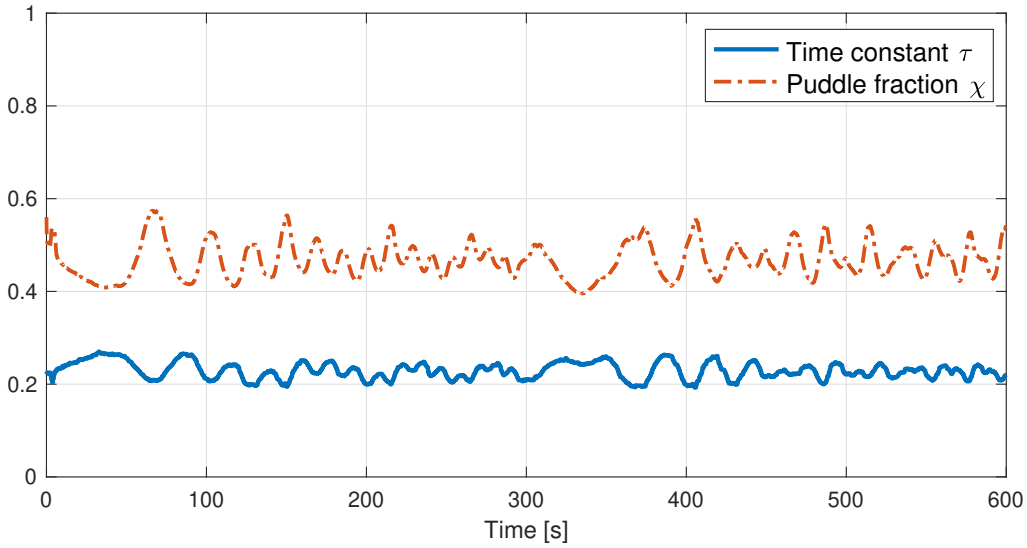

FIGURE 7.14. Fuel puddle dynamics parameter estimation.

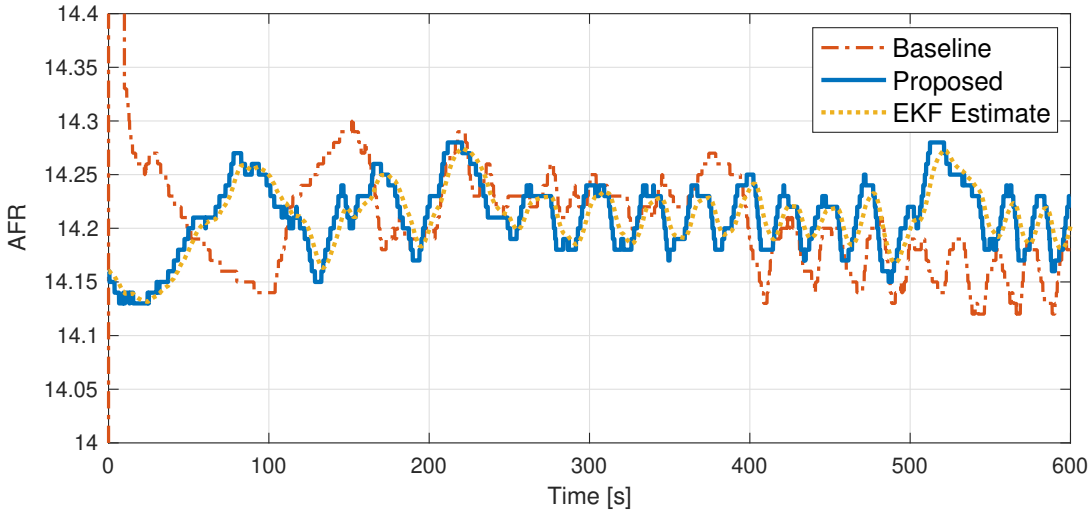

FIGURE 7.15. AFR and its estimation trajectories under the chirp speed drive cycle.

shows the ability of this proposed control to adapt to these fast variations. The fuel puddle dynamics estimation from the EKF is presented in Fig. 7.14. The corresponding AFR control responses of the baseline gain scheduling like PID control and the proposed composite control are depicted in Fig. 7.15. Fig. 7.16 and Table 7.1 describe the AFR error statistics (histograms and normal probability plots) of the two different control methods with respect to mean squared error (MSE), mean absolute error (MAE), and standard deviation (SD). Compared with the gain scheduling baseline PID control, the proposed control can achieve better AFR control response, e.g. over 25% improvement of the proposed control compared to the baseline control has been achieved in term of MAE. The proposed observers-based control leads to less fluctuation and reduced peak values in the AFR response when the engine changes the speed, showing the benefit of the use of nonlinear observers. Moreover, the proposed control does not rely on look-up tables, which can potentially reduce the cost of the engine calibration process.

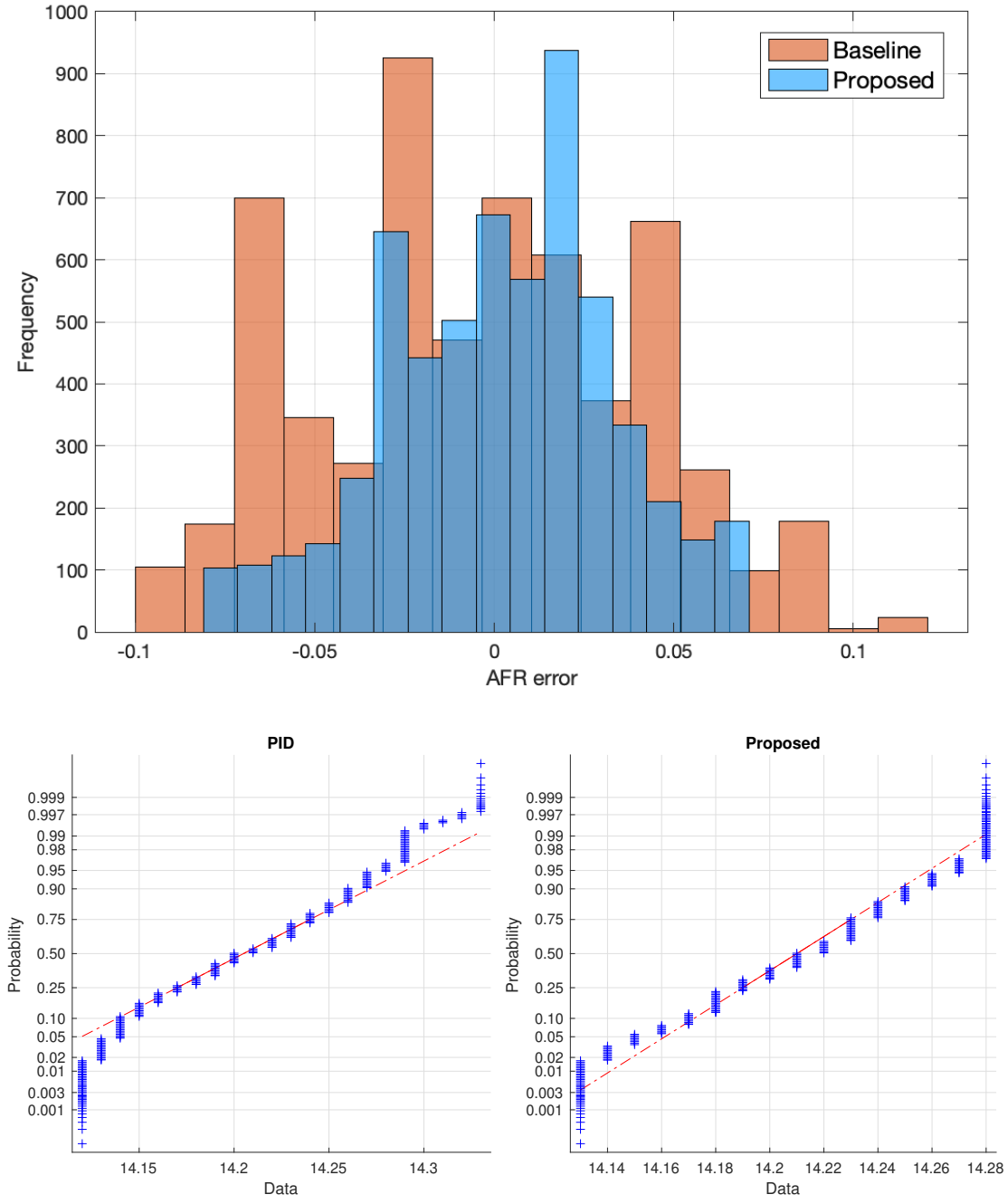

FIGURE 7.16. AFR control error statistics.

7.8 Conclusions

In this chapter, a dedicated model of Wankel engine dynamics was first developed for the design of different nonlinear observers and AFR control. The regulation of AFR was first reformulated into a fuel mass flow rate tracking problem. As a widespread issue for PFI engines, the nonlinear fuel puddle dynamics were online estimated using an extended Kalman filter by taking the unknown parameters as augmented states. The complex air-filling dynamics were lumped and estimated using novel observer techniques. Then a feedback control was designed combining the observers to stabilise the AFR. Comparative numerical simulations and practical engine tests validated that the proposed method can regulate the AFR around the stoichiometric value at both transients and steady states, where the newly proposed unknown input observer demonstrates better robustness against disturbances.

CHAPTER 8

AIR-FUEL RATIO CONTROL VIA Q-LEARNING: THEORY AND EXPERIMENTAL VALIDATION

The ordinary method for AFR regulation in commercial internal combustion engines is a PID controller, or more commonly, a PI controller. In the previous chapter, we have addressed the AFR control problem and its significance to the emissions, where we proposed a novel control method using nonlinear observers to compensate the effect of the model uncertainty and nonlinearity. However, the observer-based controller still is not optimal in the sense of minimising some objective function. This chapter will investigate the use of the proposed adaptive optimal control based on reinforcement learning for the AFR control problem in the aim of creating a learning-based controller that is model-free and learns the optimal control solution in real time.

8.1 Introduction

For the control of vehicle engines, nearly every field of current and classical control theory has been investigated. This chapter contributes to the already extensive liter-

ature on engine control in automobiles. The originality of this work stems from the following concept: Consider a control algorithm that is currently in use in a production vehicle. The algorithm is tuned for vehicle operation across the whole operating regime and is created according to particular requirements. In certain ways, the algorithm has been optimised for the engine in terms of performance, fuel efficiency, and tailpipe emissions, thanks to extensive research and development and calibration. The typical calibration and control processes in place today can be used to increase engine performance further through controller design. Use of the neural-network-based learning control design technique introduced in this chapter is an alternative to the standard approach.

The final result of our Q-learning process is a controller that has learned to provide optimal control signals under various operating conditions. We emphasise that such an adaptive optimal controller will be obtained after a specially designed learning process that performs approximate dynamic programming. Once a controller is learned and obtained (offline or online), it will be applied to perform the task of engine control. The performance of the controller can be further refined and improved through continuous learning in real-time vehicle operations. We note that continuous learning and adaptation to improve controller performance is one of the key promising attributes of the present approach. Continuous learning and adaptation for optimal individual engine performance over the entire operating regime and vehicle conditions would be desirable for future engine controller designs. One can often use offline engine data for initial simulation studies during the initial stage of the adaptive critic neural network learning. For practical reasons, we run the proposed controller in real time to test the learning ability in actual scenarios.

