

ROTATIONAL DOUBLE INVERTED PENDULUM

Thesis

Submitted to

The School of Engineering of the

UNIVERSITY OF DAYTON

In Partial Fulfillment of the Requirements for

The Degree of

Master of Science in Electrical Engineering

By

Bo Li

UNIVERSITY OF DAYTON

Dayton, Ohio

August, 2013



ROTATIONAL DOUBLE INVERTED PENDULUM

Name: Li, Bo

APPROVED BY:

Raúl Ordóñez, Ph.D.
Advisor Committee Chairman
Associate Professor, Department of
Electrical and Computer Engineering

Vijayan K. Asari, Ph.D.
Committee Member
Professor, Department of Electrical and
Computer Engineering

Ralph Barrera, Ph.D.
Committee Member
Adjunct Professor, Department of
Electrical and Computer Engineering

John G. Weber, Ph.D.
Associate Dean
School of Engineering

Tony E. Saliba, Ph.D.
Dean, School of Engineering
& Wilke Distinguished Professor

© Copyright by

Bo Li

All rights reserved

2013

ABSTRACT

ROTATIONAL DOUBLE INVERTED PENDULUM

Name: Li, Bo
University of Dayton

Advisor: Dr. Raúl Ordóñez

The thesis deals with the stabilization control of the Rotational Double Inverted Pendulum (RDIP) System. The RDIP is an extremely nonlinear, unstable, underactuated system of high order. A mathematical model is built for the RDIP with the Euler-Lagrange (E-L) equation. A Linear Quadratic Regulator (LQR) controller is designed for this system and its stability analysis is presented in the Lyapunov method. We re-develop the Direct Adaptive Fuzzy Control (DAFC) method in our case for the purpose of exploring the possibility to improve the performance of the LQR control of the system. The simulation results of these two control schemes with their comparative analysis show that the DAFC is able to enhance the LQR controller by increasing its robustness in the RDIP control.

For my family

ACKNOWLEDGMENTS

I would like to express my special gratitude to my advisor Dr. Raúl Ordóñez for his tremendous support and help through the learning process of this master thesis. Without his guidance and persistent help this project would not have been possible within the limited time frame. Furthermore I would also like to thank my committee members, Dr. Asari and Dr. Barrera, who have willingly shared their precious time and provided me with the useful comments, remarks and engagement on the dissertation. In addition, a thank you for all the members in the lab KL302, whose companion and suggestions have supported me during the process of this thesis and helped me a lot. Last but not the least, I would like to thank my loved ones for their endless love and support. I will be grateful forever for your love.

TABLE OF CONTENTS

ABSTRACT	iii
DEDICATION	iv
ACKNOWLEDGMENTS	v
LIST OF FIGURES	viii
LIST OF TABLES	ix
I. INTRODUCTION	1
II. SYSTEM DESCRIPTION	4
2.1 Rotational Double Inverted Pendulum Configurations	4
2.2 Euler-Lagrange Equation	8
2.3 Modeling	9
III. CONTROL USING LINEAR QUADRATIC REGULATOR	14
3.1 Introduction	14
3.2 Implementation	17
3.3 LQR Tuning	20
3.4 Stability Analysis	22
IV. ADAPTIVE CONTROL	29
4.1 Introduction	29
4.2 Takagi-Sugeno Fuzzy System	31
4.3 Theory	33
4.3.1 Bounding Control	37
4.3.2 Adaptation Algorithm	37
4.3.3 Sliding-mode Control	38
4.4 Implementation	39

V.	SIMULATION RESULTS	42
5.1	Open-loop Model Verification	42
5.2	Linear Quadratic Regulator Results	44
5.3	Direct Adaptive Fuzzy Control Results	47
VI.	CONCLUSION	53
	BIBLIOGRAPHY	56

LIST OF FIGURES

2.1	Rotational double inverted pendulum schematic.	5
2.2	Velocity analysis for Link 2.	10
4.1	Membership function with $c = 1$ and $\sigma = 0.25$	32
5.1	Open-loop simulation in the initial states $\theta_1 = 57.2958^\circ$, $\theta_2 = 5.7296^\circ$ and $\theta_3 = 5.7296^\circ$	43
5.2	LQR simulation result with $Q = I^6$ and $R = I$	45
5.3	Tuned LQR simulation result.	46
5.4	Control signals of the LQR and its tuned version.	47
5.5	DAFC simulation result.	49
5.6	DAFC control signal.	50
5.7	LQR vs. DAFC.	52

LIST OF TABLES

3.1	R.O.A searching result with $q_1 = \pi$	27
3.2	Largest R.O.A for different q_1	27
5.1	Membership function settings for the DAFC.	48

CHAPTER I

INTRODUCTION

An inverted pendulum is a pendulum which has its links rotating above its pivot point. It is often implemented either with the pivot point connected with a base arm that can rotate horizontally (described in [1]) or mounted on a cart that can move in a fixed horizontal line (introduced in [2]). The links of the pendulum are usually limited to 1 degree of freedom by affixing the links to an axis of rotation. It is obvious that an inverted pendulum is inherently unstable, and must be actively balanced in order to remain upright while a normal pendulum is stable when hanging downwards. This can be done by applying a torque at the pivot point for a rotational inverted pendulum as considered in this thesis or moving the pivot point horizontally for the case of an inverted pendulum on a cart. A simple demonstration of moving the pivot point to control the pendulum is achieved by balancing an upturned broomstick on the end of one's finger. The inverted pendulum control is a classic problem in dynamics and control theory and is used to verify the performance and demonstrating the effectiveness of control algorithms.

The Rotational Double Inverted Pendulum (RDIP) takes the classic rotational single pendulum problem to the next level of complexity. The RDIP is composed of a rotary arm that attaches to a servo system which provides a torque to the base arm to control the whole system, a short bottom rod

connected to the arm and a top long rod. It is an underactuated (i.e., it has fewer inputs than degrees of freedom) and extremely nonlinear unstable system due to the gravitational forces and the coupling arising from the Coriolis and centripetal forces. Since the RDIP presents considerable control-design challenges, it is an attractive tool utilized for developing different control techniques and testing their performances. Related applications include stabilizing the take-off of a multi-stage rocket, as well as modeling the human posture system.

Nearly all works on pendulum control concentrate on two problems: stabilization of the inverted pendulums and pendulums swing-up control design. The first topic is concerned with the controller design to maintain the pendulum in the upright position. In the RDIP case, controllers are designed to balance two vertical rods by manipulating the angle of the base arm. The second one refers to an adequate algorithm to swing up the pendulum from its stable equilibrium [2], the downward position to the upright position. In this thesis we concentrate on the balancing control of the pendulum without investigating swing-up details.

There is a variety of works devoted into the control design of the RDIP. A simple mathematical model for the RDIP has been built in [3], which takes the angles and angular velocities of the base arm and the two pendulums as the system outputs and ignores all the friction terms for the rotational joints and the DC motor. It also presents an alternative of the least squares theory to come up with a controller providing a domain of convergence for the pendulum. A more precise system model has been developed in this paper by using the modeling method mentioned in [4] with the help of the E-L equation. Another control structure is proposed in [5] by compensating individually the multiple loop delays, which is suitable to be used in a networked control system environment. In

this paper, we seek to balance the RDIP with the LQR and the DAFC. We will discuss the details about these two methods in the following chapters.

This paper mainly serves for two purposes. Firstly, the simulation and experiment results in [6] indicates that, even with a non-minimum-phase plant, the adaptive fuzzy controller is still able to make a good control performance for the single inverted pendulum: the angle of the pendulum can converge to the origin, although the base arm trends to rotate with a constant angular velocity, which, in control notation, is another stable state. This paper tries to explore the possibility of this finding in a more complicated case, the RDIP. Furthermore, assuming that this possibility exists, we try to improve the control performance to make all the system states converge to zero. Secondly, this paper also provides a fundamental theoretical basis for the control experiment of the SV02+DBIP double inverted pendulum kit from the Quanser Company in the lab KL302.

This paper is organized as follows. In Chapter II, we present a description of the RDIP and develop a mathematical model for the system. In Chapter III and IV, the LQR and DAFC are introduced and developed for the RDIP in details respectively. Chapter V shows the simulation results of these two controllers and presents an analysis on them. In Chapter VI, the concluding remarks, we summarize the overall results, provide a broad assessment of the apparent advantages and disadvantages of the LQR and the DAFC control techniques, and provide some future research directions which help to identify limitations of the scope and content of this paper.

CHAPTER II

SYSTEM DESCRIPTION

In this Chapter we will focus on the mathematical model building of the RDIP. The more we know about a dynamic system, the more accurate a mathematical model can be obtained. With accurate mathematic models, faster, more accurate and effective controllers can be designed, since mathematic models allow design, test and development of controllers with the help of some powerful engineering softwares such as MATLAB [MathWorks].

2.1 Rotational Double Inverted Pendulum Configurations

The RDIP experiment platform that we use for simulation to be described later consists of a horizontal base arm (denoted as Link 1) driven by a servo motor and two vertical pendulums (denoted as Link 2 and 3) that move freely in the plane perpendicular to Link 1, as shown in Figure 2.1 about which we will talk in details later. Since we will focus on the stabilization of the pendulums, it is convenient to set the coordinate system as in Figure 2.1. In this paper, the mathematical model of the RDIP will be developed by the use of the Euler-Lagrange (E-L) function. A simple mathematical model has been presented in [3], which assumes that the acceleration of the base arm is able

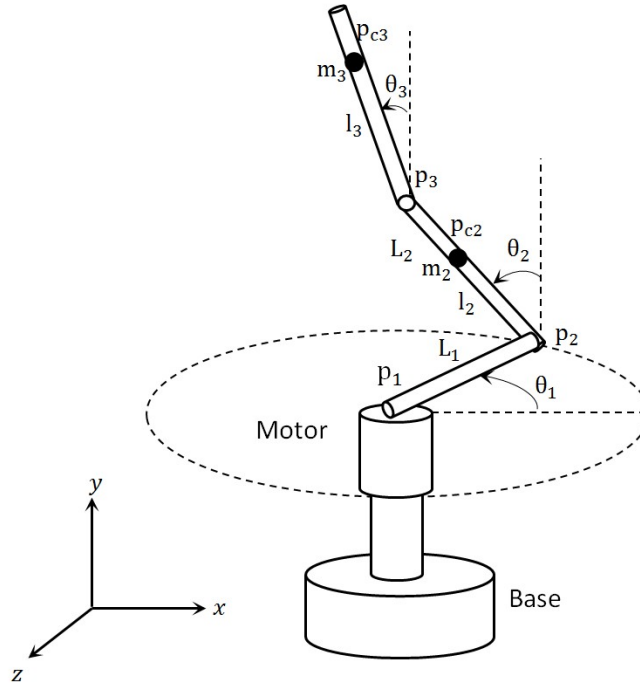


Figure 2.1: Rotational double inverted pendulum schematic.

to be manipulated directly and therefore chosen as the system control input. In this dissertation, a more practical assumption is taken under which the torque of the motor to the base arm is the control signal. Moreover, the pivoting friction factors will be taken care of for the goal to build a more precise model and simulate the real system we have in the lab.

We will use some additional basic assumptions of the system attributes similar with [3]:

- All the link angles and the angular velocities are accessible at each time step, since we know that we can access these data with the help of the encoders on the links and high rate of data acquisition is possible for our experiment platform.

- The viscous frictions of the arm and the two pendulums are considered while the static friction, backlash and plane slackness are ignored.
- The apparatus is light weight and has low inertia resulting in a structure with low stiffness and a tendency to vibrate.
- System dynamics is slow enough to be controlled.

Figure 2.1 shows the basic configurations of the RDIP. The arrows on the arcs show the positive direction for the rotary movement of the links. The straight dash lines denote the origin of the displacement of the link angles. For example when the horizontal Link 1 is centered and the vertical Link 2 and 3 are in the upright position, all of the position variables are zero. The state variables of the links are:

θ_1 Angle of Link 1 in the horizontal plane.

$\dot{\theta}_1$ Velocity of Link 1 in the horizontal plane.

$\ddot{\theta}_1$ Acceleration of Link 1.

θ_2 Angle of Link 2 in the vertical plane.

