Optimal Control of a Two-Wheeled Self-Balancing Robot by Reinforcement Learning

---

A Thesis

Presented to

the Faculty of the School of Engineering and Applied Science

University of Virginia

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Electrical Engineering

---

by

Linyuan Guo

May 2020

APPROVAL SHEET

This thesis is submitted in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering

Linyuan Guo, Author

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

Professor Zongli Lin, Thesis Advisor

Professor Gang Tao

Professor Joanne Bechta Dugan

Accepted for the School of Engineering and Applied Science:

Dean, School of Engineering and Applied Science

May, 2020

Acknowledgments

Firstly, I would like to express my deepest and sincere gratitude to my advisor Professor Zongli Lin. I would never have finished this work without his patient guidance and persistent encouragement. His professional knowledge and experience in control theory and application have helped me to overcome many difficulties and problems I met in completing this research. His dynamism, sincerity, vision and concentration in academia have deeply inspired me. There is no doubt that it is a great honor and privilege to work and study under his guidance.

Secondly, I would like to thank my colleagues in my research group, including Syed Ali Asad Rizvi, Yusheng Wei, Tingyang Meng, Haoyi Ma and Qilin Song, who have often given me valuable suggestions during the process of my research.

Finally, I would like to express my profound gratitude to my parents, who have provided me continuous encouragement and consistent support through my years of study and during my research. This thesis would not have been accomplished without their help.

Abstract

This thesis concerns optimal control of the linear motion, tilt motion, and yaw motion of a two-wheeled self-balancing robot. Traditional optimal control methods for the two-wheeled self-balancing robot usually require a precise model of the system, and other control methods exist that achieve stabilization in the face of parameter uncertainties. In practical applications, it is often desirable to realize optimal control in the absence of the precise knowledge of the system parameters. This thesis proposes to use a new feedback-based reinforcement learning method to solve the linear quadratic regulation (LQR) control problem for the two-wheeled self-balancing robot. The proposed control scheme is completely online and does not require any knowledge of the system parameters. The proposed input decoupling mechanism and pre-feedback law overcome the commonly encountered computational difficulties in implementing the learning algorithms, which the former shortens the learning transient phase and the latter improves the system performance. Both state feedback optimal control and output feedback optimal control are presented. Numerical simulation shows that the proposed optimal control scheme is capable of stabilizing the system and converging to the LQR solution obtained through solving the algebraic Riccati equation.

Contents

List of Figures

Chapter 1

Introduction

This chapter gives a brief introduction about the two-wheeled self-balancing robot and the reinforcement learning control, followed by the thesis outline.

1.1 Two-Wheeled Self-Balancing Robot

The two-wheeled self-balancing robot (TWSBR) is a typical robot that has potential application prospects in many areas, such as transportation and exploration. Design and control of the TWSBR have attracted substantial attention in both academia and industry over the past decades. The TWSBR is an inherently unstable, high-order, multivariable, nonlinear, and strongly coupled system, and represents an underactuated mechanical system. For such an underactuated mechanical system, which has fewer control inputs than the generalized coordinates, it is necessary to indirectly control the underactuated generalized coordinates through dynamic coupling. Underactuation, while resulting in a smaller number of actuators and thus helping to reduce the manufacturing costs and failure rate, poses challenges to control design. Furthermore, unlike simpler systems like the pendulum on a cart system that are restrained to a guided trajectory, the TWSBR moves on its own trajectory while balancing the

pendulum. One of the difficulties of controlling the TWSBR is to simultaneously control its linear motion, tilt motion and yaw motion [1]. In addition, control of the TWSBR with system parameter uncertainties is essential in practical applications.

Several control methods were used to stabilize the TWSBR. In the work of [23], velocity and position control of the TWSBR using partial feedback linearization was proposed. In the work of [24], a well-known pole-placement state feedback controller was designed for the TWSBR. In [27], an adaptive integral backstepping controller with the velocity estimator for the TWSBR was presented to stabilize the system. Other traditional control methods, including PID control, fuzzy control and sliding mode control, were also investigated in previous works. In the works of [1], [2] and [26], the conventional PID or PD controllers with the adaptation and robustness abilities were proposed for the TWSBR. In [3], adaptive fuzzy logic control of the TWSBR with parametric and functional uncertainties was investigated. In the work of [25], fuzzy logic control of the TWSBR was investigated in order to achieve balance and velocity control of the system. In the work of [4], two sliding mode control methods were proposed for the TWSBR with parameter unknown and external disturbance. In [5], nonlinear adaptive sliding mode controllers were proposed for the two-wheeled human transportation vehicle with system parameter uncertainties and variations. Some of these methods are capable of controlling the TWSBR in the absence of the precise knowledge of the system parameters, but do not achieve optimal control. In practical applications, it is often desirable to achieve optimality beyond simple stabilization. The design of the traditional LQR controller based on the solution of the Riccati equation achieves the goal of optimal control [6], [7]. However, this control method requires precise knowledge of the system model. A control scheme that realizes optimal control in the absence of precise knowledge of the system model is desirable.

1.2 Reinforcement Learning Control

Reinforcement learning is a type of machine learning, which is a popular method for solving dynamic optimization problems. Reinforcement learning is motivated by the living organism learning mechanism by which animals reveal the capability of learning, adapting and optimizing their behaviors by interacting with the environments. Reinforcement learning used to solve optimization problems involves an agent that interacts with its environment and modifies its actions or control policies based on some stimuli or reward received in response to its actions. Reinforcement learning indicates a cause-and-effect relationship between actions and reward or punishment, which matches well with the framework of the feedback mechanism in control community [18], [21]. Fig 1.1 illustrates a block diagram of the reinforcement learning mechanism, which has been attracted significant attention in designing optimal feedback controllers. Based on this mechanism, a reinforcement learning-based controller is able to learn the optimal control parameters and stabilize the system without requiring the system model information. In addition to realizing optimal control of the system, the reinforcement learning-based controller also has the adaptation ability by adapting to the changes in system dynamics during the learning process [28].

Q-learning is a type of reinforcement learning algorithm, which is completely online in nature and does not use any prior information of system dynamics [22]. This technique was introduced in the work of [30] and is based on learning the Q-function involving both the states and the control actions. The Q-function is the sum of the single step cost of implementing an arbitrary control action in the current state and the total cost of implementing a specific policy from the next state to all the future states. As a Q-function includes information about control actions in every state, the best control action in each state can be chosen by identifying only the Q-function.

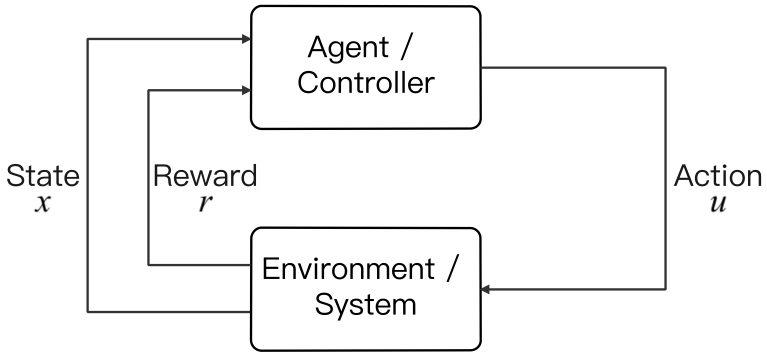


Figure 1.1: Reinforcement learning mechanism

In other words, the purpose of the Q-learning algorithm is to estimate the optimal Q-function. Once the optimal Q-function is learned, the optimal control action can be obtained by minimizing or maximizing the optimal Q-function [29]. The Q-learning Policy Iteration (PI) algorithm was proposed in the work of [9], which requires an initially stabilizing control policy, and the Q-learning Value Iteration (VI) algorithm was presented in the work of [10], which can start with an arbitrary control policy. A review of Q-learning LQR control was given in the work of [11]. However, full-state feedback was needed in these papers.

In many practical cases, an output feedback control scheme is more desirable because it requires fewer sensors and therefore is more cost effective and reliable. The output feedback reinforcement learning control method was proposed in the work of [12], which used the value function approximation (VFA) method to develop PI and VI-based algorithms. In that work, the cost function consists of a discounting factor that helps to overcome the bias issue associated with the excitation signals. However, the resulting optimal controller is different from the optimal controller obtained through solving the Riccati equation, and the use of discounted cost function

may cause the loss of stability inherent from the optimal controller corresponding to the undiscounted cost function. In the works of [13] and [14], the output feedback Q-learning algorithms were proposed without using discounting factor in the cost function. The optimal controllers learned by these algorithms are the same as the one obtained through solving the Riccati equation and the closed-loop stability is guaranteed.

1.3 Thesis Outline

In this thesis, we propose to use the Q-learning method to design optimal controllers for the TWSBR. Both state feedback and output feedback are considered. In order to overcome the commonly encountered numerical difficulties associated with high dimensionality and strong instability of the open-loop system in implementing Q-learning control algorithms, we propose to explore the physical properties of the system and adopt an input decoupling mechanism and a pre-feedback law before applying the Q-learning algorithms.

The remainder of the thesis is organized as follows. Chapter 2 presents the description of the system, Chapter 3 provides the control design algorithms, and Chapter 4 presents simulation results for optimal control of the robot. Some concluding remarks are made in Chapter 5, where some future research topics are also pointed out.

Chapter 2

Description of the System

In this chapter, we describe the TWSBR system using Newtonian mechanics. The TWSBR is composed of a pair of identical wheels, along with their actuators, a chassis, and an inverted pendulum. The chassis sustains the inverted pendulum and the pair of wheels. The wheel actuators generate torques to rotate the wheels with respect to the chassis. The motion control unit controls the wheel actuators so as to move and stabilize the robot [1], [8]. Fig. 2.1 illustrates the system. The robot is able to execute linear motion along the X-axis, rotate around the Z-axis to execute tilt motion, and rotate around the Y-axis to execute yaw motion. The parameters are defined in Table 2.1. The dynamics of the system can be described by the following equations [15], [19]. For the left wheel,