8.2 AFR Control Problem Formulation

It has been shown in Chapter 7 that the AFR regulation problem can be formulated into a fuel flow tracking problem as the feedback control error e used in the AFR

controller can be defined as

$$e = \dot{m}_{fd} - \dot{m}_f = \frac{1}{\lambda_d}\dot{m}_a - \dot{m}_f \tag{8.1}$$

where $\dot{m}_{fd}$ and λ_d are the desired fuel mass flow rate and AFR. Thus, its derivative is calculated as

$$\dot{e} = \frac{1}{\lambda_d}\ddot{m}_a - \ddot{m}_f \tag{8.2}$$

Clearly, $\ddot{m}_a$ is the derivative of complex nonlinear air-filling dynamics (5.5) and $\ddot{m}_f$ is the derivative of (5.6) with fuel puddle dynamics (5.7). The dynamics of $\ddot{m}_a$ and $\ddot{m}_f$ is unknown in models and not measurable in practice. Chapter 7 employed nonlinear observers to estimate the dynamics of $\ddot{m}_a$ and $\ddot{m}_f$ for the AFR control design. The error dynamics were formulated as

$$\dot{e} = M - u_d \tag{8.3}$$

Based on this formulation, we further extend the idea using the structure of a PI controller, i.e. we let the control input u_d be in the form

$$u_d = -k_p^* e - k_i^* e_i \tag{8.4}$$

with $k_p^* > 0$ and $k_i^* > 0$ being the optimal proportional and integral gain.

By defining the integral error

$$e_i = \int_0^t e(\tau)d\tau \tag{8.5}$$

we can define an augmented error state as

$$E = \begin{bmatrix} e_i \\ e \end{bmatrix} \tag{8.6}$$

Considering

$$\dot{e} = -k_p^* e - k_i^* e_i + M \tag{8.7}$$

so that the error state equation is

$$\dot{E} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} E + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u^* + M \tag{8.8}$$

where u^* is the optimal PI control

$$u^* = u_d = \begin{bmatrix} -k_i^* & -k_p^* \end{bmatrix} E = -K^* E \tag{8.9}$$

Now we formulate the control problem in the following.

For the error state system

$$\dot{E} = AE + Bu + M \tag{8.10}$$

with $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ and $B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, the objective is to find the optimal control $u = u^*$ such that an infinite-horizon integral cost

$$V(E) = \int_t^\infty r(E,u) d\tau \tag{8.11}$$

is minimised, where the utility $r(t) = E^T S E + R u^2$ with positive definite S and R.

Remark 8.1 *Recall M is the lumped unknown nonlinear term defined in Chapter 7. This term can be estimated via the unknown input observer so it will be treated as a known variable in the following control development.* ◇

8.3 Adaptive Optimal AFR Control Design

This section presents the design of the adaptive optimal controller for the AFR. Fig. 8.1 presents a schematic diagram of the proposed Q-learning-based AFR control system.

8.3.1 Parameterisation of Nonlinear Q-function

Recalling the idea of Q-learning in Chapter 3, we create an action-dependent version of value function $Q(E,u)$: such that $Q^*(E,u^*) = V^*(E)$. For the continuous-time non-

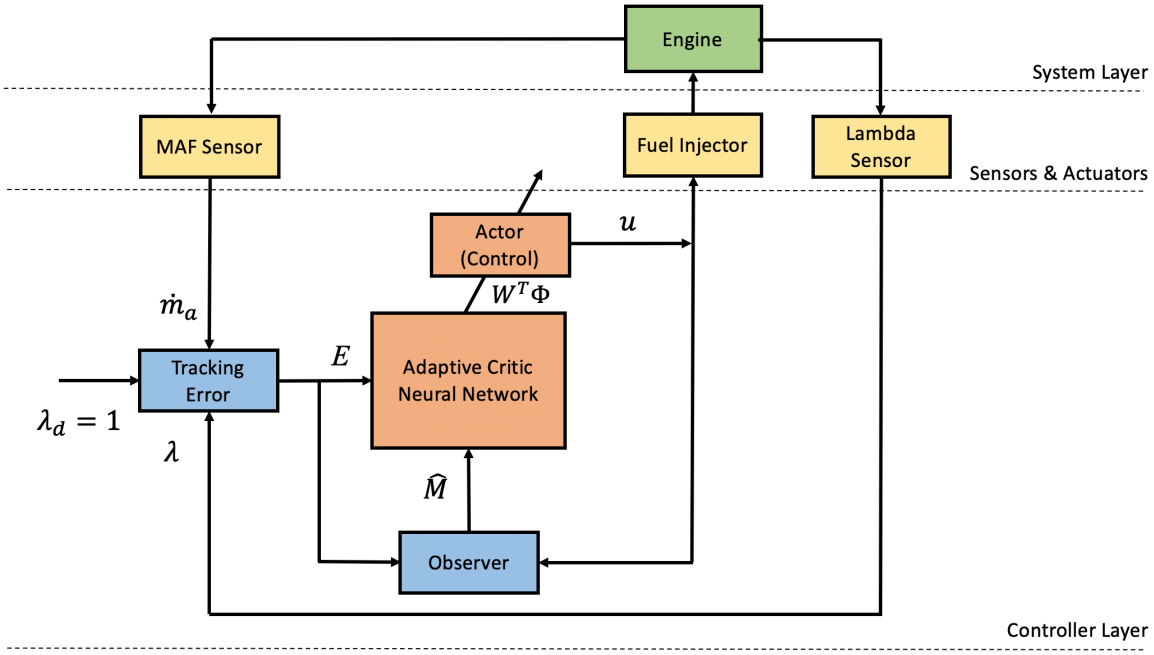

FIGURE 8.1. A schematic diagram of the proposed Q-learning-based AFR control system.

linear input-affine system (8.10), the Q-function can be explicitly defined by adding the Hamiltonian onto the optimal value V^* as

$$\begin{aligned} Q(E,u) &:= V^*(E) + \mathcal{H}(E,u,\nabla V^*) \\ &= \underbrace{V^*(E) + S(E) + (\nabla V^*)^{\mathsf{T}} AE + M}_{F_{EE}(E)} + \\ &\underbrace{(\nabla V^*)^{\mathsf{T}} Bu}_{F_{Eu}(E,u)} + \underbrace{Ru^2}_{F_{uu}(u)} \end{aligned} \tag{8.12}$$

where $F_{EE}(E)$, $F_{Eu}(E,u)$, and $F_{uu}(u)$ are the lumped terms that can be approximated respectively via neural networks.

Lemma 8.1 *The Q-function defined in* (8.12) *is positive definite with the optimisation scheme* $Q^*(E,u^*) = \min_u Q(E,u)$. *The optimal Q-function* $Q^*(E,u^*)$ *has the same optimal value* $V^*(E)$ *as for the value function* $V^u(E)$ (8.11), *i.e.* $Q^*(E,u^*) = V^*(E)$ *when applying the*

optimal control u^*. ◇

Proof. See *Lemma 3.1* in Chapter 3 □

8.3.2 Adaptive Critic for Q-function Approximation

We approximate the Q-function (8.12) using a critic neural network by

$$Q(E,u) = W^\top \Phi(E,u) + \varepsilon_Q(E,u) \tag{8.13}$$

where $\Phi(E,u) \in \mathbb{R}^n$ denotes the activation function vector with the number n of neurons in the hidden layer; $W \in \mathbb{R}^n$ is the weight vector; $\varepsilon_Q(E,u)$ is the neural network approximation error; and $W^\top \Phi(E,u)$ can be explicitly expressed according to the three components $F_{EE}(E)$, $F_{Eu}(E,u)$, and $F_{uu}(u)$ in (8.12) as

$$W^\top \Phi(E,u) = \begin{bmatrix} W_{EE}^\top & W_{Eu}^\top & R \end{bmatrix} \begin{bmatrix} \Phi_{EE}(E) \\ \Phi_{Eu}(E)u \\ \Phi_{uu}(u) \end{bmatrix} \tag{8.14}$$

where $\Phi_{EE} \in \mathbb{R}^{n_{EE}}$, $\Phi_{Eu} \in \mathbb{R}^{n_{Eu}}$ and $\Phi_{uu} = u^2$. The regressor $\Phi(E,u)$ is selected to provide a complete independent basis such that $Q(E,u)$ is uniformly bounded with $n = (n_{EE} + n_{Eu} + 1)$. Recalling from the Weierstrass higher-order approximation theorem ([164]), the approximation error $\varepsilon_Q(E,u)$ is bounded for a fixed n within a compact set Ω and as the number of neurons $N_{EE} \to \infty$ and $N_{Eu} \to \infty$, i.e., $n \to \infty$, we have $\varepsilon_Q(E,u) \to 0$.

One needs to derive the Bellman equation in terms of the Q-function to update the critic. By Bellman's principle of optimality ([41]), we have the following optimality equation

$$V^*(E(t-T)) = \int_{t-T}^{t} r(E(\tau), u(\tau))\mathrm{d}\tau + V^*(E(t)) \tag{8.15}$$

The result from *Lemma 8.1* showed that $Q^*(E,u^*) = V^*(E)$, which means we can rewrite (8.15) in terms of $Q^*(E,u^*)$ as

$$
\begin{aligned}
&\overbrace{-\int_{t-T}^{t} r(E,u)\mathrm{d}\tau}^{-\rho(E,u)} = Q^*(E(t),u^*(t)) \\
&\qquad\qquad - Q^*(E(t-T),u^*(t-T)) \\
&= \underbrace{W^\mathsf{T}\Phi(E(t),u^*(t)) - W^\mathsf{T}\Phi(E(t-T),u^*(t-T))}_{W^\mathsf{T}\Delta\Phi(E,u^*)} \\
&\quad + \varepsilon_{BQ}(E,u)
\end{aligned} \tag{8.16}
$$

with the integral reinforcement $\rho(E,u)$, the difference $\Delta\Phi(t) = \Phi(E(t),u^*(t)) - \Phi(E(t-T),u^*(t-T))$, and the Bellman equation residual errors $\varepsilon_{BQ} = \varepsilon_Q(E(t),u^*(t)) - \varepsilon_Q(E(t-T),u^*(t-T))$ being bounded for bounded ε_Q. Define two auxiliary variables $\mathcal{P} \in \mathbb{R}^{n\times n}$ and $\mathcal{Q} \in \mathbb{R}^n$ by low-pass filtering the variables in (8.16) as

$$
\begin{cases}
\dot{\mathcal{P}} = -\ell\mathcal{P} + \Delta\Phi(t)\Delta\Phi(t)^\mathsf{T}, \;\; \mathcal{P}(0) = 0 \\
\dot{\mathcal{Q}} = -\ell\mathcal{Q} + \Delta\Phi(t)\rho(E,u), \quad \mathcal{Q}(0) = 0
\end{cases} \tag{8.17}
$$

with a filter parameter $\ell > 0$.

The adaptive critic neural network can be written as

$$
\hat{Q}(E,u) = \hat{W}^\mathsf{T}\Phi(E,u) \tag{8.18}
$$

where $\hat{W}$ and $\hat{Q}(E,u)$ denote the current estimate of W and $Q(E,u)$, respectively.