$\dot{\theta}_2$ Velocity of Link 2 in the vertical plane.

$\ddot{\theta}_2$ Acceleration of Link 2.

θ_3 Angle of Link 3 in the vertical plane.

$\dot{\theta}_3$ Velocity of Link 3 in the vertical plane.

$\ddot{\theta}_3$ Acceleration of Link 3.

Some additional definitions are needed here as follows:

J_i Moment of inertia of Link i . J_1 is about its pivot while J_i is about its center of mass P_{ci} for $i = 2, 3$.

l_i Distance from the center of rotation of Link i to its center of mass, $i = 1, 2, 3$.

m_i Mass of Link i , $i = 1, 2, 3$.

g Gravity with the value $g = 9.81m/s^2$ towards the center of the earth.

L_i Length of Link i , $i = 1, 2, 3$.

b_i Viscous damping coefficient of the bearing on which Link i rotates, $i = 1, 2, 3$.

2.2 Euler-Lagrange Equation

The E-L method, introduced in details in [7], is applied in the derivation of the equations of motion for the RDIP dynamics since the Newtonian approach of applying Newton's laws of motion is highly complicated in this case. The solutions to the E-L equation for the action of a system are capable of describing the evolution of a physical system according to the Hamilton's principle of stationary action in Lagrangian mechanics. In classical mechanics, it is equivalent to Newton's laws of motion, but it has the advantage that it takes the same form in any system of generalized coordinates, and it is better suited to generalizations.

The E-L equation is an equation satisfied by a function q , of a real argument t , which is a stationary point of the functional

$$S(q) = \int_a^b L(t, q(t), \dot{q}(t)) dt, \quad (2.1)$$

where q is the function to be found: $q : [a, b] \subset \mathbb{R} \rightarrow X$, $t \mapsto x = q(t)$ such that q is differentiable, $q(a) = x_a$ and $q(b) = x_b$; $\dot{q}$ is the derivative of q satisfying that $\dot{q} : [a, b] \rightarrow T_{q(t)}X$, $t \mapsto v = \dot{q}(t)$ with $T_{q(t)}X$ denotes the tangent space of X at $q(t)$; L is a real-valued function with continuous first partial derivatives: $L : [a, b] \times TX \rightarrow \mathbb{R}$, $(t, x, v) \mapsto L(t, x, v)$ with TX being the tangent bundle of X defined by $TX = \bigcup_{x \in X} \{x\} \times T_x X$. The E-L equation, then, is given by

$$L_x(t, q(t), \dot{q}(t)) - \frac{d}{dt} L_v(t, q(t), \dot{q}(t)) = 0, \quad (2.2)$$

where L_x and L_v denote the partial derivatives of L with respect to x and v respectively.

To determine the equations of motion for the system dynamics, we follow the following steps:

1. Determine the kinetic energy K and the potential energy P .

2. Compute the Lagrangian

$$L = K - P. \quad (2.3)$$

3. Compute $\frac{\partial L}{\partial \dot{q}}$.

4. Compute $\frac{\partial L}{\partial \dot{q}}$ and from it, $\frac{d}{dt} \frac{\partial L}{\partial \dot{q}}$. It is important that $\dot{q}$ be treated as a complete variable rather than a derivative.

5. Solve the revised E-L equation for the system with the generalized forces

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = Q_q, \quad (2.4)$$

where Q_q are the generalized forces and q are the generalized coordinates.

2.3 Modeling

The RDIP works as follows. The movement of the arm on the base, Link 1, is constrained to the $x-o-z$ plane and rotating around the y axis. The movements of the other two links are constrained to a vertical plane perpendicular to Link 1. Link 1 is driven by a DC motor, which generates a torque to control the system and is described in [8]. Here we will not discuss the servo system. Therefore, the control input of the RDIP is the torque applied to Link 1. The control objective is to maintain the pendulums Link 2 and 3 in the upright position with Link 1 in the origin position.

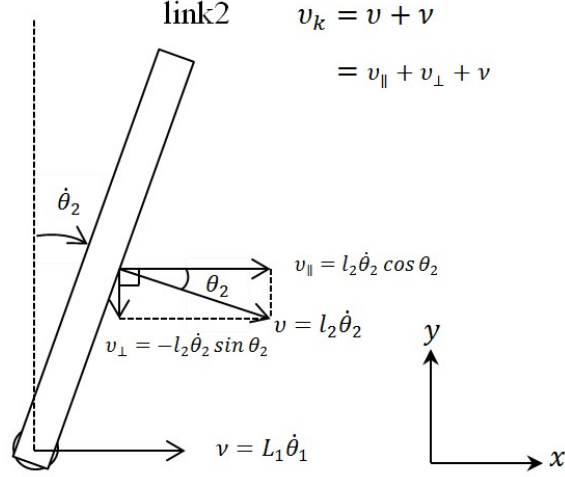


Figure 2.2: Velocity analysis for Link 2.

The total kinetic energy of each link in our system is given by the combination of its moving kinetic term K_m and its rotating kinetic term K_r as

$$\begin{aligned} K_m &= \frac{1}{2}mv^2, \\ K_r &= \frac{1}{2}J\dot{\theta}_i^2, \end{aligned} \quad (2.5)$$

where v and $\dot{\theta}_i$ are respectively the moving velocity and the rotational angular velocity. We provide the analysis of the total kinetic energy of the Link 2 in Figure 2.2 to help readers make an analysis for other two links with the same method. The potential energy is easy to get, thus we do not discuss it further. In general, we will have the total kinetic energy for the whole system as

$$\begin{aligned} K &= \frac{1}{2}J_1\dot{\theta}_1^2 + \frac{1}{2}J_2\dot{\theta}_2^2 + \frac{1}{2}J_3\dot{\theta}_3^2 + \frac{1}{2}m_2 \left[\left(L_1\dot{\theta}_1 + l_2\dot{\theta}_2 \cos \theta_2 \right)^2 + \left(-l_2\dot{\theta}_2 \sin \theta_2 \right)^2 \right] + \\ &\quad \frac{1}{2}m_3 \left[\left(L_1\dot{\theta}_1 + l_2\dot{\theta}_2 \cos \theta_2 + l_3\dot{\theta}_3 \cos \theta_3 \right)^2 + \left(-l_2\dot{\theta}_2 \sin \theta_2 - l_3\dot{\theta}_3 \sin \theta_3 \right)^2 \right], \end{aligned} \quad (2.6)$$

and the total potential energy

$$P = m_3gl_2 \cos \theta_2 + m_3g (L_2 \cos \theta_2 + l_3 \cos \theta_3). \quad (2.7)$$

Now we can obtain the Lagrangian by applying (2.6) and (2.7) to (2.3)

$$L = K - P.$$

Applying the E-L equation (2.4) to (2.3) results in three coupled non-linear equations.

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_1} - \frac{\partial L}{\partial \theta_1} = \tau - b_1 \dot{\theta}_1 \quad (2.8)$$

becomes

$$\begin{aligned} \tau = & (J_1 + L_1^2 (m_2 + m_3)) \ddot{\theta}_1 + L_1 (m_2 l_2 + m_3 L_2) \cos \theta_2 \ddot{\theta}_2 + L_1 m_3 l_3 \cos \theta_3 \ddot{\theta}_3 \\ & + b_1 \dot{\theta}_1 - L_1 (m_2 l_2 + m_3 L_2) \dot{\theta}_1^2 \sin \theta_2 - L_1 m_3 l_3 \dot{\theta}_3^2 \sin \theta_3. \end{aligned} \quad (2.9)$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_2} - \frac{\partial L}{\partial \theta_2} = -b_2 \dot{\theta}_2 \quad (2.10)$$

becomes

$$\begin{aligned} 0 = & -L_1 (m_2 l_2 + m_3 L_2) \cos \theta_2 \ddot{\theta}_1 - (J_2 + L_2^2 m_3 + l_2^2 m_2) \ddot{\theta}_2 - L_2 m_3 l_3 \cos (\theta_2 - \theta_3) \ddot{\theta}_3 \\ & - b_2 \dot{\theta}_2 - L_2 m_3 l_3 \dot{\theta}_3^2 \sin (\theta_2 - \theta_3) + (m_2 l_2 + m_3 L_2) g \sin \theta_2. \end{aligned} \quad (2.11)$$

And

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_3} - \frac{\partial L}{\partial \theta_3} = -b_3 \dot{\theta}_3 \quad (2.12)$$

becomes

$$\begin{aligned} 0 = & -L_1 m_3 l_3 \cos \theta_3 \ddot{\theta}_1 - L_2 m_3 l_3 \cos (\theta_2 - \theta_3) \ddot{\theta}_2 - (J_3 + l_3^2 m_3) \ddot{\theta}_3 \\ & - b_3 \dot{\theta}_3 + L_2 m_3 l_3 \dot{\theta}_2^2 \sin (\theta_2 - \theta_3) + m_3 l_3 g \sin \theta_3. \end{aligned} \quad (2.13)$$

If the equations are parameterized they reduce to a more manageable form. Define $h_1, h_2, h_3, h_4,$

h_5, h_6, h_7 and h_8 as

$$\begin{aligned} h_1 &= J_1 + L_1^2 (m_2 + m_3), \\ h_2 &= L_1 (m_2 l_2 + m_3 L_2), \\ h_3 &= L_1 m_3 l_3, \\ h_4 &= J_2 + L_2^2 m_3 + l_2^2 m_2, \\ h_5 &= L_2 m_3 l_3, \\ h_6 &= J_3 + l_3^2 m_3, \\ h_7 &= (m_2 l_2 + m_3 L_2) g, \\ h_8 &= m_3 l_3 g. \end{aligned} \quad (2.14)$$

The dynamic equations are reduced into the form

$$\begin{aligned}\tau &= h_1\ddot{\theta}_1 + h_2\cos\theta_2\ddot{\theta}_2 + h_3\cos\theta_3\ddot{\theta}_3 + b_1\dot{\theta}_1 - h_2\dot{\theta}_1^2\sin\theta_2 - h_3\dot{\theta}_3^2\sin\theta_3, \\ 0 &= -h_2\cos\theta_2\ddot{\theta}_1 - h_4\ddot{\theta}_2 - h_5\cos(\theta_2 - \theta_3)\ddot{\theta}_3 - b_2\dot{\theta}_2 - h_5\dot{\theta}_3^2\sin(\theta_2 - \theta_3) + h_7\sin\theta_2, \\ 0 &= -h_3\cos\theta_3\ddot{\theta}_1 - h_5\cos(\theta_2 - \theta_3)\ddot{\theta}_2 - h_6\ddot{\theta}_3 - b_3\dot{\theta}_3 + h_5\dot{\theta}_2^2\sin(\theta_2 - \theta_3) + h_8\sin\theta_3.\end{aligned}\quad (2.15)$$

To make the system dynamics more accessible, we assume that $\theta = [\theta_1, \theta_2, \theta_3]^\top$, $\dot{\theta} = [\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3]^\top$ and $\ddot{\theta} = [\ddot{\theta}_1, \ddot{\theta}_2, \ddot{\theta}_3]^\top$. Then (2.15) can be rewritten in the same form with the equations in [3] as

$$F(\theta)\ddot{\theta} + G(\theta, \dot{\theta}) + V(\theta) = ub(\theta), \quad (2.16)$$

where

$$F(\theta) = \begin{bmatrix} h_1 & h_2\cos\theta_2 & h_3\cos\theta_3 \\ -h_2\cos\theta_2 & -h_4 & -h_5\cos(\theta_2 - \theta_3) \\ -h_3\cos\theta_3 & -h_5\cos(\theta_2 - \theta_3) & -h_6 \end{bmatrix}, \quad (2.17)$$

$$G(\theta, \dot{\theta}) = \begin{bmatrix} b_1\dot{\theta}_1 - h_2\dot{\theta}_1^2\sin\theta_2 - h_3\dot{\theta}_3^2\sin\theta_3 \\ -b_2\dot{\theta}_2 - h_5\dot{\theta}_3^2\sin(\theta_2 - \theta_3) \\ -b_3\dot{\theta}_3 + h_5\dot{\theta}_2^2\sin(\theta_2 - \theta_3) \end{bmatrix}, \quad (2.18)$$

$$V(\theta) = \begin{bmatrix} 0 \\ h_7\sin\theta_2 \\ h_8\sin\theta_3 \end{bmatrix}, \quad (2.19)$$

$$b(\theta) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}. \quad (2.20)$$

From the mathematical model, we can get the conclusion about the natural characteristics of the RDIP similar with [9]:

- **Open-loop Instability:** As we have mentioned before, the upper equilibrium is an unstable equilibrium for the system. A little disturbance will lead the open-loop system to leave from the equilibrium and fall down to the downward position which is the stable equilibrium of the system. This characteristic can be seen by the MATLAB simulation result in Figure 5.1 when applied this model in some initial states without adding a control signal into the system.
- **Coupling Characteristic:** According to the mathematical model, we can see the strong coupling characteristic between the state variables of the pendulum. This can be observed by

developing the dynamics into a state differential equation form. We will prove this later in Section 3.2.