$$m\ddot{x}_l = f_l - H_l \tag{2.1}$$

$$J_\omega \ddot{\theta}_l = C_l - f_l R \tag{2.2}$$

For the right wheel,

$$m\ddot{x}_r = f_r - H_r \tag{2.3}$$

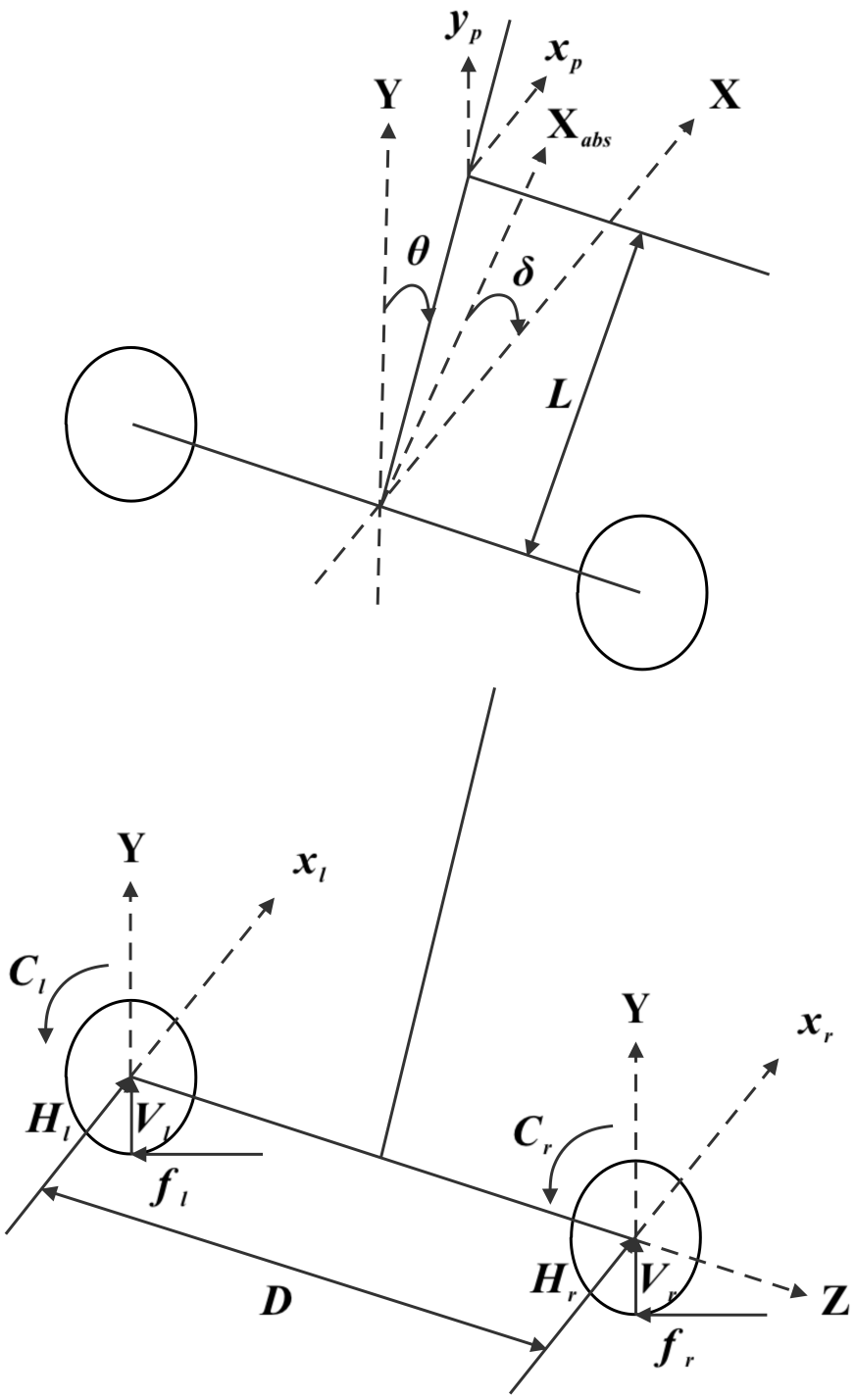


Figure 2.1: A diagram of the TWSBR

Table 2.1: Definitions of parameters of the system

Symbol and Unit	Definition
M [kg]	Mass of the chassis (with the inverted pendulum part)
m [kg]	Mass of each wheel
R [m]	Radius of the wheel
D [m]	Distance between the two wheels
L [m]	Distance between the center of gravity of the robot and the Z-axis
J_δ [kg·m^2]	Moment of inertia of the chassis with respect to the Y-axis
J_p [kg·m^2]	Moment of inertia of the chassis with respect to the Z-axis
J_ω [kg·m^2]	Moment of inertia of the left (or right) wheel with respect to the Z-axis
v [m/s]	Linear speed of the robot
θ [rad]	Tilt angle of the robot
ω [rad/s]	Tilt angular velocity of the robot
δ [rad]	Yaw angle of the robot
$\dot{\delta}$ [rad/s]	Yaw angular velocity of the robot
x_l, x_r [m]	Displacements of the left and right wheels
x_p, y_p [m]	The position of the center of gravity of the robot
H_l, H_r [N]	Interacting forces between the wheels and the chassis on the X-axis
V_l, V_r [N]	Interacting forces between the wheels and the chassis on the Y-axis
f_l, f_r [N]	Frictions between the wheels and the ground surface
C_l, C_r [N·m]	Torques generated from the left and right actuators
θ_l, θ_r [rad]	Rotational angles of the left and right wheels

$$J_\omega \ddot{\theta}_r = C_r - f_r R \tag{2.4}$$

For the chassis,

$$M\ddot{x}_p = H_l + H_r \tag{2.5}$$

$$M\ddot{y}_p = V_l + V_r - Mg \tag{2.6}$$

$$J_p \ddot{\theta} = (V_l + V_r)\, L \sin\theta - (H_l + H_r)\, L \cos\theta - (C_l + C_r) \tag{2.7}$$

$$J_\delta \ddot{\delta} = \frac{D}{2}\,(H_l - H_r) \tag{2.8}$$

The relationships between the rotational angles of the two wheels and their displacements are given by

$$x_l = R\theta_l, \quad x_r = R\theta_r \tag{2.9}$$

Furthermore, the relationship between the yaw angle of the robot and the displacements of the wheels is

$$D\delta = x_l - x_r \tag{2.10}$$

Letting $x = \frac{1}{2}(x_l + x_r)$, we have

$$x_p = x + L\sin\theta, \quad y_p = L\cos\theta \tag{2.11}$$

Combining (2.1) - (2.11), we obtain the following nonlinear equations of the system,

$$\ddot{x}\left(M + 2m + \frac{2J_\omega}{R^2}\right) + ML\left(\ddot{\theta}\cos\theta - \dot{\theta}^2\sin\theta\right) = \frac{1}{R}(C_l + C_r) \tag{2.12}$$

$$\left(\frac{2J_\delta}{D} + \frac{DJ_\omega}{R^2} + Dm\right)\ddot{\delta} = \frac{1}{R}(C_l - C_r) \tag{2.13}$$

$$\begin{aligned} J_p\ddot{\theta} = {} & 2\ddot{x}L\left(m + \frac{J_\omega}{R^2}\right)\cos\theta + MgL\sin\theta - ML^2\ddot{\theta}\sin^2\theta \\ & - ML^2\dot{\theta}^2\sin\theta\cos\theta - \left(1 + \frac{L\cos\theta}{R}\right)(C_l + C_r) \end{aligned} \tag{2.14}$$

Linearizing these nonlinear equations around $\theta = 0$, we obtain the following linear state space model of the system,

$$\begin{bmatrix} \dot{x} \\ \dot{v} \\ \dot{\theta} \\ \dot{\omega} \\ \dot{\delta} \\ \ddot{\delta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{23} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & a_{43} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ v \\ \theta \\ \omega \\ \delta \\ \dot{\delta} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ b_{21} & b_{22} \\ 0 & 0 \\ b_{41} & b_{42} \\ 0 & 0 \\ b_{61} & b_{62} \end{bmatrix} \begin{bmatrix} C_l \\ C_r \end{bmatrix} \tag{2.15}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ v \\ \theta \\ \omega \\ \delta \\ \dot{\delta} \end{bmatrix} \tag{2.16}$$

where $\begin{bmatrix} x & v & \theta & \omega & \delta & \dot{\delta} \end{bmatrix}^T$ is the state vector, $\begin{bmatrix} C_l & C_r \end{bmatrix}^T$ is the input vector, $y = \begin{bmatrix} x & \theta & \delta \end{bmatrix}^T$ is the output vector, and the parameters are defined as,

$$
\begin{aligned}
a_{23} &= \frac{-M^2L^2g}{MJ_p + 2\left(J_p + ML^2\right)\left(m + J_\omega/R^2\right)} \\
a_{43} &= \frac{M^2gL + 2MgL\left(m + J_\omega/R^2\right)}{MJ_p + 2\left(J_p + ML^2\right)\left(m + J_\omega/R^2\right)} \\
b_{21} &= b_{22} = \frac{\left(J_p + ML^2\right)/R + ML}{MJ_p + 2\left(J_p + ML^2\right)\left(m + J_\omega/R^2\right)} \\
b_{41} &= b_{42} = \frac{-\left(R + L\right)M/R - 2\left(m + J_\omega/R^2\right)}{MJ_p + 2\left(J_p + ML^2\right)\left(m + J_\omega/R^2\right)} \\
b_{61} &= -b_{62} = \frac{D/2R}{J_\delta + \frac{D^2}{2R}\left(mR + \frac{J_\omega}{R}\right)}
\end{aligned}
$$

The objective of this thesis is to present a control scheme that is capable of realizing optimal control of the TWSBR when system parameters listed above are unknown.

Chapter 3

Design of Optimal Controllers

In this chapter, we describe optimal controllers that stabilize the linear motion, tilt motion and yaw motion of the TWSBR system in the absence of any knowledge of the values of the system parameters. The control scheme is completely online in nature and utilizes a Q-learning to solve the LQR control problem. We will present both the state feedback optimal control method and the output feedback optimal control method for the system.

3.1 Input Decoupling and Pre-feedback

In [11], [13], [14], the Q-learning algorithms are developed that achieve optimal control of the system in the absence of a model. However, computational issues emerge when these control algorithms are applied to the TWSBR because of the high order and the strong open-loop instability of the TWSBR. We propose to take advantages of the structure of the matrices in the model and the physical characteristics of the system and introduce an input decoupling mechanism to decouple the 6th order system model into two lower order systems and a pre-feedback law that moderates the open-loop instability. These measures prove to mitigate the computational issues

in the learning process and improve the system behavior.

3.1.1 Input decoupling

From the state space model in (2.15), we know that the wheel torques C_l and C_r have influences on the motion in all three directions simultaneously, which means there exists a coupling problem in the system. Motivated by the work of [15], we utilize a decoupling unit that transforms the wheel torques C_l and C_r into the new control inputs C_θ and C_δ. These two new control inputs control the tilt motion and the yaw motion, independently. Such a decoupling mechanism takes the form of

$$\begin{bmatrix} C_l \\ C_r \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{bmatrix} \begin{bmatrix} C_\theta \\ C_\delta \end{bmatrix} \tag{3.1}$$

Combining (2.15) and (3.1), we have

$$\begin{bmatrix} \dot{x} \\ \dot{v} \\ \dot{\theta} \\ \dot{\omega} \\ \dot{\delta} \\ \ddot{\delta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{23} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & a_{43} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ v \\ \theta \\ \omega \\ \delta \\ \dot{\delta} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ b_{21} & 0 \\ 0 & 0 \\ b_{41} & 0 \\ 0 & 0 \\ 0 & b_{61} \end{bmatrix} \begin{bmatrix} C_\theta \\ C_\delta \end{bmatrix} \tag{3.2}$$

More specifically, under the new control inputs C_θ and C_δ, the system is decoupled into two subsystems. Subsystem I governs the linear motion and the tilt motion of the system, while Subsystem II governs the yaw motion. The state space model of the two subsystems are given as follows.

Subsystem I:

$$\begin{bmatrix} \dot{x} \\ \dot{v} \\ \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & a_{23} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & a_{43} & 0 \end{bmatrix} \begin{bmatrix} x \\ v \\ \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 \\ b_{21} \\ 0 \\ b_{41} \end{bmatrix} C_\theta \tag{3.3}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ v \\ \theta \\ \omega \end{bmatrix} \tag{3.4}$$

Subsystem II:

$$\begin{bmatrix} \dot{\delta} \\ \ddot{\delta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta \\ \dot{\delta} \end{bmatrix} + \begin{bmatrix} 0 \\ b_{61} \end{bmatrix} C_\delta \tag{3.5}$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \delta \\ \dot{\delta} \end{bmatrix} \tag{3.6}$$

3.1.2 Pre-feedback

By the physical characteristics of the TWSBR, we know that the instability of the system is mainly caused by the instability of the inverted pendulum part, and the third state θ in (3.3), the tilt angle of the inverted pendulum, has the most influence on controlling the robot. In order to mitigate the instability of Subsystem I and make the control algorithms easier and faster to converge to the optimal solution, we include a pre-feedback with gain $K = \begin{bmatrix} 0 & 0 & k_3 & 0 \end{bmatrix}$ applied to Subsystem I before executing the learning algorithms. This pre-feedback renders Subsystem I from exponentially unstable to polynomially unstable.

3.2 State Feedback Optimal Q-Learning Control

Consider a discrete-time linear time-invariant system,

$$x_{k+1} = Ax_k + Bu_k,\ x_k \in \mathbb{R}^n,\ u_k \in \mathbb{R}^m \tag{3.7}$$

where (A, B) is controllable.

The LQR problem is to determine the feedback control sequence that minimizes the following cost function

$$J = \sum_{i=0}^{\infty} r\left(x_i, u_i\right) = \sum_{i=0}^{\infty} \left(x_i^T Q x_i + u_i^T R u_i\right) \tag{3.8}$$

with the one step utility function $r\left(x_k, u_k\right) = x_k^T Q x_k + u_k^T R u_k$ and the user-defined weighting matrices $Q = Q^T \geq 0$ and $R = R^T > 0$. Assume that $\left(A, \sqrt{Q}\right)$ is detectable.