Now we design the adaptation law using the sliding mode technique to update $\hat{W}$ such that

$$
\dot{\hat{W}} = -\Gamma\mathcal{P}\frac{M'}{||M'||} \tag{8.19}
$$

where $M' \in \mathbb{R}^n$ is defined as $M' = \mathcal{P}\hat{W} + \mathcal{Q}$ and $\Gamma \succ 0$ is a diagonal adaptive learning gain to be tuned.

Lemma 8.2 *Given the adaptation law* (8.19)*, if $u(t)$, $\Delta\Phi(t)$, and the system states $E(t)$ are persistently excited, the estimation error of weight $\tilde{W} = W - \hat{W}$ will converge to a compact set in finite time.* ◇

Proof. See *Lemma 3.2* in Chapter 3. □

8.3.3 Adaptive Optimal Control via Q-learning

We reconstruct the optimal control u^* from (6.11) based on the parameterisation of $Q(E,u)$ (8.12) such that

$$u^* = -\frac{1}{2}R^{-1}W_{Eu}^{\top}\Phi_{Eu}E + \varepsilon_{Qu} \tag{8.20}$$

where ε_{Qu} and ε_{Eu} are bounded approximation errors due to ε_Q and ε_E, $W_{Eu}^{\top}\Phi_{Eu}E$ accounts for the term $BE^{\top}\nabla V_x^*$, and W_{uu} is essentially predefined R (see (8.12)). Therefore, one can determine the optimal control directly using the adaptive critic (8.18) if the weight $\hat{W}$ converges to the actual weight W. The control law (actor) will be

$$u = -\frac{1}{2}R^{-1}\hat{W}_{Eu}^{\top}\Phi_{Eu}E \tag{8.21}$$

We summarise the result for this Q-learning algorithm as

Theorem 8.1 *Given the engine system* (8.10) *with the value function* (8.11) *and Q-function* (8.12)*, the adaptive critic neural network* (8.18) *with the adaptation law* (8.19) *and the actor* (8.21) *form an adaptive optimal control so that the adaptive critic weight estimation error* $\tilde{W}$ *will converge to a compact set and the control input (the actor)* u *will converge to a small bounded set around its optimal control solution* u^* *in finite time.*

Proof. See *Theorem 3.1* in Chapter 3. □

8.4 Experimental Results and Discussion

The engine test-rig described in Chapter 7 is used as the experimental platform to validate the efficacy of the proposed AFR control.

In the experiment, the engine speed was controlled through an available speed control. In order to satisfy the PE condition, the augmented error state E needs to be

persistently excited. In this case, instead of regulating the AFR to a constant stoichiometric value, one needs to excite the AFR response to a certain level. Hence, a chirp speed profile is designed between 3,000 to 4,000 [RPM] with increasing and decreasing frequency to create a dynamic oscillating operating condition, while the throttle is operated manually within the range of 0 - 15 [^{o}] to achieve high-frequency variation. Fig. 8.2 presents the engine speed chirp profile and Fig. 8.3 shows the manually-operated throttle angle while the air mass flow rate from the MAF sensor and the AFR response from the lambda sensor are presented in Fig. 8.4 and 8.5, respectively. It can be found that the air mass flow rate trajectory follows a similar trend to the throttle angle. The AFR lambda oscillates around $\lambda_d = 1$. The high magnitude of the lambda transient response in the initial 50 [s] could be explained by the learning process of the adaptive critic.

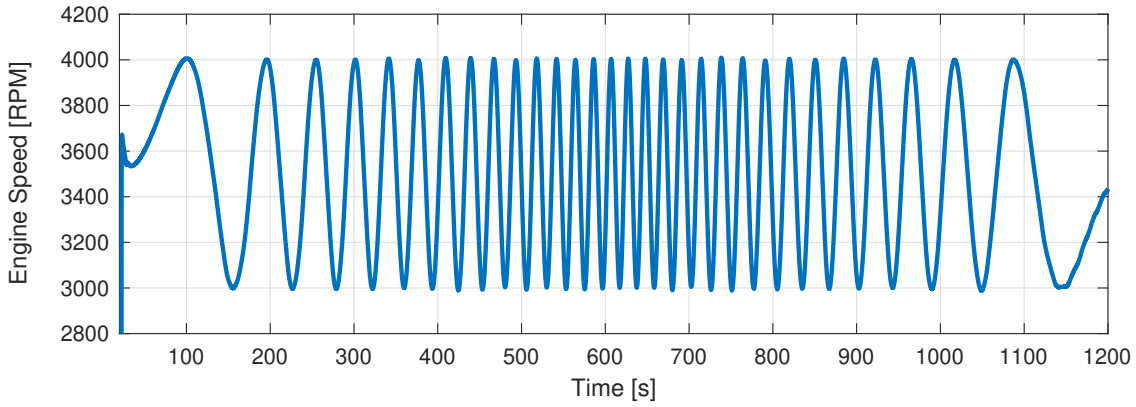

FIGURE 8.2. Engine speed chirp profile.

The aim is to implement the Q-learning algorithm for the adaptive optimal AFR controller. However, the experimental result would be difficult to verify because the ideal weight of the adaptive critic neural network is unknown in practice and the optimal control is not available. One way to verify the effectiveness of the proposed adaptive optimal AFR control (Q-learning) is by comparing the control signal with an optimal control response. Since we already know the system dynamics A and B, it is feasible to use the GPI algorithm (see chapter 3) first to get an optimal response and then

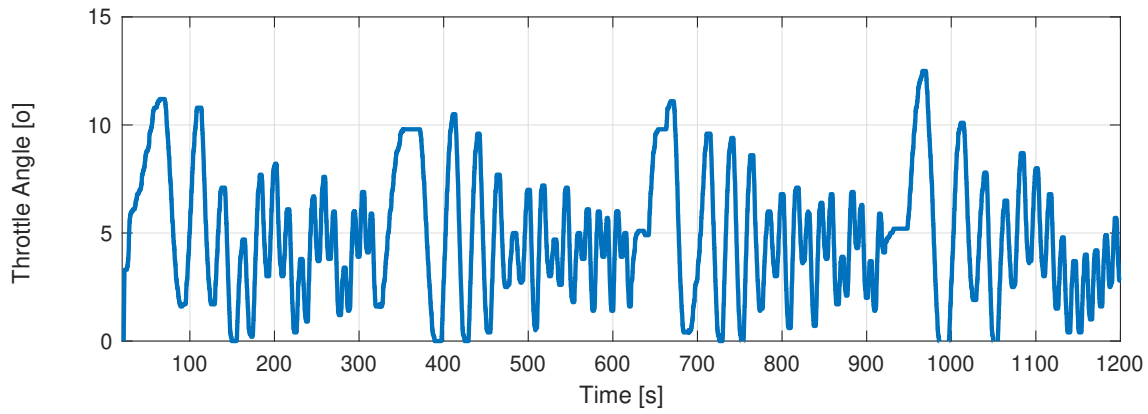

FIGURE 8.3. Manually-operated throttle angle profile.

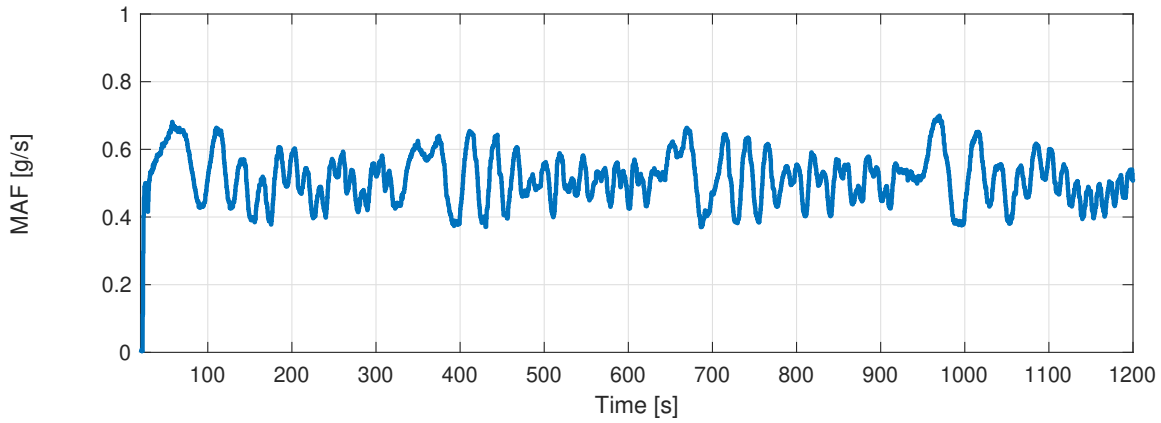

FIGURE 8.4. Air mass flow rate MAF sensor response.

compare it with the response of the Q-learning controller. If the proposed Q-learning controller is optimal, the responses should be close.

The adaptive optimal AFR controller based on GPI is implemented as follows. We choose the value function as (6.7) with $S = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$ and $R = 1$. The activation function $\Phi(E,u)$ of the adaptive critic neural network (8.14) is selected as $\Phi(E,u) = [e_i^2 \;\; e_i e \;\; e^2]^\mathsf{T}$ with the number of neurons $n = 3$. The tuning parameters are chosen as such: the sample period $T = 1$s, the filter parameter $\ell = 1$, the adaptive learning gain $\Gamma = 7$.

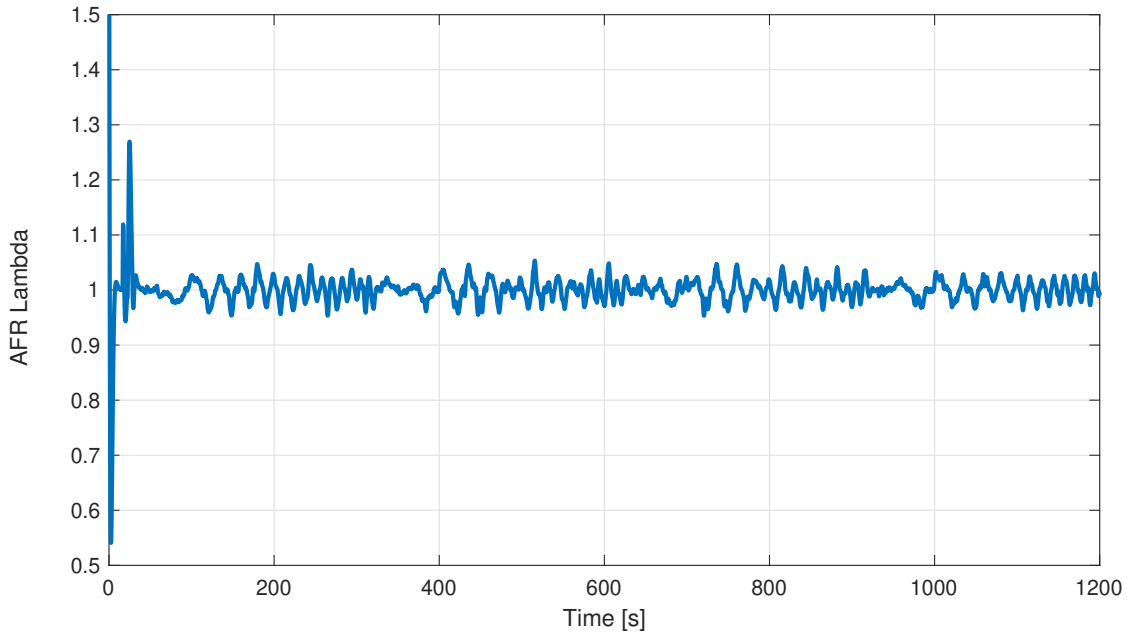

FIGURE 8.5. AFR lambda sensor response.