So far, a nonlinear mathematical model has been built for the RDIP. In the following two chapters, we will design controllers for the RDIP based on this model. In Chapter V, we will check the validation of this model by simulation and we will analyze its behavior in some initial states without control.

In attempting to further develop the mathematical model for the RDIP, several challenges present themselves. The principal ones are:

- The emergence of vibrational modes associated with unmodeled dynamics related to the elasticity of the structure.
- The model of the motor system and its incorporation with the RDIP model.

CHAPTER III

CONTROL USING LINEAR QUADRATIC REGULATOR

3.1 Introduction

Linear Quadratic Regulator (LQR) is one of the main results of the theory of optimal control which is concerned with operating a system at the minimum cost. In this theory, system dynamics are usually described by a set of linear differential equations and the cost in the control process is represented as a quadratic functional.

Suppose we have a dynamic process characterized by the vector-matrix differential equation

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

where x is the state variables, u is the control input, A and B are known matrices. The goal is to seek a feedback gain K which will be applied in the linear control law

$$u = -Kx, \quad (3.2)$$

so as to minimize the cost function V expressed as the integral of a quadratic form in the state x plus a second quadratic form in the control u

$$V = \frac{1}{2} \int_0^\infty \left(x^\top Q x + u^\top R u \right) dt, \quad (3.3)$$

where Q is a positive semi-definite symmetric matrix and R is a positive definite symmetric matrix. With this assumption, the first integral term $x^\top Q x$ is always positive or zero and the second term $u^\top R u$ is always positive at each time t for all values of x and u . This guarantees that V is well-defined. In terms of eigenvalues, the eigenvalues of Q should be non-negative and of R should be positive. Usually, Q and R are selected to be diagonal for convenience, thus some entries of Q will be positive with some possible zeros on its diagonal while all the entries of R must be positive. Note that R is invertible.

The cost function V is a performance index of the cost of the whole control process and it can be interpreted as an energy function. The magnitude of the control action itself is included in the cost function so as to keep the cost, which is due to the control action itself, to be limited. Since both the state x and the control input u are weighted in V , if V is small, then both x and u are kept to be small. Furthermore, if V is minimized, then it is certainly finite, and since it is an infinite integral of x , this implies that x goes to zero as t goes to infinity, which guarantees that the closed-loop system will be stable.

The plant is linear and the cost function V is quadratic. For this reason, the problem of determining the state feedback control which regulates the states to zero to minimize V is called the Linear Quadratic Regulator (LQR).

By solving the algebraic Riccati equation (ARE) for P

$$A^\top P + PA + Q - PBR^{-1}B^\top P = 0, \quad (3.4)$$

we obtain the optimal LQR gain K as

$$K = R^{-1}B^\top P. \quad (3.5)$$

The minimal value of the performance criterion V using this gain is given by

$$V(x_0) = x_0^\top P x_0, \quad (3.6)$$

which only depends on the initial condition x_0 . This means that the cost of using the LQR gain can be computed from the initial conditions before the control is ever applied to the system.

Note that Q and R are set by control engineers. In effect, the LQR algorithm takes care of the tedious work done by engineers in optimizing the controller. However, one still needs to specify the weighting factors Q and R and compare the results with the specified design goals. Often this means that it will be an iterative process for engineers to judge the produced "optimal" controllers through simulation and then adjust the weighting factors to get a controller more suitable with the specified design goals. A clear linkage between the adjusted matrices and the resulting changes in control behavior is hard to find, which limits the application of the LQR based controller synthesis.

The design procedure for finding the LQR feedback K is:

1. Select LQR design matrices Q and R .
2. Solve the ARE for P .

3. Find the LQR gain using $K = R^{-1}B^\top P$.

The MATLAB routine “lqr(A,B,Q,R)” is used here to perform the numerical procedure for solving the ARE.

3.2 Implementation

For the LQR control design, we need to linearize the RDIP dynamics at the upright equilibrium.

By defining x as

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}, \quad (3.7)$$

we can rewrite (2.15) as

$$\dot{x} = f(x) + g(x)u, \quad (3.8)$$

where $f(x) \in \mathbb{R}^6$ and $g(x) \in \mathbb{R}^6$ (Note that vectors in this paper are column vectors by default)

$$f(x) = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ f_4(x) \\ f_5(x) \\ f_6(x) \end{bmatrix}, \quad (3.9)$$

$$g(x) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ g_4(x) \\ g_5(x) \\ g_6(x) \end{bmatrix}. \quad (3.10)$$

We do not expand the terms $f_i(x)$ and $g_i(x)$ for $i = 4, 5, 6$ because their complete forms are too long to be expanded here. Readers can easily get these terms from our dynamics with the help of a computer. From the expansion form of (3.8), we find that all the terms $f_i(x)$ and $g_i(x)$ are related to the states from x_2 to x_6 . This indicates a strong coupling relationship between the three links which we mentioned in the previous chapter.

Then we linearize the system with the method in [10]

$$A = \frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} & \frac{\partial f_1}{\partial x_4} & \frac{\partial f_1}{\partial x_5} & \frac{\partial f_1}{\partial x_6} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \frac{\partial f_2}{\partial x_4} & \frac{\partial f_2}{\partial x_5} & \frac{\partial f_2}{\partial x_6} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} & \frac{\partial f_3}{\partial x_4} & \frac{\partial f_3}{\partial x_5} & \frac{\partial f_3}{\partial x_6} \\ \frac{\partial f_4}{\partial x_1} & \frac{\partial f_4}{\partial x_2} & \frac{\partial f_4}{\partial x_3} & \frac{\partial f_4}{\partial x_4} & \frac{\partial f_4}{\partial x_5} & \frac{\partial f_4}{\partial x_6} \\ \frac{\partial f_5}{\partial x_1} & \frac{\partial f_5}{\partial x_2} & \frac{\partial f_5}{\partial x_3} & \frac{\partial f_5}{\partial x_4} & \frac{\partial f_5}{\partial x_5} & \frac{\partial f_5}{\partial x_6} \\ \frac{\partial f_6}{\partial x_1} & \frac{\partial f_6}{\partial x_2} & \frac{\partial f_6}{\partial x_3} & \frac{\partial f_6}{\partial x_4} & \frac{\partial f_6}{\partial x_5} & \frac{\partial f_6}{\partial x_6} \end{bmatrix}, \quad (3.11)$$

$$B = \frac{\partial g}{\partial u} = \begin{bmatrix} \frac{\partial g_1}{\partial u} \\ \frac{\partial g_2}{\partial u} \\ \frac{\partial g_3}{\partial u} \\ \frac{\partial g_4}{\partial u} \\ \frac{\partial g_5}{\partial u} \\ \frac{\partial g_6}{\partial u} \end{bmatrix}. \quad (3.12)$$

Substituting the origin $x_0 = 0$ and $u = 0$ to A and B we will have the linearized system

$$A = \frac{1}{T} \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & h_3 h_5 h_7 - h_2 h_6 h_7 & h_2 h_5 h_8 - h_3 h_4 h_8 & b_1 h_4 h_6 - b_1 h_5^2 & b_2 h_2 h_6 - b_2 h_3 h_5 & b_3 h_3 h_4 - b_3 h_2 h_5 \\ 0 & h_7 h_3^2 - h_1 h_6 h_7 & h_1 h_5 h_8 - h_2 h_3 h_8 & b_1 h_2 h_6 - b_1 h_3 h_5 & b_2 h_1 h_6 - b_2 h_3^2 & b_3 h_3 h_4 - b_3 h_2 h_5 \\ 0 & h_1 h_5 h_7 - h_2 h_3 h_7 & h_8 h_2^2 - h_1 h_4 h_8 & b_1 h_3 h_4 - b_1 h_2 h_5 & b_2 h_2 h_3 - b_2 h_1 h_5 & b_3 h_1 h_4 - b_3 h_2^2 \end{bmatrix}, \quad (3.13)$$

$$B = \frac{1}{T} \begin{bmatrix} 0 \\ 0 \\ 0 \\ h_4 h_6 - h_5^2 \\ h_2 h_6 - h_3 h_5 \\ h_3 h_4 - h_2 h_5 \end{bmatrix}, \quad (3.14)$$

where

$$T = h_6 h_2^2 - 2 h_2 h_3 h_5 + h_4 h_3^2 + h_1 h_5^2 - h_1 h_4 h_6. \quad (3.15)$$

Another simple approach in [2] is also good for linearizing the system model by using the approximation

$$\begin{aligned}\sin(\theta_i) &= \theta_i, \\ \cos(\theta_i) &= 1,\end{aligned}\tag{3.16}$$

under the condition that $\theta_i \approx 0, i = 1, 2, 3$.

We will use this method to linearize our second system model expressed in the form of (2.16). Thus we have

$$\bar{F}(\theta)\ddot{\theta} + \bar{G}(\theta, \dot{\theta}) + \bar{V}(\theta) = u\bar{b}(\theta),\tag{3.17}$$

where,

$$\bar{F}(\theta) = \begin{bmatrix} h_1 & h_2 & h_3 \\ -h_2 & -h_4 & -h_5 \\ -h_3 & -h_5 & -h_6 \end{bmatrix},\tag{3.18}$$

$$G(\theta, \dot{\theta}) = \begin{bmatrix} b_1\dot{\theta}_1 \\ -b_2\dot{\theta}_2 \\ -b_3\dot{\theta}_3 \end{bmatrix},\tag{3.19}$$

$$V(\theta) = \begin{bmatrix} 0 \\ h_7\theta_2 \\ h_8\theta_3 \end{bmatrix},\tag{3.20}$$

$$b(\theta) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.\tag{3.21}$$

Here we ignore all the high-order terms of θ_i in both of these two methods. High-order terms contain at least quadratic quantities of θ_i . Since if θ_i are small, their squares are even smaller, the high-order terms can be neglected.

With the linear state matrices A and B , we are able to figure out the controllability of the system.

The controllability matrix can be computed with the help of MATLAB

$$C = [A \quad BA \quad B^2A \quad B^3A \quad B^4A \quad B^5A]$$

$$= \begin{bmatrix} 0 & -137.033 & 717.598 & -10067.0 & 157330.0 & -2565733.0 \\ 0 & -14.5856 & -18.2875 & 936.228 & -18645.9 & 318164.0 \\ 0 & -133.981 & 1752.88 & -30799.6 & 492875.0 & -8090099.0 \\ -137.033 & 717.598 & -10067.0 & 157330.0 & -2565733.0 & 41787900 \\ -14.5856 & -18.2875 & 936.228 & -18645.9 & 318164.0 & -5232977.0 \\ -133.981 & 1752.88 & -30799.6 & 492875.0 & -8090099.0 & 131834000 \end{bmatrix}. \quad (3.22)$$

The rank of the matrix C is 6, therefore the linearized system is controllable. We simply choose $Q = I^6$ and $R = I$ to obtain the first gain K that stabilizes the system by using the MATLAB command “lqr(A,B,Q,R)” and then adjust the entries of these two matrices to optimize the performance of the LQR controller in the following section.

3.3 LQR Tuning

As we discussed in section 3.1, Q and R are selected by the design engineer. Different choice of these design parameters will lead to different control performance for the closed-loop system. Generally speaking, a large Q means that, to keep V small, the state x must be smaller, resulting in the poles of the closed-loop system matrix $A_c = (A - BK)$ being further left in the s-plane so that the states converge faster to zero. On the other hand, selecting R large means that the control input u must be smaller to keep V small which implies a less control effort. In this case the poles are slower, resulting in larger values of the state x .