The optimal control law is given by,

$$u_k^* = K^* x_k = -\left(R + B^T P^* B\right)^{-1} B^T P^* A x_k \tag{3.9}$$

where $P^* = P^{*T} > 0$ is the unique positive definite solution to the algebraic Riccati equation (ARE),

$$A^T P A - P + Q - A^T P B \left(R + B^T P B\right)^{-1} B^T P A = 0 \tag{3.10}$$

Determining the optimal control policy through solving the ARE requires the precise knowledge of the system model and parameters. In what follows we recall the Q-learning-based control algorithms that result in the optimal control policy in the

absence of the knowledge of the system parameters in matrices A and B [9], [11], [14], [17], [20].

The cost function is defined as,

$$V_K(x_k) = \sum_{i=k}^{\infty} r(x_i, u_i) \tag{3.11}$$

which gives the cost of following a control policy $u_k = Kx_k$ starting from state x_k. Under a stabilizing policy, this cost function takes the following quadratic form,

$$V_K(x_k) = x_k^T P x_k,\ P = P^T > 0 \tag{3.12}$$

Motivated by Bellman optimality principle, (3.11) can be expressed as,

$$V_K(x_k) = r(x_k, Kx_k) + V_K(x_{k+1}) \tag{3.13}$$

The Q-function is then defined as,

$$Q_K(x_k, u_k) = r(x_k, u_k) + V_K(x_{k+1}) \tag{3.14}$$

which is the sum of the single step cost of implementing an arbitrary control u_k from state x_k and the total cost of implementing a policy K from x_{k+1} and all future states. This Q-function can be expressed as,

$$\begin{aligned} Q_K(x_k, u_k) &= x_k^T Q x_k + u_k^T R u_k + x_{k+1}^T P x_{k+1} \\ &= x_k^T Q x_k + u_k^T R u_k + (Ax_k + Bu_k)^T P (Ax_k + Bu_k) \\ &= \begin{bmatrix} x_k \\ u_k \end{bmatrix}^T \begin{bmatrix} H_{xx} & H_{xu} \\ H_{ux} & H_{uu} \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} \\ &\triangleq z_k^T H z_k \end{aligned} \tag{3.15}$$

with $z_k = \begin{bmatrix} x_k^T & u_k^T \end{bmatrix}^T$ and the submatrices defined as

$$\begin{aligned} H_{xx} &= Q + A^T P A \in \mathbb{R}^{n \times n} \\ H_{ux} &= B^T P A \in \mathbb{R}^{m \times n} \\ H_{xu} &= A^T P B \in \mathbb{R}^{n \times m} \\ H_{uu} &= R + B^T P B \in \mathbb{R}^{m \times m} \end{aligned} \tag{3.16}$$

By choosing a greedy action, the improved policy $K^{'}$ can be obtained when the cost V_K associated with the control policy K is given, which can be expressed as,

$$\begin{aligned} K^{'} x_k &= \arg\min_u \left(r\left(x_k, u_k\right) + V_K\left(x_{k+1}\right) \right) \\ &= \arg\min_u \left(Q_K\left(x_k, u_k\right) \right) \end{aligned} \tag{3.17}$$

indicating that the improved control policy $K^{'}$ can be obtained by solving $\left(\partial / \partial u_k\right) Q_K = 0$. The cost of the improved control policy $K^{'}$ (the new policy) is better than or equal to the cost of the current control policy K, which can be expressed as $V_{K'} \leq V_K$. After several policy improvements, the cost is able to converge to the optimal cost V^*, while the policy can converge to the optimal control policy K^* [9], [17], [20]. This policy improvement mechanism forms the basis of the iterative improvement algorithms, which will be shown in Algorithms 1, 2, 3 and 4 below. The optimal policy K^* can be obtained when the optimal cost V^* is given. Then, we define the optimal Q-function as,

$$Q^*\left(x_k, u_k\right) = r\left(x_k, u_k\right) + V^*\left(x_{k+1}\right) \tag{3.18}$$

The following optimal control policy K^* can be obtained as,

$$K^* x_k = \arg\min_u \left(Q^*\left(x_k, u_k\right) \right) \tag{3.19}$$

which means the optimal controller can be obtained by minimizing the optimal Q-function Q^* corresponding to P^* and H^* by (3.15). By setting $(\partial/\partial u_k)\, Q^* = 0$, the optimal result of u_k can be obtained as,

$$u_k^* = -\left(H_{uu}^*\right)^{-1} H_{ux}^* x_k \tag{3.20}$$

Substituting (3.16) into (3.20), we arrive at the same result as given by (3.9), which is obtained by solving the ARE.

We would like to obtain the recursive form of the Q-function so that the reinforcement learning techniques can be applied to learn the optimal controller. Motivated by Bellman optimality principle and combing (3.13), (3.14), (3.15), the recursive form of the Q-function can be derived as below:

$$\begin{aligned} Q_K\left(x_k, u_k\right) &= x_k^T Q x_k + u_k^T R u_k + Q_K\left(x_{k+1}, K x_{k+1}\right) && (3.21) \\ z_k^T H z_k &= x_k^T Q x_k + u_k^T R u_k + z_{k+1}^T H z_{k+1} && (3.22) \end{aligned}$$

Equation (3.21) is the LQR Bellman Q-learning equation. In (3.22), u_{k+1} is computed as

$$u_{k+1} = -\left(H_{uu}\right)^{-1} H_{ux} x_{k+1} \tag{3.23}$$

The matrix H in Q-function is related to system dynamics and parameters and is unknown, the reinforcement learning techniques can be applied to learn the H matrix, and the optimal controller can be obtained. To this end, we parameterize the matrix H in (3.15) as,

$$Q_K = \bar{H}^T \bar{z}_k \tag{3.24}$$

where

$$\begin{aligned}\bar{H} &= \text{vec}\,(H) \\ &\triangleq \left[\, h_{11}\ ,\ 2h_{12}\ ,\ \cdots\ ,\ 2h_{1l}\ ,\ h_{22}\ ,\ 2h_{23}\ ,\ \cdots\ ,\ 2h_{2l}\ ,\ \cdots\ ,\ h_{ll} \,\right]^T \in \mathbb{R}^{l(l+1)/2}\end{aligned}$$

with $l = m + n$. The regression vector $\bar{z}_k \in \mathbb{R}^{l(l+1)/2}$ can be expressed as,

$$\begin{aligned}\bar{z}_k &= z_k \otimes z_k \\ \bar{z} &= \left[\, z_1^2\ ,\ z_1 z_2\ ,\ \cdots\ ,\ z_1 z_l\ ,\ z_2^2\ ,\ z_2 z_3\ ,\ \cdots\ ,\ z_2 z_l\ ,\ \cdots\ ,\ z_l^2 \,\right]^T\end{aligned}$$

Then, we can obtain the following Bellman equation,

$$\bar{H}^T \bar{z}_k = x_k^T Q x_k + u_k^T R u_k + \bar{H}^T \bar{z}_{k+1} \tag{3.25}$$

Then, the state feedback Q-learning PI and VI algorithms are presented in the following.

Algorithm 1: State Feedback Q-Learning Policy Iteration (PI)

Initialization. Start with a stabilizing control policy u_k^0 with $H^0 = I$. Then, for the following iterations $j = 1, 2, \cdots$, repeat until the convergence criterion is met,

$$\| \bar{H}^j - \bar{H}^{j-1} \| < \epsilon$$

for the constant scalar ϵ that can be set by users according to the requirement of system optimal accuracy.

Policy Evaluation. Determine the least-squares solution of

$$\left(\bar{H}^j\right)^T (\bar{z}_k - \bar{z}_{k+1}) = x_k^T Q x_k + u_k^T R u_k$$

Policy Update. Determine an improved control policy using

$$u_k^{j+1} = -\left(H_{uu}^j\right)^{-1} H_{ux}^j x_k$$

Algorithm 2: State Feedback Q-Learning Value Iteration (VI)

Initialization. Start with an arbitrary control policy u_k^0 with $H^0 = I$. Then, for the following iterations $j = 1, 2, \cdots$, repeat until the convergence criterion is met,

$$\| \bar{H}^j - \bar{H}^{j-1} \| < \epsilon$$

for the constant scalar ϵ that can be set by users according to the requirement of system optimal accuracy.

Policy Evaluation. Determine the least-squares solution of

$$\left(\bar{H}^j\right)^T \bar{z}_k = x_k^T Q x_k + u_k^T R u_k + \left(\bar{H}^{j-1}\right)^T \bar{z}_{k+1}$$

Policy Update. Determine an improved control policy using

$$u_k^{j+1} = -\left(H_{uu}^j\right)^{-1} H_{ux}^j x_k$$

In both Algorithms 1 and 2, the policy evaluation step utilizes the Bellman equation (3.25) to learn the $\bar{H}$ matrix in each iteration, we can rewrite (3.25) in a more compact form, which can be expressed as a linear equation below:

$$\Phi^T \bar{H} = \Upsilon \tag{3.26}$$

which can be converted to the least-squares form as,

$$\bar{H}^j = \left(\Phi\Phi^T\right)^{-1}\Phi\Upsilon \tag{3.27}$$

where the $\Phi \in \mathbb{R}^{l(l+1)/2\times L}$ and the $\Upsilon \in \mathbb{R}^{L\times 1}$ of the state feedback PI algorithm defined as

$$\begin{aligned}\Phi &= \left[\begin{array}{c} \bar{z}_k^1 - \bar{z}_{k+1}^1 \ , \ \bar{z}_k^2 - \bar{z}_{k+1}^2 \ , \ \cdots \ , \ \bar{z}_k^L - \bar{z}_{k+1}^L \end{array}\right] \\ \Upsilon &= \left[\begin{array}{c} r^1\left(x_k, u_k\right) \ , \ r^2\left(x_k, u_k\right) \ , \ \cdots \ , \ r^L\left(x_k, u_k\right) \end{array}\right]^T\end{aligned}$$

and the $\Phi \in \mathbb{R}^{l(l+1)/2\times L}$ and the $\Upsilon \in \mathbb{R}^{L\times 1}$ of the state feedback VI algorithm defined as

$$\begin{aligned}\Phi &= \left[\begin{array}{c} \bar{z}_k^1 \ , \ \bar{z}_k^2 \ , \ \cdots \ , \ \bar{z}_k^L \end{array}\right] \\ \Upsilon &= \left[\begin{array}{c} r^1\left(x_k, u_k\right) + \left(\bar{H}^{j-1}\right)^T \bar{z}_{k+1}^1 \ , \ \cdots \ , \ r^L\left(x_k, u_k\right) + \left(\bar{H}^{j-1}\right)^T \bar{z}_{k+1}^L \end{array}\right]^T\end{aligned}$$

$L \geq l\left(l+1\right)/2$ data samples of u_k, x_k, x_{k+1} need to be collected to form the matrices Φ and Υ in each iteration for both Algorithms 1 and 2.

In addition, from the policy update step in both Algorithms 1 and 2, we know that the policy u_k^{j+1} can be obtained by minimizing the Q-function of the jth policy. We notice that u_k is linearly dependent on x_k, which means that $\Phi\Phi^T$ is singular. In order to guarantee a unique solution to (3.27), we add excitation signals in u_k for both Algorithms 1 and 2. That is, the following rank condition must be satisfied,

$$\text{rank}\left(\Phi\right) = l\left(l+1\right)/2 \tag{3.28}$$

3.3 Output Feedback Optimal Q-Learning Control

Consider a discrete-time linear time-invariant system,

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k,\ x_k \in \mathbb{R}^n,\ u_k \in \mathbb{R}^m \\ y_k &= Cx_k,\ y_k \in \mathbb{R}^p \end{aligned} \tag{3.29}$$

where (A, B) is controllable and (A, C) is observable.

Let the quadratic cost function be

$$J = \sum_{i=0}^{\infty} r\left(x_i, u_i\right) = \sum_{i=0}^{\infty} \left(y_i^T Q_y y_i + u_i^T R u_i\right) \tag{3.30}$$

with the one step utility function $r\left(x_k, u_k\right) = y_k^T Q_y y_k + u_k^T R u_k$ and the user-defined weighting matrices $Q_y = Q_y^T \geq 0$ and $R = R^T > 0$. Let $Q = C^T Q_y C$ and $Q = \sqrt{Q}^T \sqrt{Q}$. Assume that $\left(A, \sqrt{Q}\right)$ is detectable [16].