Fig. 8.6 and 8.7 present the augmented error state trajectories and the adaptive critic weight convergence of the GPI controller.Using the "LQR" command in Matlab to solve the ARE with the given A and B, the solution P is

$$P = \begin{bmatrix} 1.7321 & 1.0000 \\ 1.0000 & 1.7321 \end{bmatrix} \tag{8.22}$$

This is close to the weight convergence result shown in Fig. 8.7, i.e. $W_1 \approx W_2 \approx P_{11} = P_{44}$ and $W_2 \approx P_{12}+P_{21}$. The small difference to the ideal value can be explained by the term estimation error due to the unknown input observer for the lumped term M.

Then, we implement the adaptive optimal AFR controller based on Q-learning as follows. The activation function $\Phi(x,\alpha)$ of the adaptive critic neural network is selected as $\Phi(E,u) = [e_i^2 \;\; e_i e \;\; e^2 \;\; e_i u \;\; eu \;\; e_i eu \;\; e_i^2 u \;\; e^2 u \;\; e_i^2 eu \;\; e_i e^2 u \;\; u^2]^\mathsf{T}$ with the number of neurons $n = 11$. The tuning parameters are chosen as such: the sample period $T = 1$s, the filter parameter $\ell = 1$, the adaptive learning gain $\Gamma = 5$. Fig. 8.8 presents the adaptive

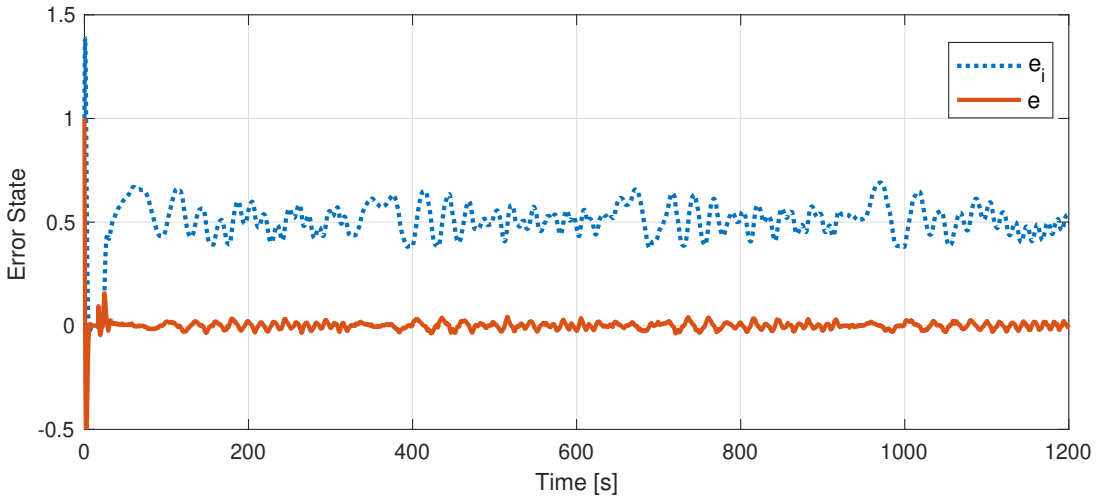

FIGURE 8.6. Error state trajectories.

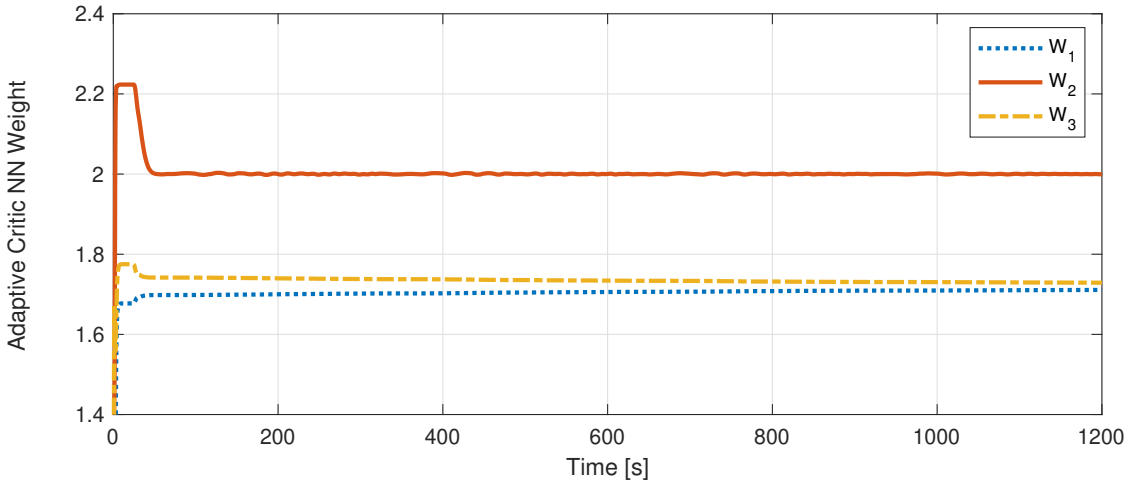

FIGURE 8.7. Convergence of the adaptive critic neural network of the GPI controller.

critic weight convergence of the Q-learning controller, where the weights converge around 600 [s]. Fig. 8.9 compares the control signal of the two controllers. It can be found that the control signal of the Q-learning controller starts to approach to that of the GPI controller after 600 [s]. This can be explained by the weight convergence of the Q-learning adaptive critic after 600 [s] in Fig. 8.8.

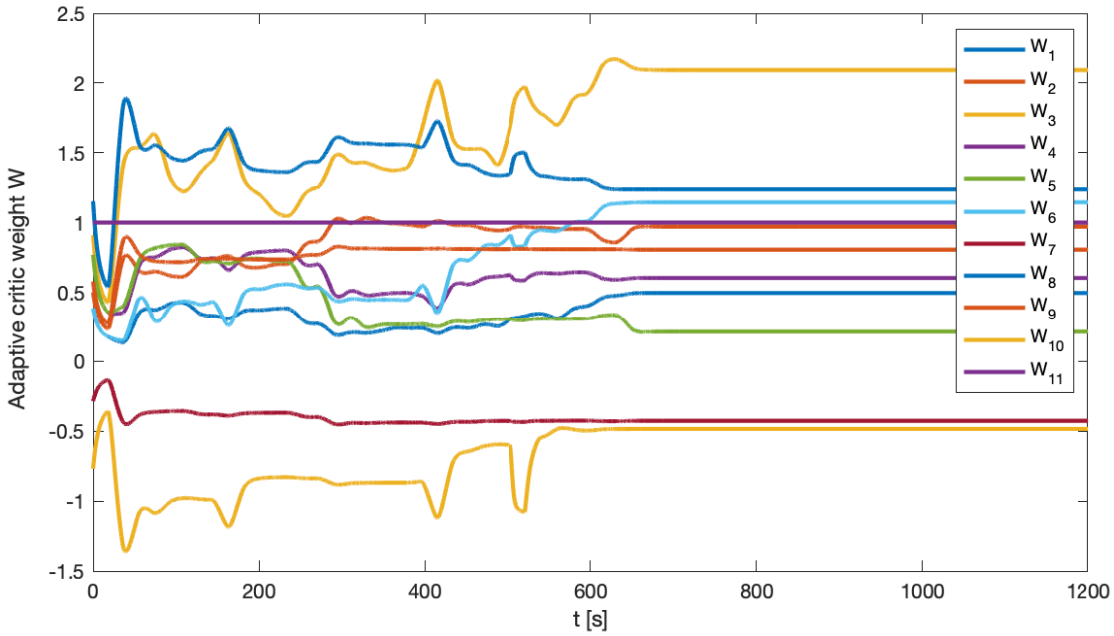

FIGURE 8.8. Convergence of the adaptive critic neural network of the Q-learning controller.

8.5 Conclusions

In this chapter, we have proposed an adaptive optimal AFR controller using reinforcement learning principles, i.e. Q-learning. The controller was formulated using a PI control structure and the optimal control can be obtained after a learning process that performs reinforcement learning. The proposed controller was implemented on a practical Wankel engine (the same experimental set-up in Chapter 7). The results were verified by comparing the results to a GPI controller. Continuous learning and adapta-

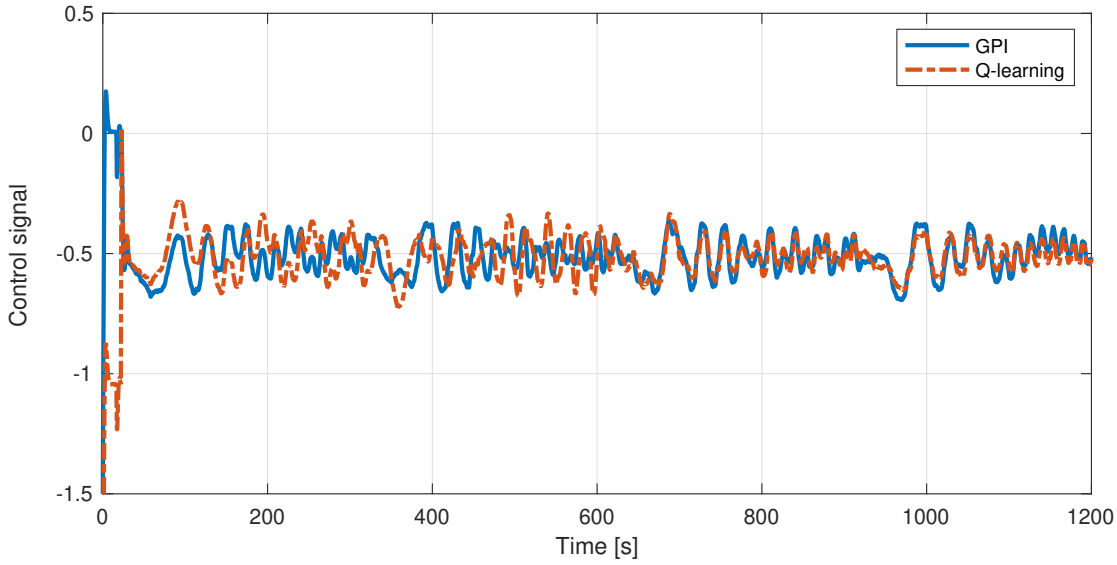

FIGURE 8.9. Comparison between the control signals of two adaptive optimal controllers: GPI and Q-learning.

tion to improve controllers performance is one of the key promising attributes of the proposed approach. The new controllers are in the form of online learning algorithms that can solve the nonlinear optimal control/observer design problems without the knowledge of the system dynamics. The performance of the controller can be further refined and improved through continuous learning in real-time vehicle operations.