A reasonable simple choice for the matrices Q and R is given by the Bryson's rule [11]. Select Q and R diagonal with

$$Q_{ii} = \frac{1}{\text{maximum acceptable value of } x_i^2}, \quad i = 1, \dots, l, \quad (3.23)$$

$$Q_{jj} = \frac{1}{\text{maximum acceptable value of } u_j^2}, \quad j = 1, \dots, k. \quad (3.24)$$

In this way, the Bryson's rule scales the variables that appear in V so that the maximum acceptable value for each term is one. This is especially important when the units used for the different components of u and x make the values of the variables numerically very different from each other.

Although Bryson's rule usually gives good results, it is just the starting point to a trial-and-error iterative design procedure aimed to obtain a controller more in line with the desirable properties for the closed-loop system.

Applying different Q and R in different initial states to construct the LQR controller in the MATLAB simulation, the magnitudes of the three angular velocities are found to be around 3. The magnitudes of the three angles are around $\pi/3$. The magnitude of the control input is around 1.

Then we choose the following Q and R to start our tuning according to the Bryson's rule

$$Q = \begin{bmatrix} 0.9119 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.9119 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.9119 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1111 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1111 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1111 \end{bmatrix}, \quad (3.25)$$

$$R = 1,$$

and finally get an optimized result after some fine tuning as

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1667 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1667 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1667 \end{bmatrix}, \quad (3.26)$$

$$R = 0.8.$$

In fact, the entries of Q and R can be adjusted separately and it is possible that it will provide an even better result. Now we only use two parameters q_1 and q_2 to adjust all the entries, since we find that the changes of the angles are similar in magnitude and the same case happens to the angular velocities. In addition, it is more convenient to show the procedure about how to adjust them and compare the tuning results of different combinations. Actually, we find that this setting has been good enough to show us a distinct improvement effect.

The tuning effect is shown in the comparison of the control results between the LQR with the identity weight matrices and the tuned LQR in Section 5.2. We can see that tuning makes an improvement to the RDIP control system. However, the effect of this optimization method is limited. For instance, in the initial states $[0 \ -0.1 \ 0.1 \ 0 \ 0 \ 0]^\top$ which we will discuss in Chapter IV and V in details, it is impossible to control the system with the LQR no matter how we tune the weight matrix. Therefore, we need to choose another way to optimize the LQR. The DAFC scheme is taken for this purpose.

3.4 Stability Analysis

Although it is possible to optimize the LQR controller by adjusting the matrices Q and R on the principle of the Bryson's rules, we still have no idea about the stability of the RDIP system near the

upright position. In fact, the stability plays a central role in the analysis of a given control method for systems. Here we will try to figure out the Region of Attraction (R.O.A) of the closed-loop RDIP system to analyze the system stability (we will define the notion of the R.O.A later in the introduction of the Lyapunov stability in this section). Once the R.O.A of the system is accessible, a standard is able to be established to compare the performances of different control methods. A method providing a larger R.O.A makes the system stable in a larger scale of initial states. We will check the stability near the upright position, one of the 2 equilibrium point. An equilibrium point is stable if all solutions starting at the nearby points stay nearby; otherwise it is unstable. It is asymptotically stable if all solutions starting at nearby points not only stay nearby, but also tend to the equilibrium point as time approaches infinity. In the case of the RDIP, the downward equilibrium is an asymptotically stable equilibrium while the upper equilibrium is unstable. Stability of equilibrium points is usually characterized in the sense of the Lyapunov function. Lyapunov's method helps to prove stability without requiring knowledge of the true physical energy, provided a Lyapunov function can be found to satisfy the following constraints. One thing we need mentioned here is that it is common that the R.O.A might not be the only concern when we decide which control method we will select for a given system. For example, we may need a quick-response controller that can force the system to converge to the origin as fast as possible. Also, there is a limit for the Lyapunov stability analysis. That is, we cannot estimate directly the performance of a controller which updates itself online, such as the DAFC method we will take in the next chapter.

Here the Lyapunov's second method for stability is used. For more details about this method readers can refer to [2]. Suppose we have a system with a point of equilibrium at $x = 0$. A Lyapunov candidate function $V(x)$ is a continuously differentiable function defined in a domain $D \subset \mathbb{R}^n \rightarrow \mathbb{R}$

that contains the origin such that

$$V(0) = 0 \text{ and } V(x) > 0 \text{ in } D - \{0\}. \quad (3.27)$$

Then the system is stable in the sense of Lyapunov if

$$\dot{V}(x) = \frac{dV(x)}{dt} \leq 0 \text{ in } D. \quad (3.28)$$

The trajectory of the system states can be constrained inside an area that it can never escape away from.

Moreover, if

$$\dot{V}(x) = \frac{dV(x)}{dt} < 0 \text{ in } D, \quad (3.29)$$

the system is asymptotically stable since the trajectory of the states approaches the origin as time progresses. An additional condition called “properness” or “radial unboundedness” is required in order to conclude global asymptotic stability. In general, the origin is stable if there is a continuously differentiable positive definite function $V(x)$ so that $\dot{V}(x)$ is negative semidefinite, and it is asymptotically stable if $\dot{V}(x)$ is negative definite.

To understand this method of stability analysis, we can visualize the Lyapunov function as the energy of a physical system. If there is no energy restored into it, the system will lose energy (due to vibration, friction or some factors else) over time and finally stop in some final resting state. This final state is called the attractor. For a system to be controlled there might be a lot of Lyapunov functions that can be applied but finding an appropriate Lyapunov function to support the stability analysis of a system is difficult.

Usually we will use a class of scalar functions of the quadratic form for which sign definiteness can be easily checked

$$V(x) = x^\top P x = \sum_{i=1}^n \sum_{j=1}^n p_{ij} x_i x_j, \quad (3.30)$$

where P is a real positive definite symmetric matrix in which case $V(x)$ is guaranteed to be a good Lyapunov candidate. Here we re-take the Q matrix that we picked up for the LQR control to construct the Lyapunov candidate function for our system

$$V(x) = x^\top Q x, \quad (3.31)$$

where $x = [x_1, x_2, x_3, x_4, x_5, x_6]^\top$.

Therefore we can easily get

$$\dot{V}(x) = x^\top Q \dot{x} + \dot{x}^\top Q x, \quad (3.32)$$

where $\dot{x} = [\dot{x}_1, \dot{x}_2, \dot{x}_3, \dot{x}_4, \dot{x}_5, \dot{x}_6]^\top$ and Q is already selected as a positive definite diagonal matrix.

Substituting $u = -Kx$ into (3.8) we have

$$\dot{x} = f(x) - g(x)Kx, \quad (3.33)$$

and then

$$\dot{V}(x) = x^\top Q [f(x) - g(x)Kx] + [f(x) - g(x)Kx]^\top Q x. \quad (3.34)$$

To this point, we are able to find the R.O.A of the closed-loop system with the LQR controller using MATLAB. The procedure is:

1. Sample a sufficiently large scale around the origin at small intervals.

2. Estimate the value of $\dot{V}(x)$ for these samples according to (3.34).

3. Pick up the nearest point of the sample set, x_p , such that

$$\dot{V}(x_p) \geq 0. \quad (3.35)$$

4. Enlarge the sample set if there is no such a point exists and repeat Step 1 to Step 3 till a point x_p is found.

5. The R.O.A can be estimated as

$$R.O.A \approx ||x_p||. \quad (3.36)$$

Note the approximation sign is used here because the R.O.A should be a region without x_p . Since x_p is the nearest point satisfying (3.35), it is a point on the boundary between the R.O.A and the area outside. Any point that has a less distance than x_p will make $\dot{V}(x)$ less than 0. If there is another x'_p for which the value of $\dot{V}(x'_p)$ is semi-positive within the boundary, then the R.O.A should be smaller than the given one. In this way, we can make sure that the statement stands. To make the identification more precise, we could use smaller intervals to sample the testing points. Another thing we need to mention here is that, we use the Euclidean norm to express the distance between a point and the origin. Therefore,

$$\begin{aligned} ||x|| &= ||x||_2 \\ &= \sqrt{x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2}. \end{aligned} \quad (3.37)$$

$q_1 = \pi$		R						
		100	10	5	4	3	2	1
q_2	1	0.0361	0.0490	0.0447	0.0436	0.0447	0.0480	0.0458
	10	0.0387	0.0490	0.0566	0.0648	0.0632	0.0616	0.0592
	100	0.0387	0.0490	0.0566	0.0648	0.0632	0.0616	0.0592
	1000	0.0387	0.0490	0.0566	0.0648	0.0632	0.0616	0.0592

Table 3.1: R.O.A searching result with $q_1 = \pi$.

q_1	R.O.A	q_2	R
π	0.0648	10	4
$\pi/2$	0.0721	10	1
$\pi/4$	0.0755	10	5
$\pi/8$	0.0700	10	10

Table 3.2: Largest R.O.A for different q_1 .

For the convenience of the simulation, we simply set the matrix Q the same as in Section 3.2

$$Q = \begin{bmatrix} q_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & q_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & q_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & q_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & q_2 \end{bmatrix}, \quad (3.38)$$

where q_1 and q_2 are selected for the angles and the angular velocities of the links respectively. Note that we only adjust two parameters q_1 and q_2 based on the same reason as in Section 3.2 and actually the six states can be adjusted separately. Table 3.1 shows us one of the searching results in simulation. Here, we fix q_1 as π , and change the values of q_2 and R with some of their representative values. Thus we can find that when $q_2 = 10$ and $R = 4$ the system get its largest R.O.A with the fixed q_1 . Table 3.2 shows that the largest R.O.A of the system with different q_1 . For each option of q_1 , not only the largest R.O.A but also its corresponding q_2 and R values are provided in this table.

From Table 3.2, we can see that when $q_1 = \pi/4$, $q_2 = 10$ and $R = 5$, the LQR gets its largest R.O.A as 0.0755 which means that if $\|x\| < 0.0755$, the derivative of the Lyapunov function is negative and therefore system is asymptotically stable. Beyond the R.O.A, the behavior of the RDIP system is not able to be predicted. There are still some points starting from which the states of the system will converge to the origin in the control process whereas the progress is not controllable. Interestingly, although the R.O.A is very small, the LQR can work in a region even beyond it. This might be possible that there exist other Lyapunov candidates can provide a better assessment about the R.O.A than the Lyapunov function we defined here. We will leave this topic for the RDIP system with LQR in the study in future.

CHAPTER IV

ADAPTIVE CONTROL

4.1 Introduction

Since LQR can only work in a small region, we intend to optimize the performance of the LQR controller. We will develop the LQR controller into a Direct Adaptive Fuzzy Control (DAFC) system as mentioned in [6] and [12]. From the conclusion of [9], we know that a fuzzy controller can be used to control the RDIP. A theoretical analysis of the stability and design of a fuzzy control system is introduced in [13] using the Takagi-Sugeno (T-S) fuzzy model. However, there are some problems about it. While this non-adaptive fuzzy control has proven its value in the application, it is difficult to specify the rule base for some plants, or need could arise to tune the rule base parameters if the plant changes. In the RDIP system, it is very hard to gather the heuristic knowledge about how to control the RDIP to make it stands upright. Since heuristics do not provide enough information to specify all the parameters of the fuzzy controller, a priori, adaptive schemes that use data gathered during the on-line operation of the controller can be used to improve the fuzzy system by making it automatically learn the parameters, to ensure that the performance objectives are met.

There has been some adaptive control schemes applied in system control. As the first adaptive fuzzy controller, the linguistic self-organizing controller is introduced in [14]. Another successful method, so-called “fuzzy model reference learning controller” is introduced in [15]-[16]. However, the problem with them is that while they appear to be practical heuristic approaches to adaptive fuzzy control there is no proof that these methods will result in a stable closed-loop system. Here, we are going to optimize the LQR controller by the DAFC which has been provided with the stability analysis in [6] and the experimental validation in [12]. Therefore the stability requirement could be met for a safety-critical system such as the RDIP experiment plant in the lab. We will make a detailed introduction to this control scheme in Section 4.3.