Then, the output feedback LQR Q-function can be derived as following. In the previous works [12], [13], when the system is observable, the system state can be expressed as below:

$$x_k = M_y \bar{y}_{k-1,k-N} + M_u \bar{u}_{k-1,k-N} \tag{3.31}$$

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

$$\bar{u}_{k-1,k-N} = \begin{bmatrix} u_{k-1} \\ u_{k-2} \\ \vdots \\ u_{k-N} \end{bmatrix}, \quad \bar{y}_{k-1,k-N} = \begin{bmatrix} y_{k-1} \\ y_{k-2} \\ \vdots \\ y_{k-N} \end{bmatrix}$$

The matrix M_y and M_u are defined as,

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

where V_N, U_N, and T_N are the observability matrix, controllability matrix and Toeplitz matrix, defined as,

$$\begin{aligned}
V_N &= \begin{bmatrix} \left(CA^{N-1}\right)^T & \left(CA^{N-2}\right)^T & \cdots & C^T \end{bmatrix}^T \\
U_N &= \begin{bmatrix} B & AB & \cdots & A^{N-1}B \end{bmatrix} \\
T_N &= \begin{bmatrix} 0 & CB & CAB & \cdots & CA^{N-2}B \\ 0 & 0 & CB & \cdots & CA^{N-3}B \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & CB \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}
\end{aligned}$$

Then, (3.31) can be written in terms of inputs and outputs data instead of states as,

$$x_k = \begin{bmatrix} M_u & M_y \end{bmatrix} \begin{bmatrix} \bar{u}_{k-1,k-N} \\ \bar{y}_{k-1,k-N} \end{bmatrix} \tag{3.32}$$

Combing (3.15) and (3.32), the output feedback LQR Q-function can be expressed as,

$$\begin{aligned}
Q_K &= \begin{bmatrix} \bar{u}_{k-1,k-N} \\ \bar{y}_{k-1,k-N} \\ u_k \end{bmatrix}^T \begin{bmatrix} H_{\bar{u}\bar{u}} & H_{\bar{u}\bar{y}} & H_{\bar{u}u} \\ H_{\bar{y}\bar{u}} & H_{\bar{y}\bar{y}} & H_{\bar{y}u} \\ H_{u\bar{u}} & H_{u\bar{y}} & H_{uu} \end{bmatrix} \begin{bmatrix} \bar{u}_{k-1,k-N} \\ \bar{y}_{k-1,k-N} \\ u_k \end{bmatrix} \\
&\triangleq z_k^T H z_k
\end{aligned} \tag{3.33}$$

with $z_k = \begin{bmatrix} \bar{u}_{k-1,k-N}^T & \bar{y}_{k-1,k-N}^T & u_k^T \end{bmatrix}^T$ and the submatrices of H defined as

$$\begin{aligned}
H_{\bar{u}\bar{u}} &= M_u^T \left(Q + A^T P A\right) M_u \in \mathbb{R}^{mN \times mN} \\
H_{\bar{u}\bar{y}} &= M_u^T \left(Q + A^T P A\right) M_y \in \mathbb{R}^{mN \times pN} \\
H_{\bar{u}u} &= M_u^T A^T P B \in \mathbb{R}^{mN \times m} \\
H_{\bar{y}\bar{u}} &= M_y^T \left(Q + A^T P A\right) M_u \in \mathbb{R}^{pN \times mN} \\
H_{\bar{y}\bar{y}} &= M_y^T \left(Q + A^T P A\right) M_y \in \mathbb{R}^{pN \times pN} \\
H_{\bar{y}u} &= M_y^T A^T P B \in \mathbb{R}^{pN \times m} \\
H_{u\bar{u}} &= B^T P A M_u \in \mathbb{R}^{m \times mN} \\
H_{u\bar{y}} &= B^T P A M_y \in \mathbb{R}^{p \times pN} \\
H_{uu} &= R + B^T P B \in \mathbb{R}^{m \times m}
\end{aligned} \tag{3.34}$$

The optimal controller can be obtained by minimizing Q_K in (3.33). By setting $(\partial/\partial u_k)\, Q^* = 0$, the optimal result of u_k can be obtained as,

$$u_k^* = -\left(H_{uu}^*\right)^{-1} \left(H_{u\bar{u}}^* \bar{u}_{k-1,k-N} + H_{u\bar{y}}^* \bar{y}_{k-1,k-N}\right) \tag{3.35}$$

Combing (3.32), (3.34) and (3.35), it is proven that the optimal controller in (3.35) converges to the optimal controller in (3.9), which is obtained by solving the ARE.

Based on the state feedback case, we can obtain the following Bellman equation in terms of inputs and outputs data,

$$\bar{H}^T \bar{z}_k = y_k^T Q_y y_k + u_k^T R u_k + \bar{H}^T \bar{z}_{k+1} \tag{3.36}$$

where we apply the user-defined weighting matrix Q_y on the outputs. The term $x_k^T Q x_k$ can always be substituted by $y_k^T Q_y y_k$ without requiring the knowledge of C

when $y_k = Cx_k$ and $Q = C^T Q_y C$, where y_k is available and x_k is unavailable. In (3.36), u_{k+1} is computed as

$$u_{k+1} = -\left(H_{uu}\right)^{-1}\left(H_{u\bar{u}}\bar{u}_{k,k-N+1} + H_{u\bar{y}}\bar{y}_{k,k-N+1}\right) \tag{3.37}$$

The matrix H is unknown and need to be learned, the reinforcement learning techniques can be applied to learn the optimal H matrix and the optimal controller. The output feedback Q-learning PI and VI algorithms are presented in the following [14].

Algorithm 3: Output Feedback Q-Learning Policy Iteration (PI)

Initialization. Start with a stabilizing control policy u_k^0 with $H^0 = I$. Then, for the following iterations $j = 1, 2, \cdots$, repeat until the convergence criterion is met,

$$\| \bar{H}^j - \bar{H}^{j-1} \| < \epsilon$$

for the constant scalar ϵ that can be set by users according to the requirement of system optimal accuracy.

Policy Evaluation. Determine the least-squares solution of

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

Policy Update. Determine an improved control policy using

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

Algorithm 4: Output Feedback Q-Learning Value Iteration (VI)

Initialization. Start with an arbitrary control policy u_k^0 with $H^0 = I$. Then, for the

following iterations $j = 1, 2, \cdots$, repeat until the convergence criterion is met,

$$\| \bar{H}^j - \bar{H}^{j-1} \| < \epsilon$$

for the constant scalar ϵ that can be set by users according to the requirement of system optimal accuracy.

Policy Evaluation. Determine the least-squares solution of

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

Policy Update. Determine an improved control policy using

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

In both Algorithms 3 and 4, the policy evaluation step utilizes the Bellman equation (3.36), we can rewrite (3.36) in the least-squares form as,

$$\bar{H}^j = \left(\Phi\Phi^T\right)^{-1} \Phi\Upsilon \tag{3.38}$$

where the $\Phi \in \mathbb{R}^{l(l+1)/2 \times L}$ and the $\Upsilon \in \mathbb{R}^{L \times 1}$ of the output feedback PI algorithm defined as

$$\begin{aligned} \Phi &= \left[\bar{z}_k^1 - \bar{z}_{k+1}^1 \ , \ \bar{z}_k^2 - \bar{z}_{k+1}^2 \ , \ \cdots \ , \ \bar{z}_k^L - \bar{z}_{k+1}^L \right] \\ \Upsilon &= \left[r^1\left(y_k, u_k\right) \ , \ r^2\left(y_k, u_k\right) \ , \ \cdots \ , \ r^L\left(y_k, u_k\right) \right]^T \end{aligned}$$

and the $\Phi \in \mathbb{R}^{l(l+1)/2 \times L}$ and the $\Upsilon \in \mathbb{R}^{L \times 1}$ of the output feedback VI algorithm

defined as

$$\begin{aligned}\Phi &= \left[\, \bar{z}_k^1 \,,\, \bar{z}_k^2 \,,\, \cdots \,,\, \bar{z}_k^L \,\right] \\ \Upsilon &= \left[\, r^1\left(y_k, u_k\right) + \left(\bar{H}^{j-1}\right)^T \bar{z}_{k+1}^1 \,,\, \cdots \,,\, r^L\left(y_k, u_k\right) + \left(\bar{H}^{j-1}\right)^T \bar{z}_{k+1}^L \,\right]^T\end{aligned}$$

$L \geq l\left(l+1\right)/2$ data samples of u_k, y_k, $\bar{u}_{k-1,k-N}$, $\bar{y}_{k-1,k-N}$, $\bar{u}_{k,k-N+1}$, $\bar{y}_{k,k-N+1}$ need to be collected to form the matrices Φ and Υ in each iteration for both Algorithms 3 and 4, $l = mN + pN + m$. As in the state feedback case, in order to obtain a unique solution to the matrix H, we add excitation signals in control inputs.

3.4 Summary

In this chapter, we have presented the state feedback Q-learning control method and the output feedback Q-learning control method to solve the LQR optimal stabilization problem for the TWSBR in the absence of any knowledge of the system parameters. Each control method is completely online in nature and consists of two control algorithms, the PI algorithm and the VI algorithm, the former of which requires to start with a stabilizing control policy and the latter can start with an arbitrary control policy. We have utilized a parametrization of the state given by the past input and output data to develop the output feedback control method, which is more desirable in practice due to a reduction in the number of sensors.

In addition, the input decoupling mechanism and the pre-feedback law are able to decouple the original system and moderate the instability of the open-loop system. Both measures help to overcome the computational issues and improve the system behavior in the learning process.

Chapter 4

Simulation Results

In this chapter, we present simulation of both state feedback and output feedback optimal control for the TWSBR. The parameters adopted in the simulation are as follows: $M = 21$ kg, $m = 0.42$ kg, $R = 0.106$ m, $D = 0.44$ m, $L = 0.3$ m, $J_w = 0.0024$ kg·m^2, $J_\delta = 0.3388$ kg·m^2, and $J_p = 0.63$ kg·m^2. The initial states in the simulation are: $x = 0.1$ m, $v = 0.1$ m/s, $\theta = 0.1$ rad, $\omega = 0.1$ rad/s, $\delta = 0.1$ rad and $\dot{\delta} = 0.1$ rad/s. The pre-feedback gain for Subsystem I is set to be $K = \begin{bmatrix} 0 & 0 & -50 & 0 \end{bmatrix}$. The sampling time is 0.1 s. We show here that the proposed control scheme is able to learn the optimal control parameters and stabilize the system.

4.1 State Feedback Optimal Q-Learning Control for the TWSBR

In the simulation of both the state feedback PI and VI algorithms, the weighting matrices are chosen to be $Q = 5 \times I$, $R = 1$, the convergence criterions for Subsystems I and II are set to be $\epsilon = 1$ and $\epsilon = 0.01$, respectively. For Subsystem I, since $l_{\mathrm{I}} = m_{\mathrm{I}} + n_{\mathrm{I}} = 1 + 4 = 5$, we need $L_{\mathrm{I}} = l_{\mathrm{I}}(l_{\mathrm{I}} + 1)/2 = 15$ data samples to satisfy

the rank condition in (3.28) to solve (3.27) in each iteration. For Subsystem II, $l_{\rm II} = m_{\rm II} + n_{\rm II} = 1 + 2 = 3$, $L_{\rm II} = l_{\rm II}(l_{\rm II}+1)/2 = 6$ data samples are required to be collected in each iteration. We use sinusoidal signals with different frequencies as the excitation signal included in the control for both the PI and VI algorithms. The initial policies for Subsystems I and II under the PI algorithm are set to be $K = \begin{bmatrix} -0.5 & -1.5 & 25 & -2.5 \end{bmatrix}$ and $K = \begin{bmatrix} 0.4 & 0.5 \end{bmatrix}$, respectively. We set the initial policies for Subsystems I and II under the VI algorithm to be respectively $K = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}$ and $K = \begin{bmatrix} 0 & 0 \end{bmatrix}$, neither of which is stabilizing. Simulation results of the state trajectory of the closed-loop system under the PI and VI algorithms are shown in Figs. 4.1 and 4.3, respectively. Note that the excitation signal is removed once the convergence criterion is met. Figs. 4.2 and 4.4 show the convergence of the parameter estimates to the optimal values for the PI and VI algorithms, respectively. We compare here the optimal control parameters obtained by the algebraic Riccati equation (ARE) and by the state feedback Q-learning PI and VI algorithms.