CHAPTER 9

CONCLUSION

In this thesis, we have developed new approaches of adaptive optimal control via reinforcement learning, i.e. GPI and Q-learning. The new controllers are in the form of online learning algorithms that can solve the nonlinear optimal control/observer design problems without knowledge or with incomplete knowledge of the system dynamics. The thesis considered the application to automotive engine control problems, i.e. idle speed control and AFR control. A 225CS Wankel rotary engine was targeted for control system development, where the process included dynamics modelling, calibration, control design/simulation, implementation, and practical experiments. The proposed control methods have been successfully applied to an engine system, which can also be applied to other complex engineering systems with proper adjustment.

9.1 Summary of Achievements

1. In Chapter 3, a new approach of the adaptive optimal control theory is established which forms the major contribution of the theoretical research. Two online adaptive optimal control algorithms are proposed based on reinforcement learning for

continuous-time nonlinear input-affine systems: 1) generalised policy iteration (GPI) and 2) Q-learning. The adaptive critic and actor are continuously and simultaneously updating each other with neither iterative steps nor an initial stabilising policy. The two approaches can online approximate the value functional/Q-functional and are partially/completely model-free. The new adaptive design enables the online verification of the persistent excitation (PE) condition and guarantees the overall closed-loop stability and the finite-time convergence. A detailed mathematical analysis and numerical simulations are provided to show the effectiveness of the algorithms.

2. In Chapter 4, the adaptive optimal control results are extended to an optimal observer design problem. An online continuous-time Q-learning algorithm is proposed to solve the optimal observer design problem online while ensuring stability and optimality. We show that the optimal solution can be obtained by approximately solving an observer Hamilton-Jacobi-Bellman (OHJB) equation. The Q-functional is approximated by an adaptive critic neural network that solves the Q-learning Bellman equation online. A case study on observer design for the Van der Pol oscillator is provided. Numerical simulations demonstrate the effectiveness of the proposed algorithm compared with the high-gain observer. Beyond the observer design, it is noted that the proposed results can be easily extended to general stabilisation and tracking control problems.

3. In Chapter 5, a set of control-oriented models are developed for a 225CS Wankel rotary engine produced by Advanced Innovative Engineering (AIE) UK Ltd. Through a synthesis approach that involves State Space (SS) principles and the artificial Neural Networks (NN), the Wankel engine models are derived by leveraging both first-principle knowledge and engine test data. By using either (or both) physical knowledge or test data, the developed models are able to describe the Wankel engine dynamics with acceptable accuracy. They are all control-oriented models that have less computational demand and should be able to run faster than the available CFD models due to their simplicity.

4. In Chapter 6, an output feedback controller is developed based on reinforcement learning for the idle speed regulation problem. The proposed controller is completely model-free and able to learn the optimal control solution online in finite time using only the measurable outputs. The regulation of idle speed can be formulated as an optimal control problem that minimises a pre-defined value function by actuating the throttle angle. Then, we incorporate the extended Kalman filter (EKF) as an optimal reduced-order state observer, which enables the online estimation of the unknown fuel puddle dynamics, to achieve an output feedback idle speed controller. The overall Lyapunov stability is proved and the simulation results of a benchmark engine demonstrate that the proposed controller can effectively regulate the idle speed to a set point under certain load disturbance.

5. In Chapter 7, a generic observer-based air-fuel ratio (AFR) control framework for automotive engine systems is presented. The complex nonlinear air-filling dynamics are lumped together and estimated using novel observer techniques. A newly-proposed unknown input observer is compared with a differentiation observer and then employed in the feedback AFR control design. Comparative simulations and practical experimental results compared to a benchmark PID controller show that the proposed control can speed up the transient response and regulate the AFR around the stoichiometric value. Moreover, the proposed controller no longer relies on look-up tables, which can potentially avoid the cost of engine calibration process.

6. In Chapter 8, an adaptive optimal AFR controller is proposed based on Q-learning which can learn to provide optimal control signals under various operating conditions. We emphasise that such an adaptive optimal controller will be obtained after a specially designed learning process that performs approximate dynamic programming. Once a controller is learned and obtained (offline or online), it will be applied to perform the task of engine control. The performance of the controller can be further refined and improved through continuous learning in real-time vehicle operations.

We note that continuous learning and adaptation to improve controllers performance is one of the key promising attributes of the present approach. The new controllers are in the form of online learning algorithms that can solve the nonlinear optimal control/observer design problems without partial/complete knowledge of the system dynamics. Although the application focus of the thesis is on a Wankel rotary engine, the proposed control methods can also be applied to general internal combustion engines and other complex engineering systems.

9.2 Key Conclusions

This Ph.D. research have covered both theoretical and practical sides of the subject. We summarise the research conclusions into three points:

1. A new adaptive optimal control scheme was formulated for continuous-time nonlinear systems to tackle 1) optimal control and 2) optimal observation problems.

2. New adaptive optimal control algorithms were developed using reinforcement learning that bring benefits such as adaptation, optimality, model-free learning: 1) GPI and 2) Q-learning.

3. Three new control systems for automotive engines were designed, implemented, and validated: 1) Q-learning-based idle speed control, 2) observer-based AFR control, and 3) Q-learning-based AFR control.

9.3 Future Work

We have considered some of the directions for the continuation of this work.

1. The idea of online adaptive optimal control can be extended to solve zero and non-zero sum games. For example, the two-player zero-sum game problem can be

viewed as a robust adaptive optimal control problem, where the controller is the minimising player and the disturbance is the maximising player. Similar to the approximate solution to the HJB equation, the Q-learning algorithm proposed in this thesis can be extended for generating a local approximate optimal Nash solution of the Hamilton-Jacobi-Issacs (HJI) equation. This idea has recently led to a paper "A Q-learning approach for the two-player zero-sum game problem for completely unknown continuous-time nonlinear systems" to be submitted to the 2022 IEEE Conference on Decision and Control (CDC).

2. Most existing literature on adaptive optimal control or ADP relies results on full-state feedback framework, where all the system states are assumed to be available. This is often not true in practice. Output feedback control eliminates the need for full state feedback and makes the design more practical. In this thesis, we have shown in Chapter 6 a version of adaptive optimal output feedback control by combining the EKF with the Q-learning-based control. It is possible to synthesise an output feedback adaptive optimal control framework by combining the adaptive optimal observer (Chapter 4) and the adaptive optimal controller (Chapter 3), i.e. full-state feedback with the reconstructed states estimated from an observer. Another different approach to achieve adaptive optimal output feedback is the direct method that investigates the online optimal solution of the output feedback gain K for the control input u and the output y:

$$u = -Ky \tag{9.1}$$

For example, the optimal K is subject to a set of coupled matrix equations if the problem is linear quadratic (see Chapter 8.1 in [30]). This can be a starting point to apply our Q-learning algorithm.

3. We have been looking at the potential application beyond the Wankel rotary engine. The results can be extended to other automotive control problems. For example, the proposed adaptive optimal controller can be applied to power management of hybrid

electric vehicles (HEVs), i.e. the design of a higher-level control algorithm that determines the proper power level to be generated, and its split between the two power sources (the engine and the battery). Commonly, a static optimisation or dynamic programming scheme is used to figure out the proper split between the two power sources using steady-state efficiency maps [226][227][228]. Our Q-learning method has the potential to be used for real-time optimal control for power management with the feature of continuous learning and adaptation.

BIBLIOGRAPHY

[1] R. S. Sutton, A. G. Barto, *et al.*, *Reinforcement learning: An introduction*.
MIT Press, 1998.

[2] A. S. Chen, G. Vorraro, M. Turner, R. Islam, G. Herrmann, S. Burgess, C. Brace, J. Turner, and N. Bailey, "Control-oriented modelling of a Wankel rotary engine: a synthesis approach of state space and neural networks," tech. rep., SAE Technical Paper 2020-01-0253, 2020.

[3] A. S. Chen, *Torque Estimation and Air-Fuel Ratio Control for Internal Combustion Engines*.
MSc thesis, University of Bristol, 2016.

[4] T. J. Norman, *A performance model of a spark ignition Wankel engine: including the effects of crevice volumes, gas leakage, and heat transfer*.
PhD thesis, Massachusetts Institute of Technology, 1983.

[5] S. Russell and P. Norvig, "Artificial intelligence: a modern approach," *URI: http://196.43.179.3:8080/xmlui/handle/123456789/655*.

[6] C. Wilson, F. Marchetti, M. Di Carlo, A. Riccardi, and E. Minisci, "Classifying intelligence in machines: a taxonomy of intelligent control," *Robotics*, vol. 9, no. 3, p. 64, 2020.

[7] A. Bensoussan, Y. Li, D. P. C. Nguyen, M.-B. Tran, S. C. P. Yam, and X. Zhou, "Machine learning and control theory," *arXiv preprint arXiv:2006.05604*, 2020.

[8] R. C. Dorf and R. H. Bishop, *Modern control systems*.
Pearson, 2011.

[9] K. Ogata and Y. Yang, *Modern control engineering*, vol. 4.
Prentice hall India, 2002.

[10] B. C. Kuo, *Automatic control systems*.
Prentice Hall PTR, 1987.

[11] I. Nagrath, *Control systems engineering*.
New Age International, 2006.

[12] N. S. Nise, *CONTROL SYSTEMS ENGINEERING, (With CD)*.
John Wiley & Sons, 2007.

[13] S. Skogestad and I. Postlethwaite, *Multivariable feedback control: analysis and design*, vol. 2.
Wiley New York, 2007.

[14] W. S. Levine, *The control handbook: Control system fundamentals*.
CRC press, 2010.

[15] K. Dutton, S. Thompson, and B. Barraclough, *The art of control engineering*.
Addison Wesley Harlow, 1997.

[16] G. F. Franklin, J. D. Powell, A. Emami-Naeini, and J. D. Powell, *Feedback control of dynamic systems*, vol. 3.
Addison-Wesley Reading, MA, 1994.

[17] K. Vamvoudakis and S. Jagannathan, *Control of Complex Systems: Theory and Applications*.
Butterworth-Heinemann, 2016.

[18] H. K. Khalil, "Noninear systems," *Prentice-Hall, New Jersey*, vol. 2, no. 5, pp. 5–1, 1996.

[19] J.-J. E. Slotine, W. Li, *et al.*, *Applied nonlinear control*, vol. 199.
Prentice Hall Englewood Cliffs, NJ, 1991.

[20] R. Lozano and X.-H. Zhao, "Adaptive pole placement without excitation probing signals," *IEEE Transactions on Automatic Control*, vol. 39, no. 1, pp. 47–58, 1994.

[21] K. B. Ariyur and M. Krstic, *Real-time optimization by extremum-seeking control*.
John Wiley & Sons, 2003.

[22] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96–114, 2006.

[23] M. Krstic, I. Kanellakopoulos, P. V. Kokotovic, *et al.*, *Nonlinear and adaptive control design*, vol. 222.
Wiley New York, 1995.

[24] S. Sastry and M. Bodson, *Adaptive control: stability, convergence and robustness*.
Courier Corporation, 2011.

[25] K. J. Åström and B. Wittenmark, *Adaptive control*.
Courier Corporation, 2013.

[26] I. D. Landau, R. Lozano, M. M'Saad, and A. Karimi, *Adaptive control: algorithms, analysis and applications*.
Springer Science & Business Media, 2011.