The DAFC attempts to directly adjust the parameters of a fuzzy or neural controller to achieve asymptotic tracking of a reference input. There are some advantages with the DAFC:

- The stability of this controller may be applied to systems with a state-dependent input gain, such as systems with the LQR gain.
- The DAFC method works for zero dynamics with minimum phase, however it looks like it also works for some of the zero dynamics with non-minimum phase.
- The direct adaptive controller allows for T-S fuzzy systems, standard fuzzy systems, or neural networks.

- The direct adaptive technique presented here allows for the inclusion of a known controller u_k so that it may be used to either enhance the performance of some pre-specified controller or stand alone as a stable adaptive controller.

For what follows in this chapter, the notation from [12] will be used. In the next section, we will introduce the T-S fuzzy system first. Reader could consult [17] and [18] to fully understand this kind of fuzzy system. Then we provide a description of the DAFC and specify the DAFC scheme for the RDIP with the LQR controller which we have obtained as the “known part” of the controller.

4.2 Takagi-Sugeno Fuzzy System

This section largely follows [6] to provide an introduction of the Takagi-Sugeno (T-S) fuzzy system. Readers can refer to [19] for more details about the T-S fuzzy system. To briefly present the notation, take a fuzzy systems denoted by $\tilde{f}(x)$. Then, $\tilde{f}(x) = \sum_{i=1}^p c_i \mu_i / \sum_{i=1}^p \mu_i$. Here, singleton fuzzification of the input $x = [x_1 \ x_2 \ \dots \ x_n]^\top$ is assumed; the fuzzy system has p rules, and μ_i is the value of the membership function for the antecedent of the i th rule given the input x . It is assumed that the fuzzy system is constructed in such a way that $\sum_{i=1}^p \mu_i \neq 0$ for all $x \in \mathbb{R}^n$. The parameter c_i is the consequent of the i th rule which, in this paper, will be taken as a linear combination of Lipschitz continuous functions $z_k(x) \in \mathbb{R}, k = 1, \dots, m - 1$, so that

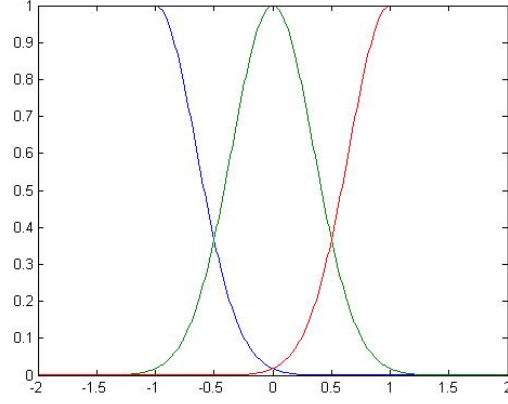


Figure 4.1: Membership function with $c = 1$ and $\sigma = 0.25$.

$c_i = a_{i,0} + a_{i,1}z_1(x) + \dots + a_{i,m-1}z_{m-1}(x), i = 1, \dots, p$. Define

$$\begin{aligned}
 z &= \begin{bmatrix} 1 \\ z_1(x) \\ \vdots \\ z_{m-1}(x) \end{bmatrix}, \\
 \zeta^\top &= \frac{[\mu_1 \dots \mu_p]}{\sum_{i=1}^p \mu_i}, \\
 A^\top &= \begin{bmatrix} a_{1,0} & a_{1,1} & \dots & a_{1,m-1} \\ a_{2,0} & a_{2,1} & \dots & a_{2,m-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{p,0} & a_{p,1} & \dots & a_{p,m-1} \end{bmatrix}.
 \end{aligned} \tag{4.1}$$

Then, the nonlinear equation that describes the fuzzy system can be written as $\tilde{f}(x) = z^\top A \zeta$. Here we present one of the membership functions we use in the DAFC design for our system in Figure 4.1.

4.3 Theory

A DAFC controller directly adjusts the parameters of a controller to meet some performance specifications. In [12], the author developed the adaptive control method by assuming that $0 < \beta_0 \leq \beta(x) \leq \beta_1 < \infty$. For the RDIP, the assumption holds for $-\infty < \beta_1 \leq \beta(x) \leq \beta_0 < 0$ which can be found out from the simulation result of the RDIP dynamics in MATLAB. Here we will re-develop the direct adaptive scheme for our case and at the same time provide readers a description of the method.

Suppose we have the dynamics of the plant as

$$\begin{aligned}\dot{x} &= f(x) + g(x)u, \\ y &= h(x),\end{aligned}\tag{4.2}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}$ is the input, $y \in \mathbb{R}$ is the output of the plant, functions $f(x), g(x) \in \mathbb{R}^n$, and $h(x) \in \mathbb{R}$ are smooth. The system has a “strong relative degree” r as

$$\begin{aligned}\dot{y} &= L_f h(x), \\ \ddot{y} &= L_f^2 h(x), \\ &\vdots \\ y^{(r)} &= L_f^r h(x) + L_g L_f^{(r-1)} h(x)u,\end{aligned}\tag{4.3}$$

where $L_f^r h(x)$ is the r th Lie derivative of $h(x)$ with respect to f and $L_g^r h(x)$ is the r th Lie derivative of $h(x)$ with respect to g

$$\begin{aligned}L_f h(x) &= \frac{\partial h}{\partial x} f(x), \\ L_g h(x) &= \frac{\partial h}{\partial x} g(x), \\ L_f^{(2)} &= L_f[L_f h(x)],\end{aligned}\tag{4.4}$$

and so on. Then we have

$$y^{(r)} = \alpha(x) + \alpha_k(t) + [\beta(x) + \beta_k(t)]u.\tag{4.5}$$

We assume that $\alpha_k(t)$ and $\beta_k(t)$ are known components of the dynamics of the plant (that may depend on the states) or known exogenous time dependent signals and that $\alpha(t)$ and $\beta(t)$ represent nonlinear dynamics of the plant that are unknown. It is also assumed that if x is a bounded state vector, then $\alpha_k(t)$ and $\beta_k(t)$ are bounded signals. Throughout the analysis to follow, both $\alpha_k(t)$ and $\beta_k(t)$ may be set to zero for all $t \geq 0$.

We have some plant assumptions as follows.

1. The plant is of relative degree $1 \leq r < n$ with the zero dynamics exponentially attractive and there exists β_0 and β_1 such that $-\infty < \beta_1 \leq \beta(x) \leq \beta_0 < 0$.
2. $x_1, \dots, x_n$ and $y, \dots, y^{(r-1)}$ are measurable. Furthermore, by the use of Lipschitz properties of $\psi(\xi, \pi)$, the plants satisfying this assumption have bounded states [20].
3. We require that $\beta_k(t) = 0, t \geq 0$ and some function $B(x) \geq 0$ such that $|\dot{\beta}(x)| = |(\partial\beta/\partial x)\dot{x}| \leq B(x)$.
4. There exists some $\alpha_1(x) \geq |\alpha(x)|$.

Although Assumption 1 is not met in our case since our system has an undetermined zero dynamics, we will derive the control scheme for our system despite this condition. From [6] we know that this method works for some simple nonlinear systems that are non-minimum-phase such as the rotational single inverted pendulum. We are trying to implement this method in a more complicated case and verify if it will still work. Assumption 1 also introduces a requirement that the controller gain $\beta(x)$

be bounded by a constant β_0 from above and a constant β_1 from below. The third restriction requires that $|\dot{\beta}(x)| \leq B(x)$ for some $B(x) > 0$. If $\|\partial\beta/\partial x\|$ and $\|\dot{x}\|$ are bounded, then some $B(x)$ may be found. If the controller gain of the system is finite, $\|(\partial\beta/\partial x)\|$ is bounded. If $y^{(i)}$ is bounded as $i = 0, \dots, r$, then plants with no zero dynamics are ensured that $\|\dot{x}\|$ is bounded since the states can be represented in terms of outputs $y^{(i)}$. For a plant has zero dynamics, if $\beta(x)$ is not dependent upon the zero dynamics, then once again we have $|\dot{\beta}(x)|$ bounded. In [6], a function of x is found as $\alpha_1(x)$ to meet Assumption 4. In this paper, we will use a constant as our global bound for $\alpha(x)$ to simplify the choice of this function. The constant is obtained by the observation in the RDIP system simulation results.

Using feedback linearization theory in [2], we assume that there exists some ideal controller

$$u^* = \frac{1}{\beta(x)} [-\alpha(x) + \nu(t)], \quad (4.6)$$

where $\nu(t)$ is a free parameter. We may express u^* in terms of T-S fuzzy model, so that

$$u^* = z_u^\top A_u^* \zeta_u + u_k + d_u(x), \quad (4.7)$$

where $z_u \in \mathbb{R}^{m_u}$, $\zeta_u \in \mathbb{R}^{p_u}$ and $A_u^* \in \mathbb{R}^{m_u \times p_u}$ is the ideal direct control parameters

$$A_u^* := \arg \min_{A_u \in \Omega_u} \left[\sup_{X \in S_x, \nu \in S_m} |z_u^\top A_u \zeta_u - (u^* - u_k)| \right]. \quad (4.8)$$

$d_u(x)$ is an approximation error which arises when u^* is represented by a fuzzy system. We assume that $D_u(x) \leq |d_u(x)|$, where $D_u(x)$ is a known bound on the error in representing the ideal controller with a fuzzy system. If $|d_u(x)|$ is to be small, then our fuzzy controller will require x and ν to be available, either through the input membership function or through $z_u^\top$. u_k is a known part of the controller. The DAFC attempts to directly determine a controller, so within this chapter we allow for a known part of the controller that is perhaps specified via heuristics or past experience

with the application of conventional direct control (in our case, LQR). The approximation of the desired control is

$$\hat{u} = z_u^\top A_u \zeta_u + u_k, \quad (4.9)$$

where the matrix A_u is updated online. The parameter error matrix

$$\phi_u(t) = A_u(t) - A_u^* \quad (4.10)$$

is used to define the difference between the parameters of the current controller and the desired controller. The control law is given by

$$u = \hat{u} + u_{sd} + u_{bd}. \quad (4.11)$$

In general, the DAFC is comprised of a bounding control term u_{bd} , a sliding-mode control term u_{sd} , and an adaptive control term $\hat{u}$. Here we define $\nu = y_m^{(r)} + \eta e_s + \bar{e}_s - a_k(t)$ with e_0 , e_s and $\bar{e}_s$ as defined

$$\begin{aligned} e_0 &= y_m - y_p, \\ e_s &= [e_0 \dots e_0^{(r-1)}][k_0 \dots k_{r-2}, 1]^\top, \\ \bar{e}_s &= \dot{e}_s - e_0^{(r)}, \end{aligned} \quad (4.12)$$

where $\hat{L}(s) = s^{r-1} + k_{r-2}s^{r-2} + \dots + k_1s + k_0$ has its poles in the open left-half plane.

Combining the above equations we can have

$$\dot{e}_s + \eta e_s = -\beta(x)(\hat{u} - u^*) - \beta(x)(u_{sd} + u_{bd}). \quad (4.13)$$

4.3.1 Bounding Control

We now define the bounding control term u_{bd} of the DAFC. The bounding control term is determined by considering

$$u_{bd} = \frac{1}{2}e_s^2. \quad (4.14)$$

We differentiate (4.14) and use (4.13) to obtain

$$\begin{aligned} \dot{v}_{bd} &= -\eta e_s^2 - e_s [\beta(x)(\hat{u} - u^*) + \beta(x)(u_{sd} + u_{bd})] \\ &\leq -\eta e_s^2 - |e_s| [\beta(x)(|\hat{u}| + |u^*|) + \beta(x)|u_{sd}|] - \beta(x)u_{bd}e_s. \end{aligned} \quad (4.15)$$

We do not explicitly know u^* , however, the bounding controller can be implemented using $\alpha_1(x) \geq |\alpha(x)|$ as

$$u_{bd} = \begin{cases} \left(-|\hat{u}| - |u_{sd}| + \frac{\alpha_1(x) + |\nu|}{\beta_0}\right) \text{sgn}(e_s) & \text{if } |e_s| > M_e, \\ 0 & \text{if } else. \end{cases} \quad (4.16)$$

Using (4.15) and (4.16), we have

$$\dot{v}_{bd} \leq -\eta e_s^2, \text{ if } |e_s| \geq M_e. \quad (4.17)$$

Thus we are ensured that if there exists a time t_0 such that $|e_s(t_0)| > M_e$, then for $t > t_0$, $|e_s(t)|$ will decrease exponentially until $|e_s(t)| \leq M_e$.