By solving the ARE (3.10), which requires the precise knowledge of the system parameters in matrices A and B, we obtain the optimal control matrices and the optimal control policy for the state feedback controller as follows:

Subsystem I:

$$\begin{aligned} H_{ux}^* &= \begin{bmatrix} -4.0091 & -9.7644 & 36.1627 & -7.2177 \end{bmatrix} \\ H_{uu}^* &= 3.2146 \\ K^* &= \begin{bmatrix} -1.2472 & -3.0376 & 11.2497 & -2.2453 \end{bmatrix} \end{aligned}$$

Subsystem II:

$$H_{ux}^* = \begin{bmatrix} 4.0731 & 4.6687 \end{bmatrix}, \quad H_{uu}^* = 3.3180, \quad K^* = \begin{bmatrix} 1.2276 & 1.4071 \end{bmatrix}$$

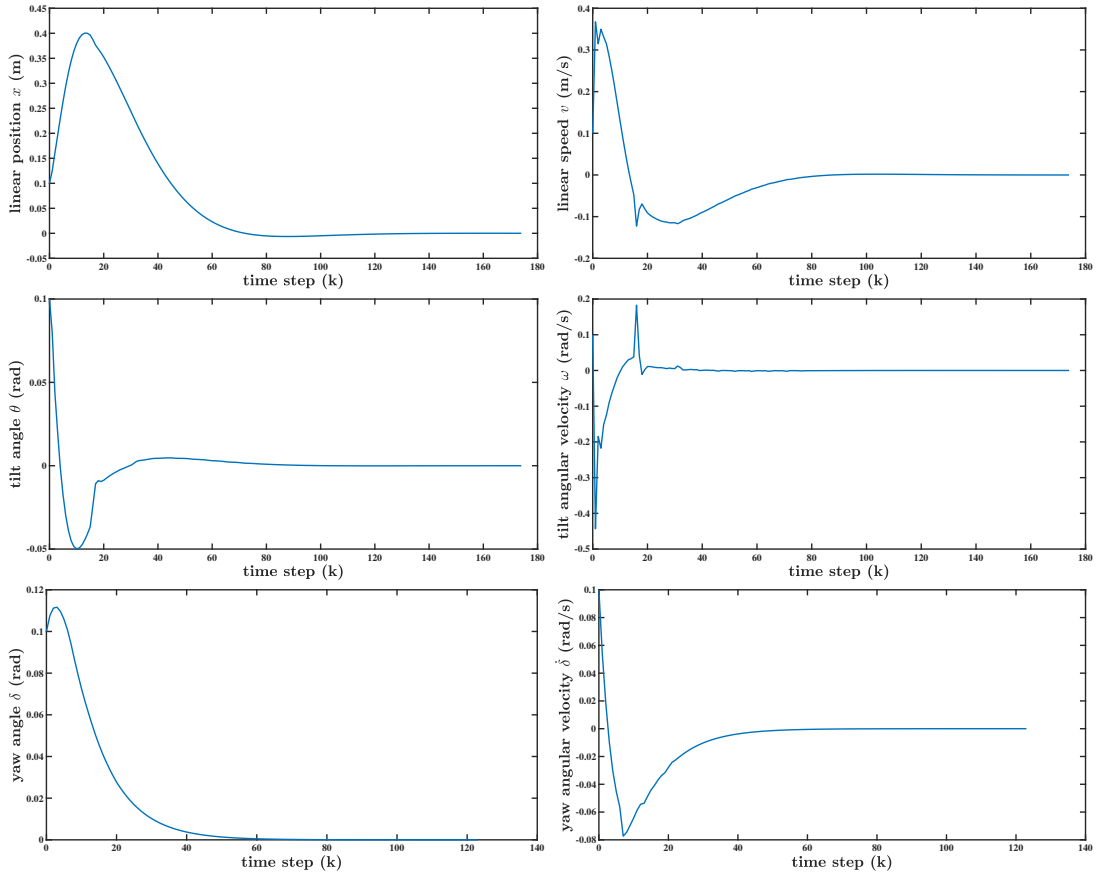
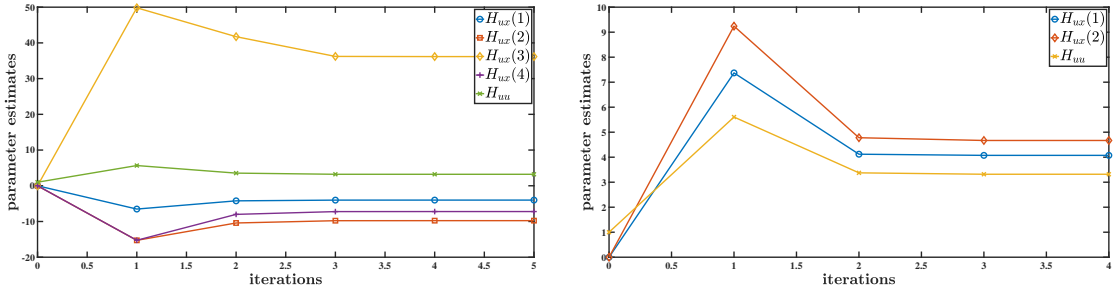


Figure 4.1: State trajectory of the closed-loop system under the state feedback Q-learning PI algorithm.



Subsystem I Subsystem II

Figure 4.2: Convergence of the parameter estimates under the state feedback Q-learning PI algorithm.

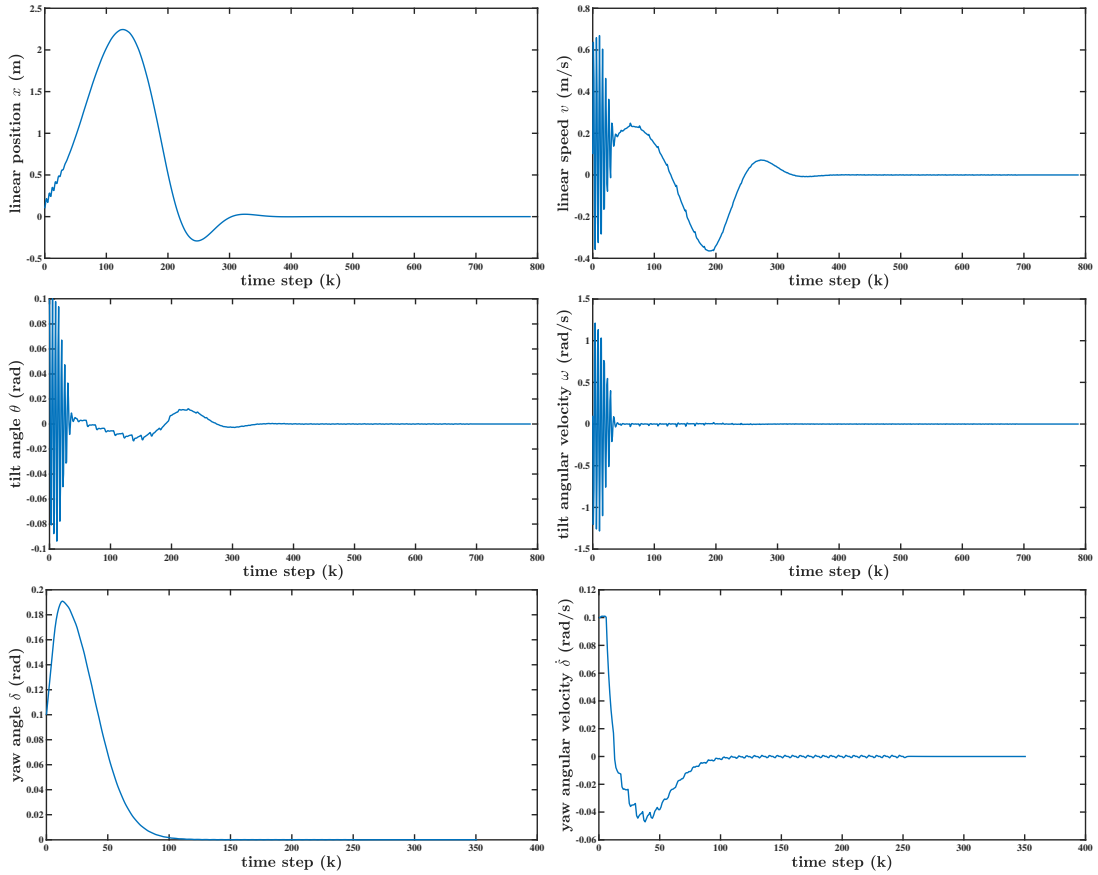
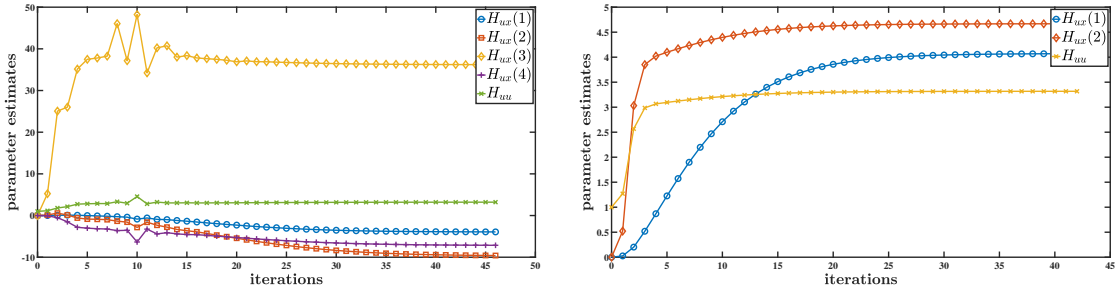


Figure 4.3: State trajectory of the closed-loop system under the state feedback Q-learning VI algorithm.



Subsystem I Subsystem II

Figure 4.4: Convergence of the parameter estimates under the state feedback Q-learning VI algorithm.

The final parameter estimates obtained by the state feedback Q-learning PI algorithm are

Subsystem I:

$$\begin{aligned} \widehat{H}_{ux} &= \begin{bmatrix} -4.0091 & -9.7644 & 36.1628 & -7.2177 \end{bmatrix} \\ \widehat{H}_{uu} &= 3.2146 \\ \widehat{K} &= \begin{bmatrix} -1.2472 & -3.0376 & 11.2497 & -2.2453 \end{bmatrix} \end{aligned}$$

Subsystem II:

$$\widehat{H}_{ux} = \begin{bmatrix} 4.0730 & 4.6686 \end{bmatrix}, \quad \widehat{H}_{uu} = 3.3180, \quad \widehat{K} = \begin{bmatrix} 1.2276 & 1.4071 \end{bmatrix}$$

The final parameter estimates obtained by the state feedback Q-learning VI algorithm are

Subsystem I:

$$\begin{aligned} \widehat{H}_{ux} &= \begin{bmatrix} -3.9286 & -9.5970 & 36.2080 & -7.1424 \end{bmatrix} \\ \widehat{H}_{uu} &= 3.2089 \\ \widehat{K} &= \begin{bmatrix} -1.2243 & -2.9908 & 11.2836 & -2.2258 \end{bmatrix} \end{aligned}$$

Subsystem II:

$$\widehat{H}_{ux} = \begin{bmatrix} 4.0704 & 4.6681 \end{bmatrix}, \quad \widehat{H}_{uu} = 3.3178, \quad \widehat{K} = \begin{bmatrix} 1.2268 & 1.4070 \end{bmatrix}$$

Simulation results show that the proposed state feedback optimal controllers learned by the PI and VI algorithms are able to stabilize the TWSBR and converge to the optimal control parameters. In addition, the number of iterations of the

PI algorithm is less than that of the VI algorithm as a result of starting with a stabilizing control policy, and therefore, the PI algorithm has a better state response. In practice, if an initial stabilizing control policy can be obtained by some preliminary knowledge of the TWSBR system, we can directly use the PI algorithm. If not, we will use the VI algorithm to learn the optimal controller and stabilize the TWSBR.