[27] P. A. Ioannou and J. Sun, *Robust adaptive control*, vol. 1.
PTR Prentice-Hall Upper Saddle River, NJ, 1996.

[28] N. Hovakimyan and C. Cao, *L1 adaptive control theory: guaranteed robustness with fast adaptation*, vol. 21.
SIAM-Society for Industrial and Applied Mathematics, 2010.

[29] P. Ioannou and B. Fidan, *Adaptive control tutorial*, vol. 11.

Siam, 2006.

[30] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal control*.
John Wiley & Sons, 2012.

[31] A. E. Bryson, *Applied optimal control: optimization, estimation and control*.
Routledge, 2018.

[32] D. E. Kirk, *Optimal control theory: an introduction*.
Courier Corporation, 2012.

[33] H. Kwakernaak and R. Sivan, *Linear optimal control systems*, vol. 1.
Wiley-Interscience New York, 1972.

[34] D. P. Bertsekas, D. P. Bertsekas, D. P. Bertsekas, and D. P. Bertsekas, *Dynamic programming and optimal control*.
No. 3, Athena Scientific Belmont, MA, 2005.

[35] M. Athans and P. L. Falb, *Optimal control: an introduction to the theory and its applications*.
Courier Corporation, 2013.

[36] B. D. Anderson and J. B. Moore, *Optimal control: linear quadratic methods*.
Courier Corporation, 2007.

[37] D. S. Naidu, *Optimal control systems*.
CRC press, 2002.

[38] D. Liberzon, *Calculus of variations and optimal control theory*.
Princeton University Press, 2011.

[39] R. B. Vinter, *Optimal control*.
Springer, 2010.

[40] H. P. Geering, *Optimal control with engineering applications*.
Springer, 2007.

[41] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, "Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers," *IEEE Control Systems*, vol. 32, no. 6, pp. 76–105, 2012.

[42] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

[43] E. F. Camacho and C. B. Alba, *Model predictive control*. Springer Science & Business Media, 2013.

[44] J. B. Rawlings and D. Q. Mayne, "Model predictive control: Theory and design," 2009.

[45] L. Del Re, F. Allgöwer, L. Glielmo, C. Guardiola, and I. Kolmanovsky, *Automotive model predictive control: models, methods and applications*, vol. 402. Springer, 2010.

[46] H. Waschl, I. Kolmanovsky, M. Steinbuch, and L. Del Re, *Optimization and optimal control in automotive systems*. Springer, 2014.

[47] A. Reig, *Optimal Control for Automotive Powertrain Applications*. PhD thesis, Universitat Politècnica de València (UPV), 2017.

[48] A. Reig, B. Pla, C. Guardiola, and J. M. Lujan, "Analytical optimal solution to the energy management problem in series hybrid electric vehicles," *IEEE Transactions on Vehicular Technology*, 2018.

[49] C. Guardiola, B. Pla, S. Onori, and G. Rizzoni, "Insight into the HEV/PHEV optimal control solution based on a new tuning method," *Control Engineering Practice*, vol. 29, pp. 247–256, 2014.

[50] J. M. Luján, C. Guardiola, B. Pla, and A. Reig, "Cost of ownership-efficient hybrid electric vehicle powertrain sizing for multi-scenario driving cycles,"

Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, vol. 230, no. 3, pp. 382–394, 2016.

[51] C. Guardiola, B. Pla, P. Bares, and H. Waschl, "Adaptive calibration for reduced fuel consumption and emissions," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 230, no. 14, pp. 2002–2014, 2016.

[52] C. Guardiola, H. Climent, B. Pla, and A. Reig, "Optimal control as a method for Diesel engine efficiency assessment including pressure and NOx constraints," *Applied Thermal Engineering*, vol. 117, pp. 452–461, 2017.

[53] J. M. Luján, C. Guardiola, B. Pla, and A. Reig, "Optimal control of a turbocharged direct injection diesel engine by direct method optimization," *International Journal of Engine Research*, p. 1468087418772231, 2018.

[54] K. Zhou, J. C. Doyle, K. Glover, *et al.*, *Robust and optimal control*, vol. 40.
Prentice Hall New Jersey, 1996.

[55] M. Green and D. J. Limebeer, *Linear robust control*.
Courier Corporation, 2012.

[56] B. A. Francis and P. P. Khargonekar, *Robust control theory*, vol. 66.
Springer Science & Business Media, 2012.

[57] I. R. Petersen, V. A. Ugrinovskii, and A. V. Savkin, *Robust Control Design Using H-infinity Methods*.
Springer Science & Business Media, 2012.

[58] D. McFarlane and K. Glover, "Robust controller design using normalized coprime factor plant descriptions (lecture notes in control and information sciences)," 1990.

[59] C. Edwards and S. Spurgeon, *Sliding mode control: theory and applications*.
CRC Press, 1998.

[60] V. Utkin, J. Guldner, and J. Shi, *Sliding mode control in electro-mechanical systems*. CRC Press, 2009.

[61] D. A. White and D. A. Sofge, *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptative Approaches*.
Van Nostrand Reinhold Company, 1992.

[62] F. L. Lewis, J. Campos, and R. Selmic, *Neuro-fuzzy control of industrial systems with actuator nonlinearities*, vol. 24.
Siam, 2002.

[63] W. T. Miller, P. J. Werbos, and R. S. Sutton, *Neural networks for control*.
MIT Press, 1995.

[64] H.-J. Zimmermann, "Fuzzy control," in *Fuzzy Set Theory and Its Applications*, pp. 203–240, Springer, 1996.

[65] B. Egardt, *Stability of adaptive controllers*, vol. 20.
Springer, 1979.

[66] E. Lavretsky and K. A. Wise, "Robust adaptive control," in *Robust and adaptive control*, pp. 317–353, Springer, 2013.

[67] B. Peterson and K. Narendra, "Bounded error adaptive control," *IEEE Transactions on Automatic Control*, vol. 27, no. 6, pp. 1161–1168, 1982.

[68] P. A. Ioannou and P. V. Kokotović, *Adaptive systems with reduced models*, vol. 68.
Springer-Verlag New York, 1983.

[69] K. Narendra and A. Annaswamy, "A new adaptive law for robust adaptation without persistent excitation," *IEEE Transactions on Automatic Control*, vol. 32, no. 2, pp. 134–145, 1987.

[70] J. B. Pomet and L. Praly, "Adaptive nonlinear regulation: Estimation from the Lyapunov equation," *IEEE Transactions on Automatic Control*, vol. 37, no. 6, pp. 729–740, 1992.

[71] B. Yao and M. Tomizuka, "Adaptive robust control of SISO nonlinear systems in a semi-strict feedback form," *Automatica*, vol. 33, no. 5, pp. 893–900, 1997.

[72] B. Yao, F. Bu, J. Reedy, and G.-C. Chiu, "Adaptive robust motion control of single-rod hydraulic actuators: theory and experiments," *IEEE/ASME Transactions on Mechatronics*, vol. 5, no. 1, pp. 79–91, 2000.

[73] B. Yao and M. Tomizuka, "Adaptive robust control of mimo nonlinear systems in semi-strict feedback forms," *Automatica*, vol. 37, no. 9, pp. 1305–1321, 2001.

[74] G. Herrmann, J. Na, and M. N. Mahyuddin, "Novel robust adaptive algorithms for estimation and control: Theory and practical examples," in *Control of Complex Systems*, pp. 661–709, Elsevier, 2016.

[75] J. Na, M. N. Mahyuddin, G. Herrmann, X. Ren, and P. Barber, "Robust adaptive finite-time parameter estimation and control for robotic systems," *International Journal of Robust and Nonlinear Control*, vol. 25, no. 16, pp. 3045–3071, 2015.

[76] J. Na, G. Herrmann, X. Ren, M. N. Mahyuddin, and P. Barber, "Robust adaptive finite-time parameter estimation and control of nonlinear systems," in *Intelligent Control (ISIC), 2011 IEEE International Symposium on*, pp. 1014–1019, IEEE, 2011.

[77] S. Boyd and L. Vandenberghe, *Convex optimization*.
Cambridge university press, 2004.

[78] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*, vol. 15.
Siam, 1994.

[79] E. Alpaydin, *Introduction to machine learning*.
MIT Press, 2009.

[80] C. Szepesvári, "Algorithms for reinforcement learning," *Synthesis lectures on artificial intelligence and machine learning*, vol. 4, no. 1, pp. 1–103, 2010.

[81] T. Hastie, R. Tibshirani, and J. Friedman, "Unsupervised learning," in *The elements of statistical learning*, pp. 485–585, Springer, 2009.

[82] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, *et al.*, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[83] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[84] P. Werbos, "Approximate dynamic programming for realtime control and neural modelling," *Handbook of intelligent control: neural, fuzzy and adaptive approaches*, pp. 493–525, 1992.

[85] R. S. Sutton, A. G. Barto, and R. J. Williams, "Reinforcement learning is direct adaptive optimal control," *IEEE Control Systems*, vol. 12, no. 2, pp. 19–22, 1992.

[86] C. J. C. H. Watkins, *Learning from delayed rewards*.
PhD thesis, King's College, Cambridge, 1989.

[87] R. Bellman, *Dynamic programming*.
Courier Corporation, 2013.

[88] D. P. Bertsekas and J. N. Tsitsiklis, "Neuro-dynamic programming: an overview," in *Proceedings of the 34th IEEE Conference on Decision and Control*, vol. 1, pp. 560–564, IEEE Publ. Piscataway, NJ, 1995.

[89] S. M. Ross, *Introduction to stochastic dynamic programming*.
Academic press, 2014.

[90] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*.
John Wiley & Sons, 2014.

[91] M. Minsky, "Steps toward artificial intelligence," *Proceedings of the IRE*, vol. 49, no. 1, pp. 8–30, 1961.

[92] P. Werbos, "Advanced forecasting methods for global crisis warning and models of intelligence," *General System Yearbook*, pp. 25–38, 1977.

[93] P. J. Werbos, "Applications of advances in nonlinear sensitivity analysis," in *System Modeling and Optimization*, pp. 762–770, Springer, 1982.

[94] P. J. Werbos, "Building and understanding adaptive systems: A statistical/numerical approach to factory automation and brain research," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 17, no. 1, pp. 7–20, 1987.

[95] P. J. Werbos, "Generalization of backpropagation with application to a recurrent gas market model," *Neural Networks*, vol. 1, no. 4, pp. 339–356, 1988.

[96] R. Y. Rubinstein and D. P. Kroese, *Simulation and the Monte Carlo method*, vol. 10.
John Wiley & Sons, 2016.

[97] D. P. Kroese, T. Taimre, and Z. I. Botev, *Handbook of monte carlo methods*, vol. 706.
John Wiley & Sons, 2013.

[98] J. Hammersley, *Monte carlo methods*.
Springer Science & Business Media, 2013.

[99] W. R. Gilks, S. Richardson, and D. Spiegelhalter, *Markov chain Monte Carlo in practice*.
CRC Press, 1995.

[100] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, 1959.

[101] A. H. Klopf, "Brain function and adaptive systems: a heterostatic theory," tech. rep., Air Force Cambridge Research Labs HANSCOM AFB MA, 1972.

[102] L. B. Booker, "Intelligent behavior as an adaptation to the task environment," 1982.