4.3.2 Adaptation Algorithm

Consider the following Lyapunov candidate equation

$$V_d = -\frac{1}{2\beta(x)}e_s^2 + \frac{1}{2}\text{tr}(\phi_u^\top Q_u \phi_u), \quad (4.18)$$

where $Q_u \in \mathbb{R}^{m_u \times m_u}$ is positive definite and diagonal, and $\phi_u = A_u - A_u^*$. Since $-\infty < \beta_1 \leq \beta(X) \leq \beta_0 < 0$, V_d is radially unbounded. The $\text{tr}(\cdot)$ is the trace operator. The Lyapunov candidate

V_d is used to describe the error in tracking and the error between the desired controller and current controller. If $V_d \rightarrow 0$, then both the tracking and learning objectives have been fulfilled. Taking the derivative of (4.18) yields

$$\dot{V}_d = -\frac{e_s}{\beta(x)}[\dot{e}_s] + \text{tr}(\phi_u^\top Q_u \dot{\phi}_u) + \frac{\dot{\beta}(x)e_s^2}{2\beta^2(x)}. \quad (4.19)$$

Substituting $\dot{e}_s$, as defined in (4.13), yields

$$\dot{V}_d = -\frac{e_s}{\beta(x)}[-\eta e_s - \beta(x)(\hat{u} - u^*) - \beta(x)(u_{sd} + u_{bd})] + \text{tr}(\phi_u^\top Q_u \dot{\phi}_u) + \frac{\dot{\beta}(x)e_s^2}{2\beta^2(x)}. \quad (4.20)$$

Using the following fuzzy controller update law

$$\dot{A}_u(t) = -Q_u^{-1} z_u \zeta_u^\top e_s, \quad (4.21)$$

since $\dot{\phi} = \dot{A}_u$, we have

$$\dot{V}_d = \frac{\eta}{\beta(x)}e_s^2 + [z_u^\top \phi_u \zeta_u - d_u + u_{sd} + u_{bd}]e_s - \text{tr}(z_u^\top \phi_u \zeta_u)e_s + \frac{\dot{\beta}(x)e_s^2}{2\beta^2(x)}. \quad (4.22)$$

The projection algorithm mentioned in [12] is used to ensure that Thus we have

$$\dot{V}_d \leq \frac{\eta}{\beta(x)}e_s^2 + [z_u^\top \phi_u \zeta_u - d_u + u_{sd} + u_{bd}]e_s - \text{tr}(z_u^\top \phi_u \zeta_u)e_s + \frac{\dot{\beta}(x)e_s^2}{2\beta^2(x)}. \quad (4.23)$$

The inequality above may equivalently be expressed as

$$\dot{V}_d \leq \frac{\eta}{\beta(x)}e_s^2 + \left[\frac{\dot{\beta}(x)e_s}{2\beta^2(x)} - d_u \right] e_s + (u_{sd} + u_{bd})e_s. \quad (4.24)$$

4.3.3 Sliding-mode Control

With u_{bd} defined before, we have

$$\begin{aligned} \dot{V}_d &\leq \frac{\eta}{\beta(x)}e_s^2 + \left[\frac{\dot{\beta}(x)e_s}{2\beta^2(x)} - d_u \right] e_s + u_{sd}e_s \\ &\leq \frac{\eta}{\beta_1}e_s^2 + \left[\frac{|\dot{\beta}(x)||e_s|}{2\beta^2(x)} + |d_u| \right] e_s + u_{sd}e_s. \end{aligned} \quad (4.25)$$

We now define the sliding-mode control term for the direct adaptive controller as

$$u_{sd} = - \left(\frac{B(x)|e_s|}{2\beta_0^2} + D_u(x) \right) \text{sgn}(e_s), \quad (4.26)$$

which ensures that $\dot{V} \leq \eta e_s^2 / \beta_1$.

4.4 Implementation

Although the theoretical analysis in [12] uses the assumption that the unknown control law u^* which the DAFC tries to identify is a feedback linearizing law, it was found experimentally in [6] that it is not necessarily the case. If the adaptation mechanism is initialized appropriately in accord with the known controller such as the LQR, the adaptation algorithm will converge to a controller that might behave in a very different manner because this mechanism seems to try to find the local optimum controller closest to its starting point in the search space and, in our case, an optimized LQR controller is found.

This finding is very important in the case that the control design involves dealing with a non-minimum-phase plant or a system with internal dynamics that are hard to identify. If a non-adaptive controller is available that can control the system regardless of whether the system is minimum-phase, then it is possible that the desirable boundedness characteristics of this controller can be incorporated into the DAFC design, and enhance by the robustness that the adaptive method provides.

In our case, we take the angle of Link 3 as the output

$$y = x_3 \quad (4.27)$$

Then we have

$$\begin{aligned}\dot{y} &= x_6, \\ \ddot{y} &= f_6(x) + g_6(x)u.\end{aligned}\tag{4.28}$$

To this point, we find that the zero dynamics of the system are very hard to identify. But we already know that the LQR controller works for the RDIP system, therefore a DAFC controller is possible to be implemented with the LQR controller.

First we present the conditions mentioned in the former section for the DAFC. we set the known bound for the approximation error $D_u(x)$ as 0.01. In practice it is often hard to have a concrete idea about the magnitude of $D_u(x)$, because the relation between u^* and its fuzzy representation might be difficult to characterize; however, it is much easier to begin with a rough, intuitive idea about this bound, and then iterate the design process and adjust it, until the performance of the controller indicates that one is close to the right value. For the simulation, we found that $D_u(x) = 0.01$ gives us good results. A small $D_u(x)$ indicates that the fuzzy system could represent the ideal controller very accurately.

We are going to search for u^* using (4.9) where $\zeta_u \in \mathbb{R}^{2187}$, with the membership functions shown in Figure 4.1. The mathematical description of the membership functions are provided in the section 5.3. We choose the number of rules $p = 3^7 = 2187$. And the matrix $A_u(t) \in \mathbb{R}^{7 \times 2187}$ is adaptively updated on line. The function vector z is taken as

$$z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \\ z_7 \end{bmatrix} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}.\tag{4.29}$$

The fuzzy system uses 3^7 rules and each $c_i(x)$ is a row of the matrix $z^\top A_u(t)$. We initialize the fuzzy system approximation by letting $A_u = 0$ since we know nothing about the optimal controller.

The DAFC control law is given by $u_d = \hat{u} + u_{sd} + u_{bd}$ as we have discussed before. The sliding term is given by (4.26). In simulation we find that $\beta(x)$ is between -128 and -135 , thus we choose β_0 as -100 . We also choose $B(x)$ as 250 for safety. The bounding term needs the assumption that $\alpha(x)$ is bounded, with $|\alpha(x)| \leq \alpha_1(x)$. We find that $\alpha(x)$ is always less than 14.6 , therefore we safely choose $\alpha_1(x)$ as 20 . Then we have u_{bd} as defined in (4.16). For simulation we use $M_e=8$, because by some calculation from the simulation results, we can find that M_e is always less than 4.6 when the system works. Actually, the parameter M_e defines a bounded, closed subset of the e_s error-state space within which the error is guaranteed to stay. ν is defined as in the section above. Here we select $\eta = 1$ and $k_0 = 5$ for the $e_s = k_0 e_0 + \dot{e}_0$. With this choice, the poles of the error transfer function are at $s = -1$ and $s = -5$, which produce a small error settling time.

The last part of the DAFC mechanism is the adaptation law, which is chosen in such a way that the output error converges asymptotically to zero, and the parameter error remains at least bounded. For this law We choose $Q_u = 1.2I^7$ with which the algorithm is able to adapt and estimate the control law $\hat{u}$ fast enough to perform well and compensate for disturbances, but without inducing oscillations typical of a too high adaptation rate.

CHAPTER V

SIMULATION RESULTS

5.1 Open-loop Model Verification

We will use the MATLAB programming engineering environment to do the simulation all through this paper. The solver “ode45” in MATLAB is used here in all cases to solve initial value problems for ordinary differential equations. It is important to notice that both the LQR and DAFC controllers are continuous time techniques, to implement them we use a digital computer, and thus are forced to implicitly use a discrete time approximation of the controller. It is reasonable to think that a proof of stability is still applicable when a continuous time technique is discretized, but such a study is outside the scope of the present work.

The initial conditions $x(0)$ are set to be $x(0) = [1 \ 0.1 \ 0.1 \ 0 \ 0 \ 0]^T$ which means the pendulums are nearly in the upright position at the starting point. Note here the initial states are expressed in *rad* or *rad/s* while we will show the states in all the figures in *deg* and *deg/s* for easy observation. And the performance of the RDIP model without control is shown below in Figure 5.1. All the

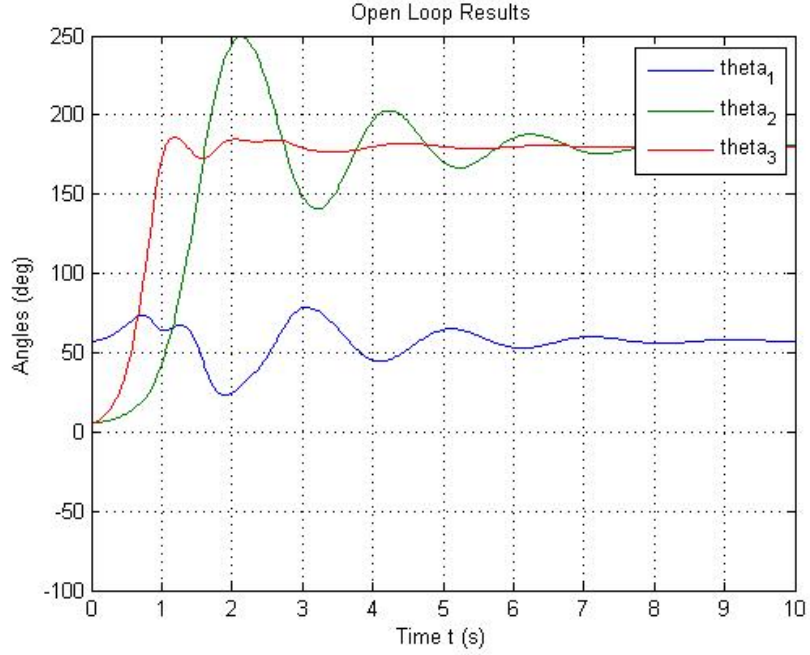


Figure 5.1: Open-loop simulation in the initial states $\theta_1 = 57.2958^\circ$, $\theta_2 = 5.7296^\circ$ and $\theta_3 = 5.7296^\circ$.

simulation results of the link angles of the RDIP will scale from -100° to 250° for the convenience to compare. The simulation time is from 0s to 10s.

As we can see from Figure 5.1, when there is no control input into the system, the pendulums fall down directly to its stable equilibrium in the downward position, where the RDIP keeps in its lowest total energy state. We can also see that the angle of Link 1 converges to a position near the initial states. Since there is no input torque and only the small viscous friction working on the base arm, the base arm can be seen as a conservative system along the horizontal plane. Therefore the base arm finally goes back to its starting point. As of now, the model we build seems acceptable to simulate the behavior of the RDIP system. During the simulation, a 3-D dynamical model is also

built to provide a direct insight of what the pendulum is doing by using the Euclidean geometry to calculate the relative position from the pivot to the end for each link.