In addition, for the original 6th order system without the decoupling mechanism, since $l_{\mathrm{o}} = m_{\mathrm{o}} + n_{\mathrm{o}} = 2 + 6 = 8$, we need to collect $L_{\mathrm{o}} = l_{\mathrm{o}}(l_{\mathrm{o}} + 1)/2 = 36$ data samples in each iteration. It is evident from the rank condition in (3.28) that the more unknown parameters we have corresponding to H, the more data samples we require in each learning iteration of the PI and VI algorithms, and the longer the learning transient phase lasts. The number of data samples needed in each iteration of the decoupled system is $L_{\mathrm{I}} + L_{\mathrm{II}} = 15 + 6 = 21$, which is less than that required of the original 6th order system. In other words, the decoupling mechanism is able to reduce the computational complexity as well as shorten the learning time since each learning iteration now takes fewer time steps due to a reduction in the number of unknown parameters. Therefore, the overall learning transient phase is shortened, which is quite desirable.

Since Subsystem I in the absence of the pre-feedback law is strongly unstable, it is hard to satisfy the rank condition in (3.28) in every iteration. The purpose of implementing the pre-feedback law for Subsystem I is to render Subsystem I from exponentially unstable to polynomially unstable. The pre-feedback law is able to relocate the two real poles of opposite signs to the imaginary axis. In other words, the pre-feedback law helps to improve the transient performance during the learning phase by preventing the system trajectory from diverging exponentially to higher magnitudes. Both the input decoupling and the pre-feedback have the advantage of making the learning algorithms easier and faster to converge to the optimal solution.

4.2 Output Feedback Optimal Q-Learning Control for the TWSBR

In the simulation of both the output feedback PI and VI algorithms, we only need to observe the linear position, tilt angle and yaw angle of the TWSBR instead of measuring all six states. The weighting indices are chosen to be $Q_y = 5$, $R = 1$, and the convergence criterions for Subsystems I and II are $\epsilon = 10$ and $\epsilon = 0.1$, respectively. For Subsystem I, since $l_{\rm I} = m_{\rm I}N_{\rm I} + p_{\rm I}N_{\rm I} + m_{\rm I} = 1 \times 2 + 2 \times 2 + 1 = 7$, we need $L_{\rm I} = l_{\rm I}(l_{\rm I} + 1)/2 = 28$ data samples in each iteration. For Subsystem II, $l_{\rm II} = m_{\rm II}N_{\rm II} + p_{\rm II}N_{\rm II} + m_{\rm II} = 1 \times 2 + 1 \times 2 + 1 = 5$, $L_{\rm II} = l_{\rm II}(l_{\rm II} + 1)/2 = 15$ data samples are collected in each iteration. Sinusoids of different frequencies are added in the control to satisfy the excitation condition for both the PI and VI algorithms. The initial policies for Subsystems I and II under the PI algorithm are set to be $K = \begin{bmatrix} 0.18 & -0.36 & -16 & 28 & 15 & -12 \end{bmatrix}$ and $K = \begin{bmatrix} 0.27 & 0.14 & 5.8 & -5.4 \end{bmatrix}$, respectively. We set the initial policies for Subsystems I and II under the VI algorithm to be $K = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ and $K = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}$, respectively. Simulation results of the state trajectory of the closed-loop system under the PI and VI algorithms are shown in Figs. 4.5 and 4.7, respectively. Again, the excitation signal is removed once the convergence criterion is met. Figs. 4.6 and 4.8 show the convergence of the parameter estimates to the optimal values for the PI and VI algorithms, respectively. We compare here the optimal control parameters obtained by the algebraic Riccati equation (ARE) and by the output feedback Q-learning PI and VI algorithms.

By solving the ARE, we obtain the optimal control matrices and the optimal control policy for our Q-learning-based output feedback controller as follows:

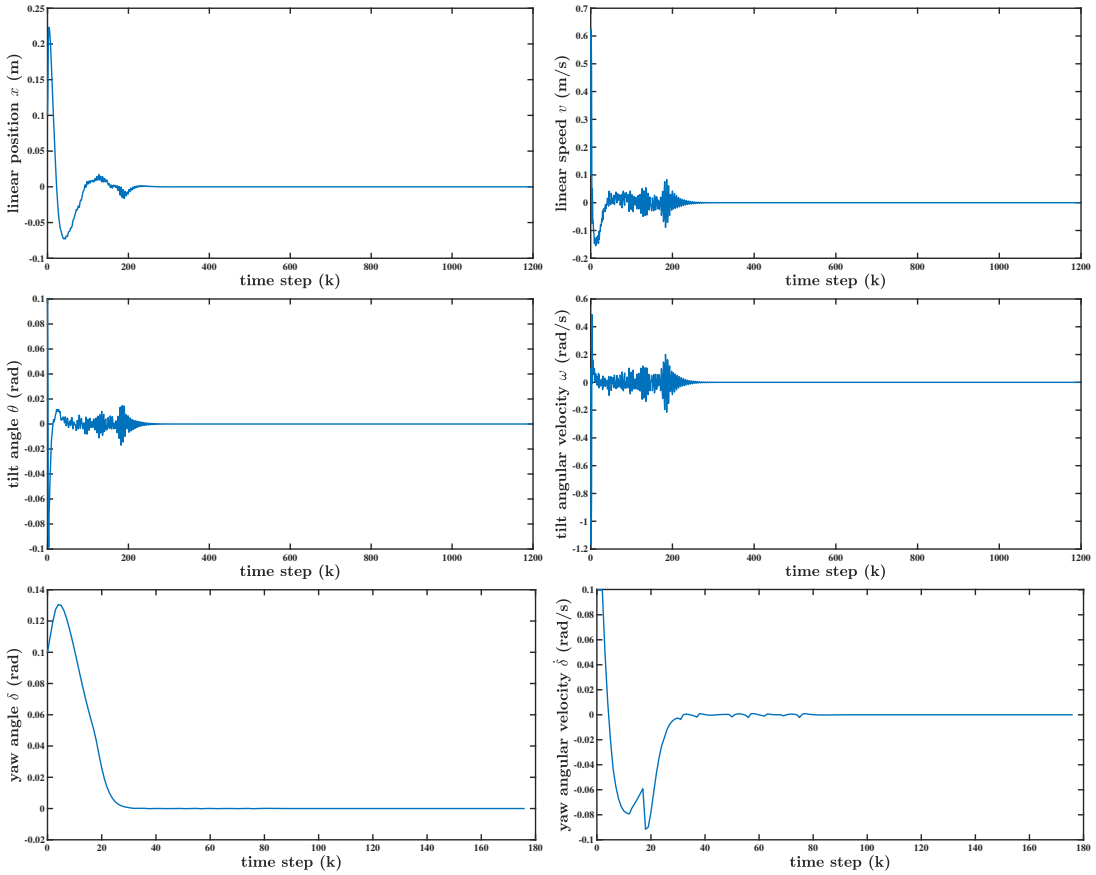


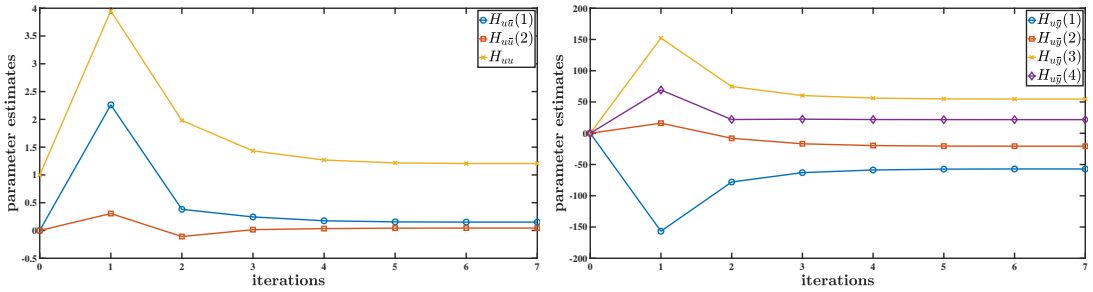
Figure 4.5: State trajectory of the closed-loop system under the output feedback Q-learning PI algorithm.

Subsystem I:

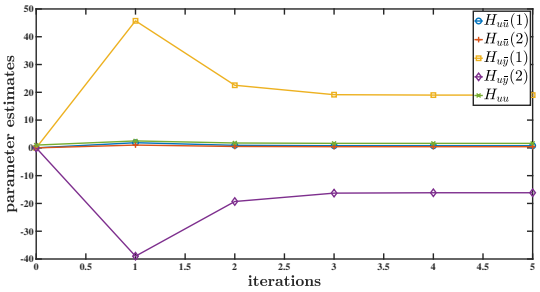
$$
\begin{aligned}
H^*_{u\bar{y}} &= \begin{bmatrix} -57.1342 & -20.7329 & 54.6796 & 21.7755 \end{bmatrix} \\
H^*_{u\bar{u}} &= \begin{bmatrix} 0.1506 & 0.0433 \end{bmatrix}, \quad H^*_{uu} = 1.2050 \\
K^* &= \begin{bmatrix} 0.1250 & 0.0359 & -47.4127 & -17.2052 & 45.3757 & 18.0703 \end{bmatrix}
\end{aligned}
$$

Subsystem II:

$$
\begin{aligned}
H^*_{u\bar{y}} &= \begin{bmatrix} 18.9972 & -16.1536 \end{bmatrix}, \quad H^*_{u\bar{u}} = \begin{bmatrix} 0.7648 & 0.4193 \end{bmatrix}, \quad H^*_{uu} = 1.6172 \\
K^* &= \begin{bmatrix} 0.4729 & 0.2593 & 11.7471 & -9.9888 \end{bmatrix}
\end{aligned}
$$



Subsystem I



Subsystem II

Figure 4.6: Convergence of the parameter estimates under the output feedback Q-learning PI algorithm.

The final parameter estimates obtained by the output feedback Q-learning PI algorithm are

Subsystem I:

$$\begin{aligned}
\widehat{H}_{u\bar{y}} &= \begin{bmatrix} -57.1354 & -20.7335 & 54.6808 & 21.7760 \end{bmatrix} \\
\widehat{H}_{u\bar{u}} &= \begin{bmatrix} 0.1506 & 0.0433 \end{bmatrix}, \quad \widehat{H}_{uu} = 1.2050 \\
\widehat{K} &= \begin{bmatrix} 0.1250 & 0.0359 & -47.4137 & -17.2056 & 45.3768 & 18.0708 \end{bmatrix}
\end{aligned}$$

Subsystem II:

$$\begin{aligned}
\widehat{H}_{u\bar{y}} &= \begin{bmatrix} 18.9972 & -16.1536 \end{bmatrix}, \quad \widehat{H}_{u\bar{u}} = \begin{bmatrix} 0.7648 & 0.4193 \end{bmatrix}, \quad \widehat{H}_{uu} = 1.6172 \\
\widehat{K} &= \begin{bmatrix} 0.4729 & 0.2593 & 11.7471 & -9.9888 \end{bmatrix}
\end{aligned}$$