[103] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*.
MIT Press, 1992.

[104] H. John, "Holland. escaping brittleness: The possibilities of general-purpose learning algorithms applied to parallel rule-based systems," *Machine Learning, an Artificial Intelligence Approach*, vol. 2, pp. 593–623, 1986.

[105] G. A. Rummery and M. Niranjan, *On-line Q-learning using connectionist systems*, vol. 37.
University of Cambridge, Department of Engineering Cambridge, England, 1994.

[106] R. S. Sutton, "Generalization in reinforcement learning: Successful examples using sparse coarse coding," in *Advances in Neural Information Processing Systems*, pp. 1038–1044, 1996.

[107] H. P. van Hasselt, *Insights in reinforcement learning*.
WorldPress, 2011.

[108] C. J. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[109] U. Kiencke and L. Nielsen, *Automotive control systems: for engine, driveline, and vehicle*.
Springer Science & Business Media, 2005.

[110] A. G. Ulsoy, H. Peng, and M. Çakmakci, *Automotive control systems*.
Cambridge University Press, 2012.

[111] R. Rajamani, *Vehicle dynamics and control*.
Springer Science & Business Media, 2011.

[112] R. Bosch, *Bosch Automotive Handbook, 10th Edition BOSCH10*.
SAE, 2018.

[113] N. Mayersohn, "The internal combustion engine is not dead yet," The New York Times, 2017.

[114] J. A. Cook, J. Sun, J. H. Buckland, I. V. Kolmanovsky, H. Peng, and J. W. Grizzle, "Automotive powertrain controla survey," *Asian Journal of Control*, vol. 8, no. 3, pp. 237–260, 2006.

[115] J. Sun, I. Kolmanovsky, J. A. Cook, and J. H. Buckland, "Modeling and control of automotive powertrain systems: A tutorial," in *American Control Conference, 2005. Proceedings of the 2005*, pp. 3271–3283, IEEE, 2005.

[116] U. Kiencke, "A view of automotive control systems," *IEEE Control Systems Magazine*, vol. 8, no. 4, pp. 11–19, 1988.

[117] L. Guzzella and C. Onder, *Introduction to modeling and control of internal combustion engine systems*.
Springer Science & Business Media, 2009.

[118] A. A. Stotsky, *Automotive engines: control, estimation, statistical detection*.
Springer Science & Business Media, 2009.

[119] T. Lozano-Perez, *Autonomous robot vehicles*.
Springer Science & Business Media, 2012.

[120] Z. Qu, *Cooperative control of dynamical systems: applications to autonomous vehicles*.
Springer Science & Business Media, 2009.

[121] R. Stone, *Introduction to internal combustion engines*, vol. 3.
Springer, 1999.

[122] J. D. Powell, "A review of IC engine models for control system design," *IFAC Proceedings Volumes*, vol. 20, no. 5, pp. 235–240, 1987.

[123] E. Hendricks and S. C. Sorenson, "SI engine controls and mean value engine modelling," tech. rep., SAE Technical paper 910258, 1991.

[124] E. Hendricks and S. Sorenson, "Mean value SI engine model for control studies," in *1990 American Control Conference*, pp. 1882–1887, IEEE, 1990.

[125] E. Hendricks, A. Chevalier, M. Jensen, S. C. Sorenson, D. Trumpy, and J. Asik, "Modelling of the intake manifold filling dynamics," tech. rep., SAE Technical Paper 960037, 1996.

[126] E. Hendricks and J. B. Luther, "Model and observer based control of internal combustion engines," in *Proceedings of International Workshop on Modeling, Emissions and Control in Automotive Engines (MECA01)*, Citeseer, 2001.

[127] J. Cassidy, M. Athans, and W.-H. Lee, "On the design of electronic automotive engine controls using linear quadratic control theory," *IEEE Transactions on Automatic Control*, vol. 25, no. 5, pp. 901–912, 1980.

[128] J. A. Cook and B. K. Powell, "Modeling of an internal combustion engine for control analysis," *IEEE Control Systems Magazine*, vol. 8, no. 4, pp. 20–26, 1988.

[129] Y. He and C. Rutland, "Application of artificial neural networks in engine modelling," *International Journal of Engine Research*, vol. 5, no. 4, pp. 281–296, 2004.

[130] H. M. Ismail, H. K. Ng, C. W. Queck, and S. Gan, "Artificial neural networks modelling of engine-out responses for a light-duty diesel engine fuelled with biodiesel blends," *Applied Energy*, vol. 92, pp. 769–777, 2012.

[131] K. Nikzadfar and A. H. Shamekhi, "An extended mean value model (emvm) for control-oriented modeling of diesel engines transient performance and emissions," *Fuel*, vol. 154, pp. 275–292, 2015.

[132] J. Deng, R. Stobart, B. Maass, *et al.*, "The applications of artificial neural networks to engines," *Artificial Neural Networks–Industrial and Control Engineering Applications*, pp. 309–332, 2011.

[133] B. Maass, R. Stobart, and J. Deng, "Prediction of NOx emissions of a heavy duty diesel engine with a nlarx model," tech. rep., SAE Technical Paper 2009-01-2796, 2009.

[134] V. Ćirović, D. Aleksendrić, and D. Mladenović, "Braking torque control using recurrent neural networks," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 226, no. 6, pp. 754–766, 2012.

[135] B. Ashok, S. D. Ashok, and C. R. Kumar, "A review on control system architecture of a SI engine management system," *Annual Reviews in Control*, vol. 41, pp. 94–118, 2016.

[136] A. S. Chen, J. Na, G. Herrmann, R. Burke, and C. Brace, "Adaptive air-fuel ratio control for spark ignition engines with time-varying parameter estimation," in *Modelling, Identification and Control (ICMIC), 2017 9th International Conference on*, pp. 1074–1079, IEEE, 2017.

[137] J. Na, G. Herrmann, C. Rames, R. Burke, and C. Brace, "Air-fuel-ratio control of engine system with unknown input observer," in *Control (CONTROL), 2016 UKACC 11th International Conference on*, pp. 1–6, IEEE, 2016.

[138] J. Yang, T. Shen, and X. Jiao, "Model-based stochastic optimal air–fuel ratio control with residual gas fraction of spark ignition engines," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 3, pp. 896–910, 2014.

[139] R. A. Zope, J. Mohammadpour, K. M. Grigoriadis, and M. Franchek, "Air-fuel ratio control of spark ignition engines with TWC using LPV techniques," in *ASME 2009 Dynamic Systems and Control Conference*, pp. 897–903, American Society of Mechanical Engineers, 2009.

[140] S. Wang and D. Yu, "A new development of internal combustion engine air-fuel ratio control with second-order sliding mode," *Journal of Dynamic Systems, Measurement, and Control*, vol. 129, no. 6, pp. 757–766, 2007.

[141] Y. Yildiz, A. M. Annaswamy, D. Yanakiev, and I. Kolmanovsky, "Spark ignition engine fuel-to-air ratio control: An adaptive control approach," *Control Engineering Practice*, vol. 18, no. 12, pp. 1369–1378, 2010.

[142] S. B. Choi and J. K. Hedrick, "An observer-based controller design method for improving air/fuel characteristics of spark ignition engines," *IEEE Transactions on Control Systems Technology*, vol. 6, no. 3, pp. 325–334, 1998.

[143] A. S. Chen and G. Herrmann, "Adaptive optimal control via continuous-time Q-learning for unknown nonlinear affine systems," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 1007–1012, IEEE, 2019.

[144] S. G. Khan, G. Herrmann, F. L. Lewis, T. Pipe, and C. Melhuish, "Reinforcement learning and optimal adaptive control: An overview and implementation examples," *Annual Reviews in Control*, vol. 36, no. 1, pp. 42–59, 2012.

[145] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 6, pp. 2042–2062, 2018.

[146] F. Y. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: An introduction," *IEEE Computational Intelligence Magazine*, vol. 4, no. 2, 2009.

[147] D. L. Kleinman, "On an iterative technique for riccati equation computations," *IEEE Trans. AC*, vol. 13, no. 1, pp. 114–115, 1968.

[148] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. L. Lewis, "Adaptive optimal control for continuous-time linear systems based on policy iteration," *Automatica*, vol. 45, no. 2, pp. 477–484, 2009.

[149] D. Vrabie and F. Lewis, "Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems," *Neural Networks*, vol. 22, no. 3, pp. 237–246, 2009.

[150] D. Vrabie, K. G. Vamvoudakis, and F. L. Lewis, *Optimal adaptive control and differential games by reinforcement learning principles*, vol. 2.
IET, 2013.

[151] K. G. Vamvoudakis and F. L. Lewis, "Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, 2010.

[152] J. Na and G. Herrmann, "Online adaptive approximate optimal tracking control with simplified dual approximation structure for continuous-time unknown nonlinear systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 1, no. 4, pp. 412–422, 2014.

[153] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control," *Automatica*, vol. 43, no. 3, pp. 473–481, 2007.

[154] B. Kiumarsi, F. L. Lewis, H. Modares, A. Karimpour, and M.-B. Naghibi-Sistani, "Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics," *Automatica*, vol. 50, no. 4, pp. 1167–1175, 2014.

[155] F. L. Lewis and K. G. Vamvoudakis, "Reinforcement learning for partially observable dynamic processes: Adaptive dynamic programming using measured output data," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 1, pp. 14–25, 2011.

[156] L. C. Baird, "Reinforcement learning in continuous time: Advantage updating," in *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, vol. 4, pp. 2448–2453, IEEE, 1994.

[157] P. Mehta and S. Meyn, "Q-learning and pontryagin's minimum principle," in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pp. 3598–3605, IEEE, 2009.

[158] J. Y. Lee, J. B. Park, and Y. H. Choi, "Integral Q-learning and explorized policy iteration for adaptive optimal control of continuous-time linear systems," *Automatica*, vol. 48, no. 11, pp. 2850–2859, 2012.

[159] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT Press, 2018.

[160] Y. Jiang and Z.-P. Jiang, "Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics," *Automatica*, vol. 48, no. 10, pp. 2699–2704, 2012.

[161] M. Palanisamy, H. Modares, F. L. Lewis, and M. Aurangzeb, "Continuous-time Q-learning for infinite-horizon discounted cost linear quadratic regulator problems," *IEEE Transactions on Cybernetics*, vol. 45, no. 2, pp. 165–176, 2015.

[162] K. G. Vamvoudakis, "Q-learning for continuous-time linear systems: A model-free infinite horizon optimal control approach," *Systems & Control Letters*, vol. 100, pp. 14–20, 2017.

[163] J. Y. Lee, J. B. Park, and Y. H. Choi, "Integral reinforcement learning for continuous-time input-affine nonlinear systems with simultaneous invariant explorations," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 5, pp. 916–932, 2015.

[164] M. Abu-Khalaf and F. L. Lewis, "Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach," *Automatica*, vol. 41, no. 5, pp. 779–791, 2005.

[165] H. K. Khalil and J. W. Grizzle, *Nonlinear systems*, vol. 3.
Prentice hall Upper Saddle River, NJ, 2002.

[166] V. I. Utkin, *Sliding modes in control and optimization*.
Springer Science & Business Media, 2013.