5.2 Linear Quadratic Regulator Results

In MATLAB, the LQR gain can be computed directly once we select the values of the parameters Q and R . The MATLAB command “LQR” can help us speed up the calculation. We will set the parameters of LQR initially as

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (5.1)$$

$$R = 1.$$

With these settings, we have the LQR gain

$$K = [-1.0000 \quad 211.7112 \quad -120.5868 \quad -2.2287 \quad 56.0199 \quad -5.3027]. \quad (5.2)$$

The same initial states as in Figure 5.1 is used. The simulation result of the link angles is shown in Figure 5.2. The control input is shown together with the one of the tuning system in Figure 5.4. We can see that LQR provides a very good performance to control the system. All the links converge to zero and the control input is relatively small compared with the control signal we will get later from the DAFC simulation. That means the LQR controller will save the energy used to control the system. It is at nearly 6s that the system succeed in converging to the origin.

Then we will try to adjust the parameters of the LQR for the purpose of optimizing the control performance. Taking many simulation trails in different initial states with different Q and R , we get an intuitive understanding about the scale of the system states. The maximum of the absolute value

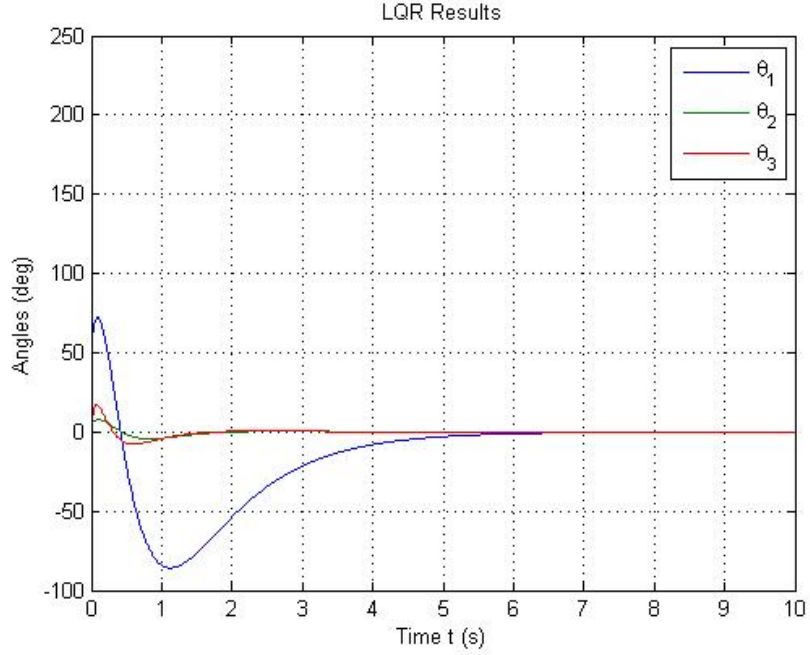


Figure 5.2: LQR simulation result with $Q = I^6$ and $R = I$.

of the six states and the input are respectively around 1.5, 0.6, 0.5, 5, 1, 5 and 1. Thus, according to the Bryson's rules, we can set the Q and R with this information to start tuning. Then after some fine tuning, we finally got the optimized LQR parameters as below

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.167 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.167 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.167 \end{bmatrix}, \quad (5.3)$$

$$R = 0.8.$$

With these Q and R , we will have

$$K = [-1.1180 \quad 135.6454 \quad -70.5044 \quad -1.8397 \quad 36.1255 \quad -2.7697]. \quad (5.4)$$

The simulation result is in Figure 5.3. Clearly, the performance of the LQR is improved by tuning.

The negative peak value of the position variation of Link 1 is nearly 50° in the improved system

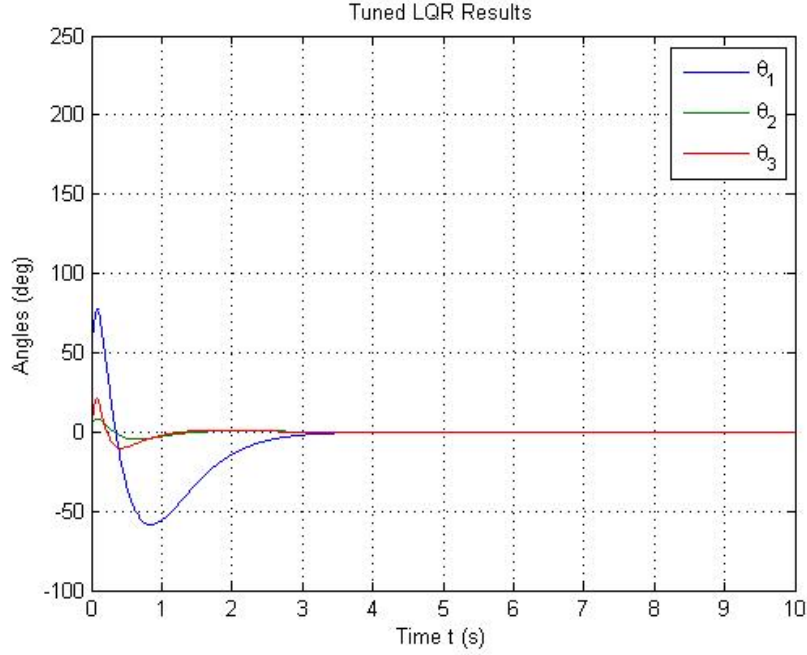


Figure 5.3: Tuned LQR simulation result.

while it is 25° larger in the original system. Besides, at around $3.5s$ the tuned system has already converged to zero while the original system is still on its way.

As we mentioned above, the control signals for the two cases are shown in Figure 5.4. Here we scale the control signals from -2 to 2 by truncating the initial large negative peaks in both of the LQR and its tuned version. For the LQR, the initial peak is -5.3961 and for the tuned one it changes to -8.1124. We can see that, the tuning control signal react quickly than the original one. It looks like the positive peak of the tuning LQR is higher than the original one, however, since it falls down faster than original one, the total energy of the tuned system used for control is not large compared with the original one. In fact, using (3.6) to calculate the total energy of the control process for these two cases, we find that the control energy cost for the tuned LQR is 5.8730, which is much smaller than the total cost of the original LQR, 19.7358. Note that here there is an intense change in both of

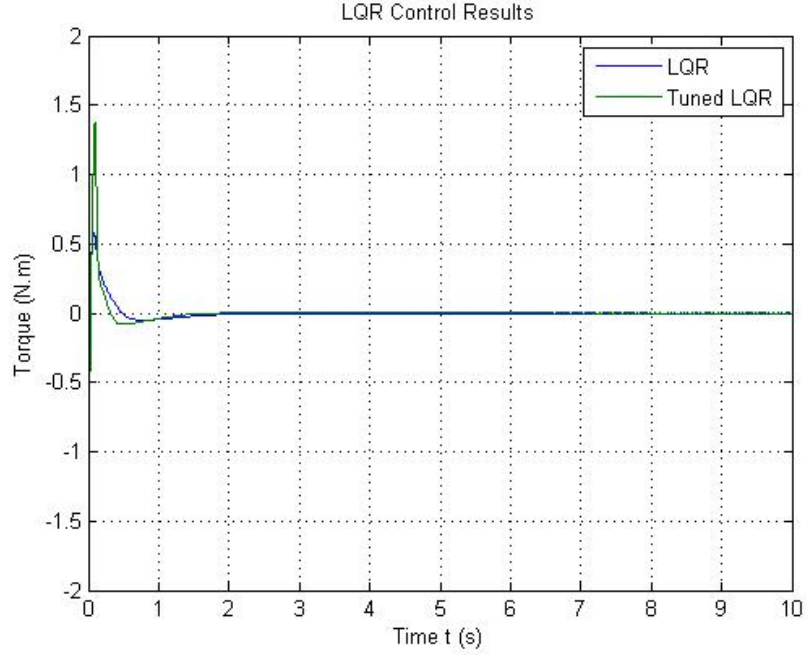


Figure 5.4: Control signals of the LQR and its tuned version.

the control signal at the beginning of the control process. We will see the same phenomenon in the DAFC Simulation.

5.3 Direct Adaptive Fuzzy Control Results

Firstly, we provide a case that the LQR loses its impact on controlling the system, even with its parameters tuned. For the initial states $x(0) = [0 \quad -0.1 \quad 0.1 \quad 0 \quad 0 \quad 0]^T$, the system states will diverge over time. We will see that the RDIP system can be controlled with the adaptive LQR control scheme.

z_i	c	σ
1	1	0.25
x_1	1	0.25
x_2	0.1	0.025
x_3	0.2	0.05
x_4	4	1
x_5	0.6	2
x_6	3	0.75

Table 5.1: Membership function settings for the DAFC.

A Gaussian membership function with the following form is used here, which is shown in Figure 4.1 in the previous Chapter.

Left:

$$\mu(x) = \begin{cases} 1 & \text{if } x \leq -c, \\ e^{-\left(\frac{x+c}{2\sigma}\right)^2} & \text{if } x > -c. \end{cases} \quad (5.5)$$

Center:

$$\mu(x) = e^{-\left(\frac{x}{2\sigma}\right)^2}. \quad (5.6)$$

Right:

$$\mu(x) = \begin{cases} e^{-\left(\frac{x-c}{2\sigma}\right)^2} & \text{if } x < c, \\ 1 & \text{if } x \geq c. \end{cases} \quad (5.7)$$

With this function, we set the membership function for each term of the z matrix as in Table 5.1.

Also, we adjust the other parameters in the bounding term, sliding term and the central equivalent term of the control scheme with the options we have picked up in Chapter IV. Note here we need to take care of not only the parameters of the adaptive fuzzy system, but also the LQR parameters Q

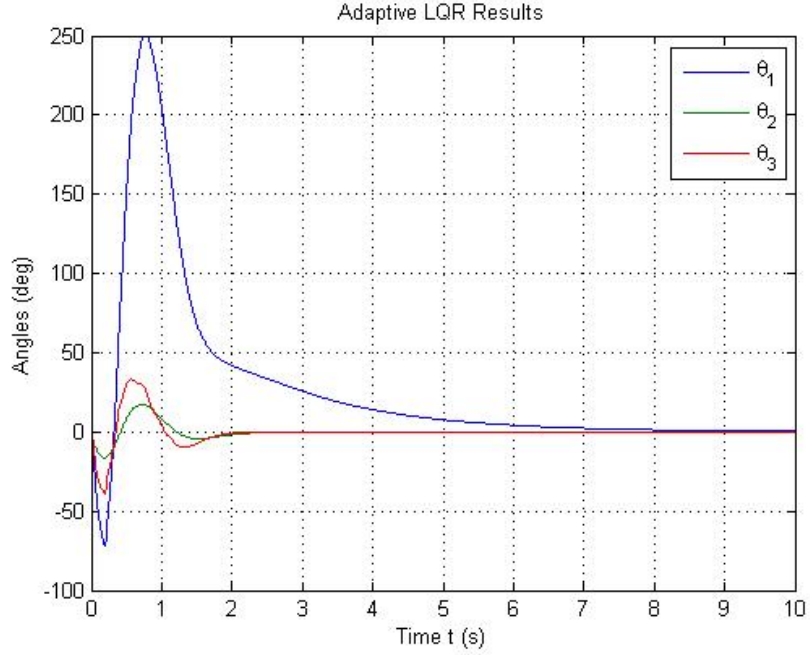


Figure 5.5: DAFC simulation result.

and R matching with the adaptive law. The Q and R are chosen here as

$$Q = \begin{bmatrix} 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.5 \end{bmatrix}, \quad (5.8)$$

$$R = 1.$$

Then we have the the result for the DAFC as in Figure 5.5 and the DAFC control signal as in Figure 5.6, where the control signal is scaled from -20 to 20 for observation by truncating the initial positive peak 58.2286.