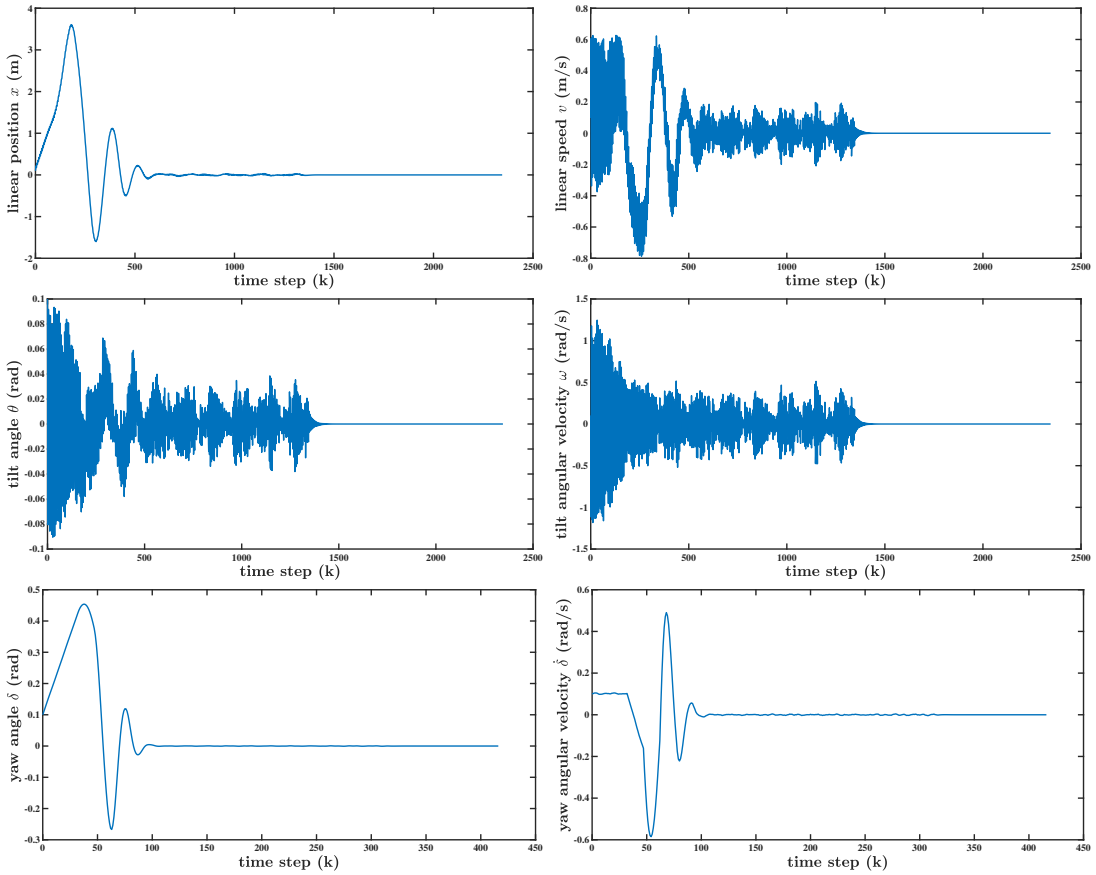
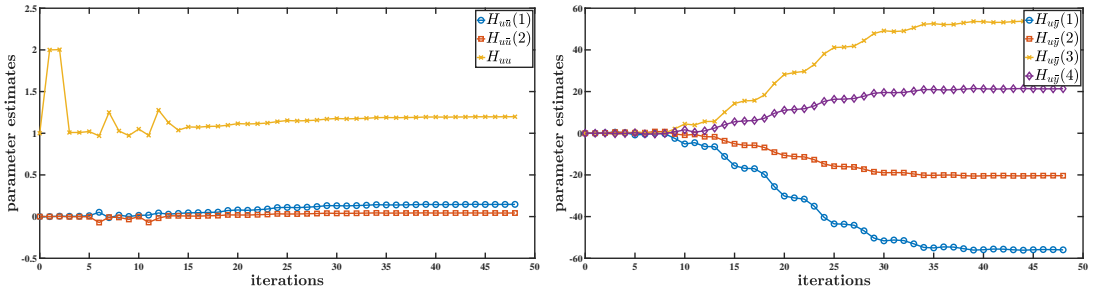


Figure 4.7: State trajectory of the closed-loop system under the output feedback Q-learning VI algorithm.

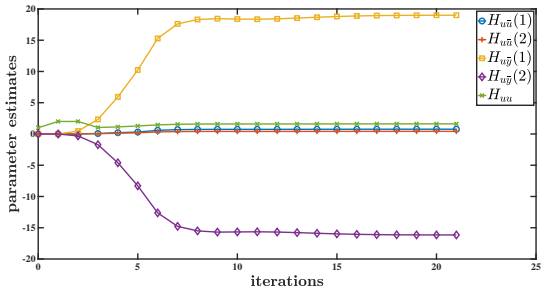
The final parameter estimates obtained by the output feedback Q-learning VI algorithm are

Subsystem I:

$$\begin{aligned}
\widehat{H}_{u\bar{y}} &= \begin{bmatrix} -55.9625 & -20.3872 & 53.5987 & 21.3703 \end{bmatrix} \\
\widehat{H}_{u\bar{u}} &= \begin{bmatrix} 0.1468 & 0.0430 \end{bmatrix}, \quad \widehat{H}_{uu} = 1.1983 \\
\widehat{K} &= \begin{bmatrix} 0.1225 & 0.0359 & -46.7017 & -17.0135 & 44.7291 & 17.8339 \end{bmatrix}
\end{aligned}$$



Subsystem I



Subsystem II

Figure 4.8: Convergence of the parameter estimates under the output feedback Q-learning VI algorithm.

Subsystem II:

$$\widehat{H}_{u\bar{y}} = \begin{bmatrix} 18.9958 & -16.1523 \end{bmatrix}, \quad \widehat{H}_{u\bar{u}} = \begin{bmatrix} 0.7647 & 0.4193 \end{bmatrix}, \quad \widehat{H}_{uu} = 1.6171$$

$$\widehat{K} = \begin{bmatrix} 0.4729 & 0.2593 & 11.7467 & -9.9884 \end{bmatrix}$$

Simulation results show that the proposed output feedback optimal controllers learned by the PI and VI algorithms are able to realize the goal of optimal control for the TWSBR.

In addition, the number of data samples required to be collected in each iteration of the decoupled system is $L_{\mathrm{I}} + L_{\mathrm{II}} = 28 + 15 = 43$. The number of data samples that need to be collected in each iteration of the original 6th order system for our proposed output feedback control is $L_{\mathrm{o}} = l_{\mathrm{o}}(l_{\mathrm{o}}+1)/2 = (m_{\mathrm{o}}N_{\mathrm{o}} + p_{\mathrm{o}}N_{\mathrm{o}} + m_{\mathrm{o}})(m_{\mathrm{o}}N_{\mathrm{o}} + p_{\mathrm{o}}N_{\mathrm{o}} +$

$m_o + 1)/2 = 78$, which is higher than that required of the decoupled system. Clearly, the decoupling mechanism also helps to overcome the computational issues and significantly shorten the learning transient in the output feedback case. The pre-feedback law in the output feedback case helps to improve the learning transient behavior in the same way as in the state feedback case. Both the input decoupling measure and the pre-feedback measure in the output feedback case have the same positive effects on the implementation of the learning algorithms as in the state feedback case.

Furthermore, compared to the state feedback learning algorithms, the output feedback learning algorithms require more data samples due to more unknowns, and thereby the optimal control policies are learned slower. However, the output feedback learning algorithms have the obvious advantage of requiring fewer sensors, which improves the reliability and the cost effectiveness of the system and is more desirable.

4.3 Robustness of the Learned Optimal Control Policy

We now examine the robustness of the optimal control policy obtained by the proposed Q-learning algorithms. When the TWSBR achieves stabilization and the convergence criteria is satisfied, we increase the mass of the robot body M, which means that in practice the robot begins to carry a load, such as a package, after the learning phase. The robustness of the optimal control policy can be determined by observing whether the robot with a load can maintain stabilization and what the maximum load is. Simulation of the state trajectory of the TWSBR with a load under the state feedback PI and VI algorithms are shown in Figs. 4.9 and 4.10, respectively. Simulation of the state trajectory of the TWSBR with a load under the output feedback PI and VI algorithms are shown in Figs. 4.11 and 4.12, respectively.

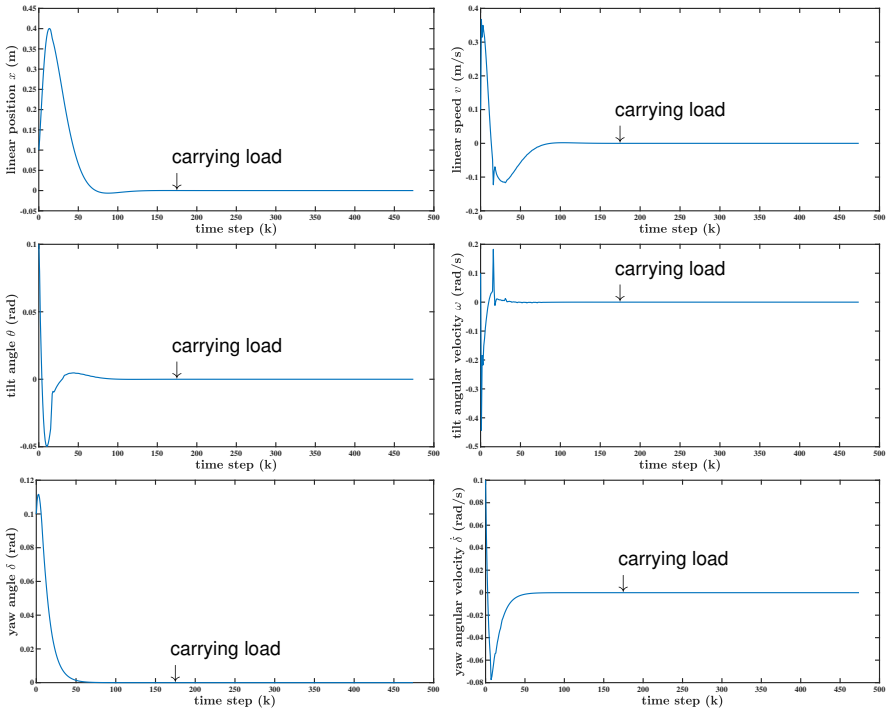


Figure 4.9: State trajectory of the closed-loop system with a 20 kg load under the state feedback Q-learning PI algorithm.

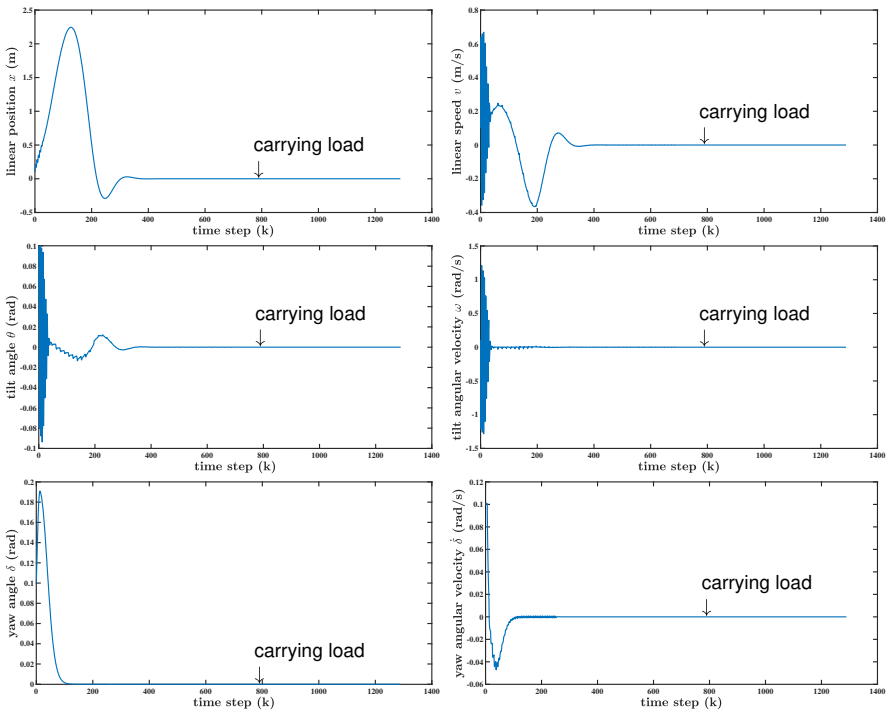


Figure 4.10: State trajectory of the closed-loop system with a 20 kg load under the state feedback Q-learning VI algorithm.