[167] V. Nevistić and J. A. Primbs, *Constrained nonlinear optimal control: a converse HJB approach*.
Tech. rep. CIT-CDS 96-021. California Institute of Technology, 1996.

[168] K. J. Åström and R. M. Murray, *Feedback systems*.
Princeton University Press, 2010.

[169] D. G. Luenberger, "Observing the state of a linear system," *IEEE Transactions on Military Electronics*, vol. 8, no. 2, pp. 74–80, 1964.

[170] D. Luenberger, "An introduction to observers," *IEEE Transactions on Automatic Control*, vol. 16, no. 6, pp. 596–602, 1971.

[171] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, 1960.

[172] G. Besançon, *Nonlinear observers and applications*, vol. 363.
Springer, 2007.

[173] P. Bernard, "Observer design for nonlinear systems," 2019.

[174] J. Primbs, "Survey of nonlinear observer design techniques," *Penn State Notes*, vol. 1, no. 1, pp. 1–18, 1996.

[175] E. Misawa and J. Hedrick, "Nonlinear observers: A state-of-the-art survey," 1989.

[176] H. K. Khalil and L. Praly, "High-gain observers in nonlinear feedback control," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 6, pp. 993–1015, 2014.

[177] G. Besançon, "Remarks on nonlinear adaptive observer design," *Systems & control letters*, vol. 41, no. 4, pp. 271–280, 2000.

[178] J. Na, G. Herrmann, X. Ren, and P. Barber, "Adaptive discrete neural observer design for nonlinear systems with unknown time-delay," *International Journal of Robust and Nonlinear Control*, vol. 21, no. 6, pp. 625–647, 2011.

[179] S. K. Spurgeon, "Sliding mode observers: a survey," *International Journal of Systems Science*, vol. 39, no. 8, pp. 751–764, 2008.

[180] C. Possieri and M. Sassano, "Deterministic optimality of the steady-state behavior of the Kalman-Bucy filter," *IEEE Control Systems Letters*, vol. 3, no. 4, pp. 793–798, 2019.

[181] S. Bonnabel and J.-J. Slotine, "A contraction theory-based analysis of the stability of the deterministic extended Kalman filter," *IEEE Transactions on Automatic Control*, vol. 60, no. 2, pp. 565–569, 2014.

[182] J. Na, G. Herrmann, and K. G. Vamvoudakis, "Adaptive optimal observer design via approximate dynamic programming," in *2017 American Control Conference (ACC)*, pp. 3288–3293, IEEE, 2017.

[183] V. Durbha and S. Balakrishnan, "New nonlinear observer design with application to electrostatic micro-actuators," in *ASME International Mechanical Engineering Congress and Exposition*, vol. 42169, pp. 101–107, 2005.

[184] D. E. Kirk, *Optimal control theory: an introduction*.
Courier Corporation, 2004.

[185] P. A. Ioannou and J. Sun, *Robust adaptive control*.
Courier Corporation, 2012.

[186] R. W. Beard, G. N. Saridis, and J. T. Wen, "Galerkin approximations of the generalized Hamilton-Jacobi-Bellman equation," *Automatica*, vol. 33, no. 12, pp. 2159–2177, 1997.

[187] E. Barbieri and R. Alba-Flores, "On the infinite-horizon LQ tracker," *Systems & Control Letters*, vol. 40, no. 2, pp. 77–82, 2000.

[188] A. J. Van Der Schaft, "L 2-gain analysis of nonlinear systems and nonlinear state feedback H_∞ control," *IEEE Transactions on Automatic Control*, vol. 37, no. 6, pp. 770–784, 1992.

[189] H. Modares and F. L. Lewis, "Linear quadratic tracking control of partially-unknown continuous-time systems using reinforcement learning," *IEEE Transactions on Automatic control*, vol. 59, no. 11, pp. 3051–3056, 2014.

[190] H. Modares and F. L. Lewis, "Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning," *Automatica*, vol. 50, no. 7, pp. 1780–1792, 2014.

[191] G. N. Saridis and C.-S. G. Lee, "An approximation theory of optimal control for trainable manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 3, pp. 152–159, 1979.

[192] P. G. Ciarlet, *Linear and nonlinear functional analysis with applications*, vol. 130.
Siam, 2013.

[193] R. A. Adams and J. J. Fournier, *Sobolev spaces*.
Elsevier, 2003.

[194] K. Hornik, M. Stinchcombe, and H. White, "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks," *Neural networks*, vol. 3, no. 5, pp. 551–560, 1990.

[195] K. Doya, "Reinforcement learning in continuous time and space," *Neural Computation*, vol. 12, no. 1, pp. 219–245, 2000.

[196] J.-P. Richard, "Time-delay systems: an overview of some recent advances and open problems," *Automatica*, vol. 39, no. 10, pp. 1667–1694, 2003.

[197] D. Astolfi and L. Marconi, "A high-gain nonlinear observer with limited gain power," *IEEE Transactions on Automatic Control*, vol. 60, no. 11, pp. 3059–3064, 2015.

[198] H. Dark, "The Wankel engine: Introduction & guide," 1974.

[199] I. Arsie, C. Pianese, and G. Rizzo, "Identification of manifold two-phase fuel flow model in a spark ignition engine with Kalman filter and least square methods," in *7th IEEE Mediterranean Conference on" Control & Systems*, pp. 28–30, 1999.

[200] M. Peden, M. Turner, J. W. Turner, and N. Bailey, "Comparison of 1-D modelling approaches for Wankel engine performance simulation and initial study of the direct injection limitations," tech. rep., SAE Technical Paper 018-01-1452, 2018.

[201] L. Tartakovsky, V. Baibikov, M. Gutman, M. Veinblat, and J. Reif, "Simulation of Wankel engine performance using commercial software for piston engines," tech. rep., SAE Technical Paper 2012-32-0098, 2012.

[202] G. Vorraro, M. Turner, and J. W. Turner, "Testing of a modern Wankel rotary engine-part i: Experimental plan, development of the software tools and measurement systems," tech. rep., SAE Technical Paper 2019-01-0075, 2019.

[203] R. Sierens, R. Baert, D. Winterbone, and P. Baruah, "A comprehensive study of Wankel engine performance," tech. rep., SAE Technical Paper 830332, 1983.

[204] G. A. Danieli, J. C. Keck, and J. B. Heywood, "Experimental and theoretical analysis of Wankel engine performance," tech. rep., SAE Technical Paper 780416, 1978.

[205] A. R. Chis, "How Wankel's rotary engine works," 2010.

[206] I. Arsie, F. De Franceschi, C. Pianese, and G. Rizzo, "Odecs-a computer code for the optimal design of SI engine control strategies," tech. rep., SAE Technical Paper 960359, 1996.

[207] A. S. Chen, G. Herrmann, J. Na, M. Turner, G. Vorraro, and C. Brace, "Nonlinear observer-based air-fuel ratio control for port fuel injected Wankel engines," in *2018 UKACC 12th International Conference on Control (CONTROL)*, pp. 224–229, IEEE, 2018.

[208] S. Shanmuganathan, "Artificial neural network modelling: An introduction," in *Artificial neural network modelling*, pp. 1–14, Springer, 2016.

[209] T. P. Vogl, J. Mangis, A. Rigler, W. Zink, and D. Alkon, "Accelerating the convergence of the back-propagation method," *Biological Cybernetics*, vol. 59, no. 4, pp. 257–263, 1988.

[210] K. S. Narendra and K. Parthasarathy, "Learning automata approach to hierarchical multiobjective analysis," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, no. 1, pp. 263–272, 1991.

[211] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the marquardt algorithm," *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, 1994.

[212] M. Thornhill, S. Thompson, and H. Sindano, "A comparison of idle speed control schemes," *Control Engineering Practice*, vol. 8, no. 5, pp. 519–530, 2000.

[213] X.-D. Sun, P. G. Scotson, and G. Balfour, "A further application of loop shaping H-infinity control to diesel engine control-driven-idle speed control," tech. rep., SAE Technical Paper 2002-01-0197, 2002.

[214] Y. Yildiz, A. Annaswamy, D. Yanakiev, and I. Kolmanovsky, "Adaptive idle speed control for internal combustion engines," in *2007 American Control Conference*, pp. 3700–3705, IEEE, 2007.

[215] X. Li and S. Yurkovich, "Sliding mode control of delayed systems with application to engine idle speed control," *IEEE Transactions on Control Systems Technology*, vol. 9, no. 6, pp. 802–810, 2001.

[216] S. Di Cairano, D. Yanakiev, A. Bemporad, I. V. Kolmanovsky, and D. Hrovat, "Model predictive idle speed control: Design, analysis, and experimental evaluation," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 1, pp. 84–97, 2011.

[217] F. L. Lewis and K. G. Vamvoudakis, "Reinforcement learning for partially observable dynamic processes: Adaptive dynamic programming using measured output data," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 1, pp. 14–25, 2010.

[218] W. Gao, Y. Jiang, Z.-P. Jiang, and T. Chai, "Output-feedback adaptive optimal control of interconnected systems based on robust adaptive dynamic programming," *Automatica*, vol. 72, pp. 37–45, 2016.

[219] I. Arsie, C. Pianese, G. Rizzo, and V. Cioffi, "An adaptive estimator of fuel film dynamics in the intake port of a spark ignition engine," *Control Engineering Practice*, vol. 11, no. 3, pp. 303–309, 2003.

[220] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *Signal processing, sensor fusion, and target recognition VI*, vol. 3068, pp. 182–194, International Society for Optics and Photonics, 1997.

[221] L. Ljung, "Asymptotic behavior of the extended Kalman filter as a parameter estimator for linear systems," *IEEE Transactions on Automatic Control*, vol. 24, no. 1, pp. 36–50, 1979.

[222] E.-W. Bai, H. Ishii, and R. Tempo, "A Markov chain Monte Carlo approach to nonlinear parametric system identification," *IEEE Transactions on Automatic Control*, vol. 60, no. 9, pp. 2542–2546, 2014.

[223] F. Dabbene, M. Sznaier, and R. Tempo, "Probabilistic optimal estimation with uniformly distributed noise," *IEEE Transactions on Automatic Control*, vol. 59, no. 8, pp. 2113–2127, 2014.

[224] F. Tjarnstrom and A. Garulli, "A mixed probabilistic/bounded-error approach to parameter estimation in the presence of amplitude bounded white noise," in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, vol. 3, pp. 3422–3427, IEEE, 2002.

[225] M. Peden, M. Turner, J. W. G. Turner, and N. Bailey, "Comparison of 1-D modelling approaches for Wankel engine performance simulation and initial study of the direct injection limitations," tech. rep., SAE Technical Paper 2018-01-1452, 2018.

[226] C. C. Lin, H. Peng, J. W. Grizzle, and J.-M. Kang, "Power management strategy for a parallel hybrid electric truck," *IEEE Transactions on Control Systems Technology*, vol. 11, no. 6, pp. 839–849, 2003.

[227] J. Liu and H. Peng, "Control optimization for a power-split hybrid vehicle," in *2006 American Control Conference*, pp. 6–pp, IEEE, 2006.

[228] N. Kim, S. Cha, and H. Peng, "Optimal control of hybrid electric vehicles based on Pontryagin's minimum principle," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 5, pp. 1279–1287, 2010.