In Figure 5.5 we find that the adaptive LQR really works for the given initial states. The system finally converges into zero. In [6], the state x_1 in the case of a single inverted pendulum converges to

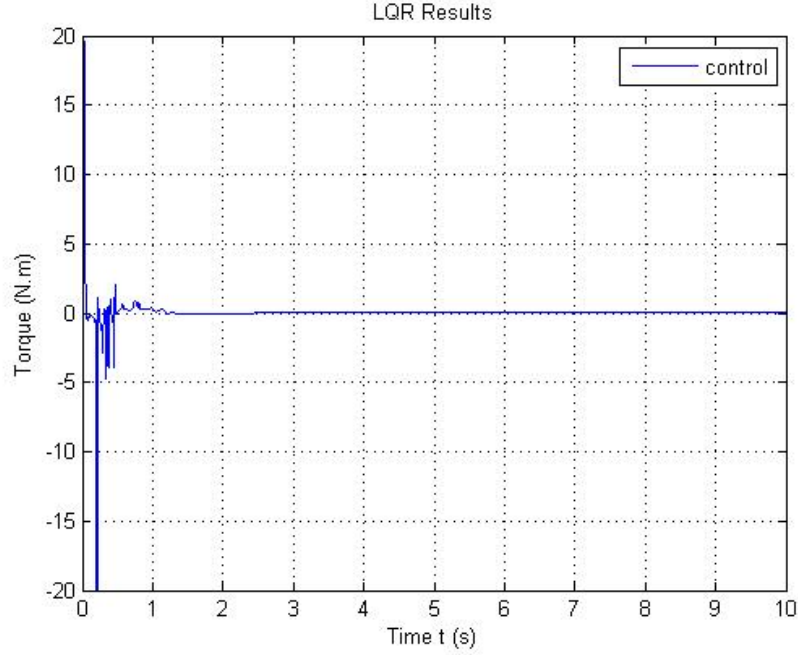


Figure 5.6: DAFC control signal.

a stable state of rotating with constant speed. If we could figure out the RDIP has a non-minimum-phase zero dynamics, together with the finding from that paper with ours, we conclude that the DAFC can work on a non-minimum-phase system and the system states will converge to a stable state. Further analysis is still needed for a theoretical evidence of the effectiveness of the DAFC among a large scale of non-minimum-phase systems.

From Figure 5.6 we can see that the control signal of the DAFC method vibrates more intensely than the conventional LQR. We find that there is a sudden change at the beginning of the control process, which also happens in the original LQR case. As we mentioned before, the DAFC tries to follow the known controller part and optimize it. Thus it may inherit some characteristic of the known controller. The magnitude of the DAFC control signal seems larger than the LQR, but it indeed improves the LQR controller by making the system stable in the given initial states that the

LQR loses its power. In addition, the control signal of the DAFC has a magnitude much larger than the LQR. Since a high feedback gains may lead to torque saturation, noise amplification, and other problems in the experiment, we will need to find a way to avoid it in future.

From the characteristics information of the DC motor for the SV02+DBIP experiment in the “Quanser Systems and Procedures” technique document in the lab KL302, we can figure out that the motor can provide to the system with a torque no less than $83.8856 N \cdot m$, which is large enough for our DAFC control scheme since the largest control signal peak that we have got is 58.2286.

In Figure 5.7, we provide the comparison of the R.O.A of the original LQR and the DAFC. Here we sample for the initial states of the system in a range where $\theta_i \in [-0.1, 0.1]$, *for* $i = 1, 2, 3$ and assume $\theta_i = 0$, *for* $i = 4, 5, 6$ to simplify the comparison and make it possible to show the result on a 3-D plot. We sample θ_1 and θ_2 at an interval of 0.02, and we sample θ_3 at an interval of 0.01. The blank region within the cube is where both the LQR and the DAFC works. The blue points shows where the DAFC still works in these initial states while the original LQR has lost its power to make the system converge to the zero point. We can see clearly that the DAFC actually increase the stability of the LQR controller.

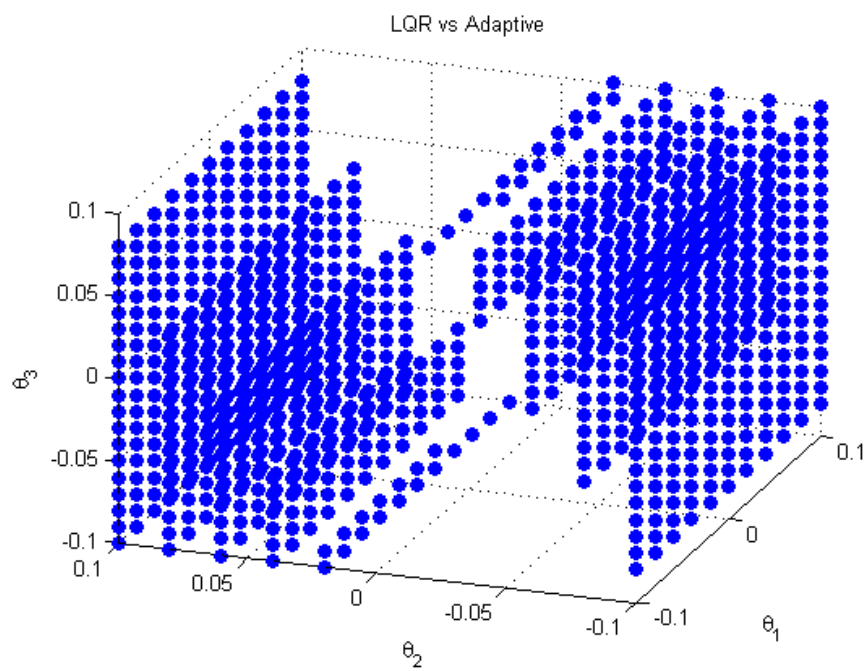


Figure 5.7: LQR vs. DAFC.

CHAPTER VI

CONCLUSION

We have studied two control approaches for the RDIP. First of all, a mathematical model is built with the E-L method. The rotary frictions of the links are considered in model building while we ignore the static friction of the system.

In the next step, we have developed a LQR controller for the system after linearizing the system with two methods which is equivalent with each other. The LQR method presents an adequate behavior on the plant in terms of our basic control objective to balance the pendulum. We introduce the Bryson's rule as a starting point for tuning the LQR controller, and then improve the performance of the LQR with some fine tuning. For the exploration of the stability of the LQR, we discuss the Lyapunov stability of the LQR controller and then get the R.O.A for the system. We can find that beyond the scope of the R.O.A, there are still a large area that the LQR controller works, such as the initial states we have used for the original LQR and its tuning test. But we cannot claim that there exists a neighborhood outside the R.O.A where the LQR control can always work. For example, in the initial states $[0 \quad -0.05 \quad 0.1 \quad 0 \quad 0 \quad 0]^T$ the LQR controller will lose its power and this point is much closer than the point $[1 \quad 0.1 \quad 0.1 \quad 0 \quad 0 \quad 0]^T$ which has been proven to be stable.

The R.O.A of the RDIP system is still limited; one can continue to adjust the parameters of the conventional LQR and the Adaptive LQR controllers to enlarge it.

Since the performance of the LQR is limited, we tried to optimize the LQR with the DAFC. This method directly approximates the ideal controller by using the T-S fuzzy set. We are able to increase robustness using this method compared with the LQR. We find that with this method, the good characteristic of the LQR can be retained and at the same time the benefits of adaptation are added. Following the simulation result in [6], we applied the adaptive fuzzy control scheme in a more complex system and get a better control result which shows that all the states of the system converges to zero, while the base arm in [13] converge to another stable state, a constant-speed rotational movement. It indicates that the DAFC is able to improve the LQR by improving its robustness adaptively. It is possible to design a DAFC that gives us bounded states in spite of the marginal stability of the zero dynamics. However, we provided no theoretical justification of the fact that this design works as it does.

The DAFC can also be improved in some ways. An approach that can be applied is the incorporation of heuristics about the inverse plant dynamics to speed the adaptation. An inverse plant is a fuzzy system that is heuristically designed to roughly approximate the plant's inverse dynamics. The details about it is discussed in [12].

One must be careful in trying to evaluate these results. It is probably not fair to say that the LQR failed and the DAFC succeeded, recalling that the pendulum does not satisfy the zero-dynamics assumption of the DAFC method. However, our experience indicates that at least in some cases, the adaptive fuzzy method we have investigated has an advantage with respect to the conventional LQR

method. It allows for more design flexibility. This is clearly illustrated by our adaptive design. The DAFC using a LQR as the known part of the controller displays an improved behavior in comparison with the conventional LQR technique. Apparently, the use of our knowledge of what the control law should be helpful to increase the robustness of the algorithm. We manage to obtain an improvement by the adaptive technique.

Although the result we have obtained seems to indicate that the adaptive LQR can improve the original LQR and even work with systems with non-minimum-phase zero dynamics, it is still necessary to evaluate the performance of the DAFC under a greater variety of conditions. It remains to be investigated how robust the controllers are against many different types of disturbances, for instance, we did not study how the adaptive fuzzy controllers react to a “white noise” disturbance in the control input.

Another thing needs to be mentioned is that, we can see from Figure 5.6, there is a large peak in the control signals at the beginning of the control process. It might be due to the zero initial states of A_u . One can try to reset the initial states of the fuzzy set to improve the magnitude of the control input.

As we mentioned before, this paper is a theoretical preparation for the RDIP experiment in the lab KL302. One can implement these control schemes in experiment to verify the simulation results, which will be a great challenge since in the experiment, there may be some disturbance, unknown dynamics or other factors.

BIBLIOGRAPHY

- [1] V. Sukonatanakarn and M. Parnichkun, “Real-time optimal control for rotary inverted pendulum,” *American Journal of Applied Sciences*, 2009.
- [2] H. K. Khalil, *Nonlinear Systems*. Upper Saddle River, NJ: Prentice Hall, 2002.
- [3] R. W. Brockett and H. Li, “A light weight rotary double pendulum: Maximizing the domain of attraction,” in *IEEE Decision and Control Conference*, (Maui, Hawaii), pp. 3299–3304, Dec 2003.
- [4] J. Driver and D. Thorpe, “Design, build and control of a single/ double rotational inverted pendulum,” tech. rep., School of Mechanical Engineering, The University of Adelaide, 2004.
- [5] V. Casanova, J. Salt, R. Piza, and A. Cuenca, “Controlling the double rotary inverted pendulum with multiple feedback delays,” *International Journal of Computers Communications and Control*, vol. 7, pp. 20–38, Mar. 2012.
- [6] R. Ordonez, J. Zumberge, J. T. Spooner, and K. M. Passino, “Adaptive fuzzy control: Experiments and comparative analysis,” *IEEE Transactions on Fuzzy Systems*, vol. 5, pp. 167–188, May 1997.
- [7] C. Fox, *An Introduction to the Calculus of Variations*. New York, NY: Dover Publications, 2010.
- [8] J. R. Movellan, “Dc motors.” <http://mplab.ucsd.edu/tutorials/DC.pdf>, Mar. 2010.

- [9] Y. Wang, "Rotation double inverted pendulum," tech. rep., School of Electrical and Computer Engineering, The University of Dayton, Apr. 2012.
- [10] Z. Gajic, *Linear Dynamic Systems and Signals*. Upper Saddle River, NJ: Prentice Hall, 2002.
- [11] J. P. Hespanha, "Lecture notes on lqr/lqg controller design." <http://www.uz.zgora.pl/~wpaszke/materialy/kss/lqrnotes.pdf>, Feb. 2005.
- [12] J. T. Spooner and K. M. Passino, "Stable adaptive control using fuzzy system and neural networks," *IEEE Transactions on Fuzzy Systems*, vol. 4, pp. 339–359, Aug. 1996.
- [13] K. Tanaka and M. Sugeno, "Stability analysis and design of fuzzy control systems," *Fuzzy Sets and Systems*, vol. 45, pp. 135–156, Jan. 1992.
- [14] T. Procyk and E. Mamdani, "A linguistic self-organizing process controller," *Automatica*, vol. 15, no. 1, pp. 15–30, 1979.
- [15] J. R. Layne and K. M. Passino, "Fuzzy model reference learning control," *J. Intell. Fuzzy Syst.*, vol. 4, no. 1, pp. 33–47, 1996.
- [16] J. R. Layne and K. M. Passino, "Fuzzy model reference learning control for cargo ship steering," in *IEEE Contr. Syst. Mag.*, pp. 23–24, Dec 1993.
- [17] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *TSMC*, 1985.
- [18] J. T. Spooner, M. Maggiore, R. Ordonez, and K. M. Passino, *Stable Adaptive Control and Estimation for Nonlinear Systems*. New York, NY: John Wiley and Sons, 2002.
- [19] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *TSMC*, vol. 15, pp. 116–132, Jan. 1985.
- [20] S. Sastry and M. Bodson, *Adaptive Control: Stability, Convergence and Robustness*. Englewood Cliffs, NJ: Prentice Hall, 1989.