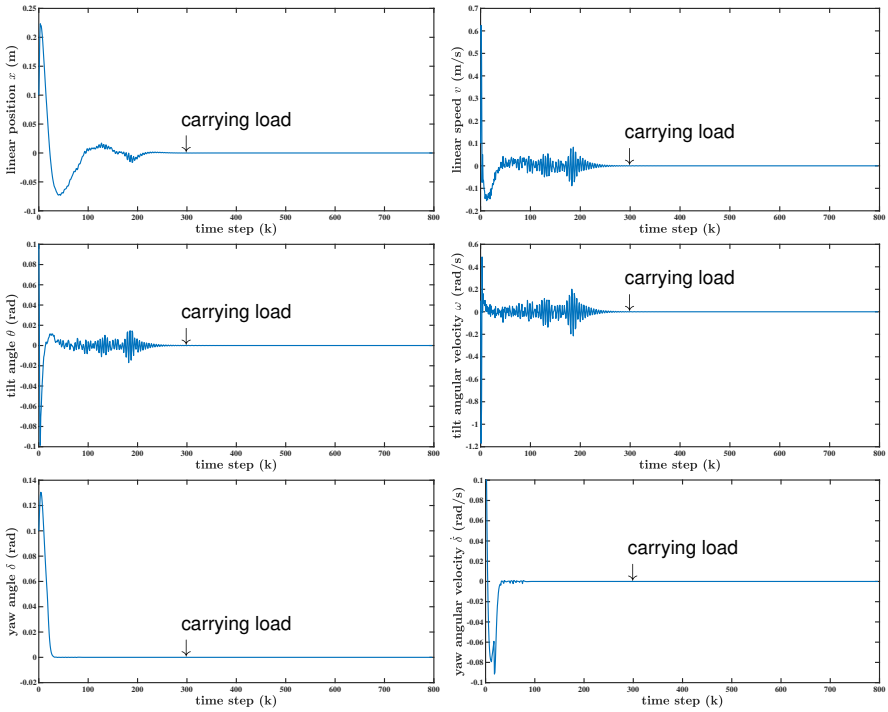


Figure 4.11: State trajectory of the closed-loop system with an 11 kg load under the output feedback Q-learning PI algorithm.

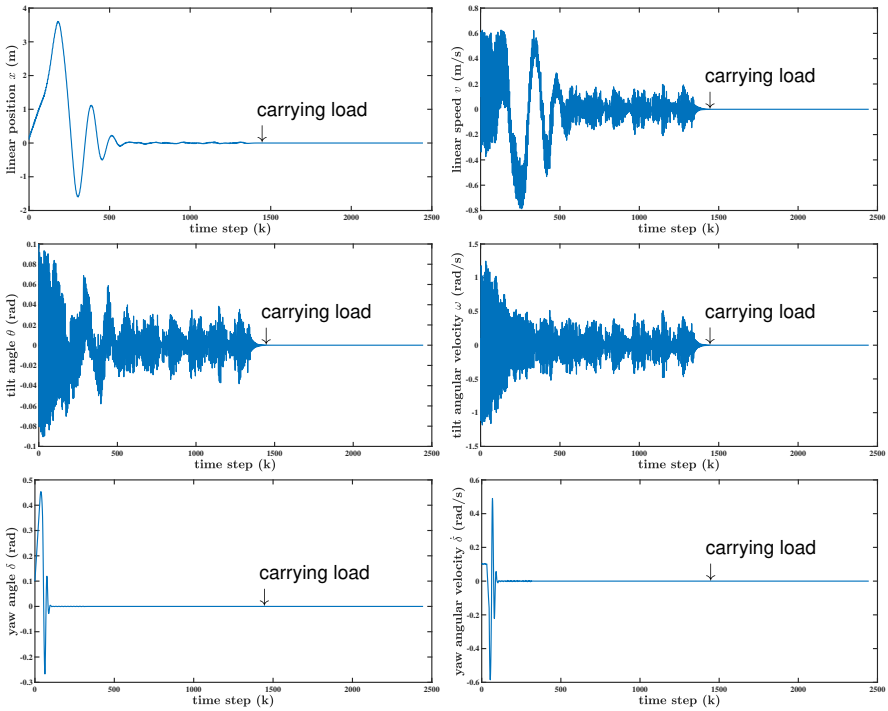


Figure 4.12: State trajectory of the closed-loop system with an 11 kg load under the output feedback Q-learning VI algorithm.

After several simulation attempts, we determined the maximum load for the TWSBR. The TWSBR under the proposed state feedback Q-learning control method can carry a maximum load of 20 kg, while the TWSBR under the proposed output feedback Q-learning control method is able to carry a maximum load of 11 kg. Simulation results show that the TWSBR with a certain load is capable of remaining stable operation, the robustness of the optimal control policy obtained by the proposed state feedback and output feedback Q-learning control methods is verified.

4.4 Summary

In this chapter, we have presented simulation of state feedback and output feedback Q-learning control for the TWSBR. Simulation results show that the optimal control parameters obtained by the proposed Q-learning algorithms converge to the optimal control parameters obtained through solving the ARE, which means the proposed state feedback and output feedback control methods are able to realize optimal control of the TWSBR. The output feedback method uses fewer sensors but learns the optimal parameters slower than the state feedback method. The PI algorithm has a better state response, while the VI algorithm eliminates the need of starting with a stabilizing control policy. The input decoupling measure and the pre-feedback measure help to overcome the computational issues in implementing the learning algorithms and make the learning algorithms easier and faster to converge to the optimal parameters. The robustness of the optimal control policy learned by the proposed Q-learning control is verified by showing through simulation that the robot is able to carry a sizable load after the learning phase and maintain stable operation.

Chapter 5

Conclusions and Future Research Topics

5.1 Conclusions

In this thesis, we addressed optimal control of the two-wheeled self-balancing robot in the absence of the knowledge of the system parameter values. The proposed control scheme uses a completely online, feedback-based Q-learning method to realize optimal control of the robot. The optimal control parameters obtained by the proposed Q-learning algorithms converge to the optimal control parameters solved by the algebraic Riccati equation. Both state feedback and output feedback were considered. The output feedback method requires fewer sensors, while the state feedback method learns the optimal parameters faster. Both the Policy Iteration (PI) algorithm and the Value Iteration (VI) algorithm were presented. The VI algorithm is able to start with an arbitrary control policy, while the PI algorithm has a better state response. The adoption of the input decoupling mechanism and the pre-feedback law have helped to overcome the commonly encountered numerical difficulties associated with high

dimensionality and strong instability of the system in applying Q-learning control algorithms. Extensive simulation shows that the input decoupling mechanism shortens the learning transient phase, the pre-feedback law improves the system behavior by preventing the system trajectory from diverging exponentially to higher magnitudes, and the proposed control results in stabilizing and robust optimal controllers.

5.2 Future Research Topics

Optimal control for the TWSBR by Q-learning, as discussed and analyzed in this thesis, is a new research topic. This work also gives rise to some new questions and problems for future work, such as,

1. Solving general optimal tracking control for the TWSBR in the absence of the knowledge of the system parameters.

2. Extensions to optimal control of the two-wheeled human transportation vehicle with a human load.

3. Relaxing the excitation conditions for safe learning practices.

Bibliography

[1] Z. Li, C. Yang, and L. Fan, *Advanced Control of Wheeled Inverted Pendulum Systems.* London, United Kingdom: Springer, 2012.

[2] S. C. Lin, C. C. Tsai, and H. C. Huang, "Adaptive robust self-balancing and steering of a two-wheeled human transportation vehicle," *Journal of Intelligent & Robotic Systems*, vol. 62, no. 1, pp. 103-123, 2011.

[3] Z. Li and C. Xu, "Adaptive fuzzy logic control of dynamic balance and motion for wheeled inverted pendulums," *Fuzzy Sets and Systems*, vol. 160, no. 12, pp. 1787-1803, 2009.

[4] J. Huang, Z.-H. Guan, T. Matsuno, T. Fukuda, and K. Sekiyama, "Sliding-mode velocity control of mobile-wheeled inverted-pendulum systems," *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 750-758, 2010.

[5] S. C. Lin, C. C. Tsai, and H. C. Huang, "Nonlinear adaptive sliding-mode control design for two-wheeled human transportation vehicle," in *Proceedings of the 2009 IEEE International Conference on Systems, Man and Cybernetics*, 2009, pp. 1965–1970.

[6] S. Jeong and T. Takahashi, "Wheeled inverted pendulum type assistant robot: design concept and mobile control," *Intelligent Service Robotics*, vol. 1, no. 4, pp. 313-320, 2008.

[7] C. Xu, M. Li, and F. Pan, "The system design and LQR control of a two-wheels self-balancing mobile robot," in *Proceedings of the 2011 International Conference on Electrical and Control Engineering*, 2011, pp. 2786-2789.

[8] K. D. Do and G. Seet, "Motion control of a two-wheeled mobile vehicle with an inverted pendulum," *Journal of Intelligent* & Robotic Systems, vol. 60, no. 3-4, pp. 577-605, 2010.

[9] S. J. Bradtke, B. E. Ydstie, and A. G. Barto, "Adaptive linear quadratic control using policy iteration," in *Proceedings of the 1994 American Control Conference*, 1994, pp. 3475-3479.

[10] T. Landelius, "Reinforcement learning and distributed local model synthesis," Ph.D. dissertation, Linköping, Sweden: Linköping University Electronic Press, 1997.

[11] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits and Systems Magazine*, vol. 9, no. 3, pp. 32-50, 2009.

[12] 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.

[13] S. A. A. Rizvi and Z. Lin, "Output feedback reinforcement Q-learning control for the discrete-time linear quadratic regulator problem," in *Proceedings of the 2017 IEEE 56th Annual Conference on Decision and Control*, 2017, pp. 1311-1316.

[14] S. A. A. Rizvi and Z. Lin, "Output feedback Q-learning control for the discrete-time linear quadratic regulator problem," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 5, pp. 1523-1536, 2018.

[15] F. Grasser, A. D'arrigo, S. Colombi, and A. Rufer, "JOE: a mobile, inverted pendulum," *IEEE Transactions on Industrial Electronics*, vol. 49, no. 1, pp. 107-114, 2002.

[16] F. L. Lewis and V. L. Syrmos, *Optimal Control.* New York, NY: John Wiley & Sons, 1995.

[17] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction.* Cambridge, MA: MIT Press, 1998.

[18] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, "Reinforcement learning and feedback control: using natural decision methods to design optimal adaptive controllers," *IEEE Control Systems Magazine*, vol. 32, no. 6, pp. 76-105, 2012.

[19] J. Wu, W. Zhang, and S. Wang, "A two-wheeled self-balancing robot with the fuzzy PD control method," *Mathematical Problems in Engineering*, vol. 2012, 2012.

[20] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming.* Belmont, MA: Athena Scientific, 1996.

[21] F.-Y. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: an introduction," *IEEE Computational Intelligence Magazine*, vol. 4, no. 2, pp. 39-47, 2009.

[22] S. A. A. Rizvi and Z. Lin, "Experience replay–based output feedback Q-learning scheme for optimal output tracking control of discrete-time linear systems," *Inter-*

national Journal of Adaptive Control and Signal Processing, vol. 33, no. 12, pp. 1825-1842, 2019.

[23] K. Pathak, J. Franch, and S. K. Agrawal, "Velocity and position control of a wheeled inverted pendulum by partial feedback linearization," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 505-513, 2005.

[24] S. W. Nawawi, M. N. Ahmad, and J. H. S. Osman, "Development of a two-wheeled inverted pendulum mobile robot," in *Proceedings of the 2007 5th Student Conference on Research and Development*, 2007, pp. 1-5.

[25] C.-H. Huang, W.-J. Wang, and C.-H. Chiu, "Velocity control realisation for a self-balancing transporter," *IET control theory & applications*, vol. 5, no. 13, pp. 1551-1560, 2011.

[26] S. C. Lin and C. C. Tsai, "Development of a self-balancing human transportation vehicle for the teaching of feedback control," *IEEE Transactions on Education*, vol. 52, no.1, pp. 157-168, 2009.

[27] T. Nomura, Y. Kitsuka, H. Suemitsu, and T. Matsuo, "Adaptive backstepping control for a two-wheeled autonomous robot," in *Proceedings of the ICROS-SICE International Joint Conference*, 2009, pp. 4687-4692.

[28] 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.

[29] S. A. A. Rizvi, "Reinforcement Learning for Model-Free Output Feedback Optimal Control," Ph.D. dissertation, University of Virginia, 2020.

[30] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. dissertation, King's College, Cambridge, 1989.