INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI®
Stable Auto-Tuning of Adaptive Controllers for Nonlinear Systems

DISSERTATION

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Hazem Numan Nounou, B.S., M.S.

* * * * *

The Ohio State University

2000

Dissertation Committee:

Kevin M. Passino, Adviser
Jose B. Cruz, Jr.
Stephen Yurkovich

Approved by

Adviser
Department of Electrical Engineering
© Copyright by

Hazem Numan Nounou

2000
ABSTRACT

This dissertation focuses on improving the performance of direct and indirect adaptive control methodologies for nonlinear discrete and continuous-time systems via the use of additional cost functions to achieve stable and hopefully higher performance operation. We will in particular focus on adjusting the parameters of the adaptive laws by optimizing some cost functions of interest (e.g., instantaneous control energy). In this dissertation, the adaptation gain and the direction of descent for discrete-time systems are updated on-line to optimize some cost function. The auto-tuning is performed in such a way that stability is maintained for any value of the adaptation gain within the feasible range. We show that auto-tuning the adaptation gain can be viewed as a special case of auto-tuning the direction of descent. Based on simulation results, we find that the lower bound of the adaptation gain tends to be used when the output error is very small. In the case where the output error is fairly large, a large adaptation gain tends to be used. Also, for one example in the direct case the auto-tuning algorithm is able to achieve a mean squared error (MSE) that is smaller in magnitude than the MSE that can be achieved using any fixed adaptation gain, with a relatively small control energy. In the indirect case, however, for this example the auto-tuning algorithm is only able to achieve an acceptable MSE with a
relatively small control energy. Based on simulation results, we find that adapting the direction of descent can be used to trade-off between the tracking performance and control energy.

Next, the adaptation gain for continuous-time systems is updated on-line to minimize the instantaneous control energy. For this purpose, a gradient-based hybrid adaptive law is used for parameter adaptation. Using this update law, some local results are established and boundedness of the control and output variables is provided. Since the hybrid adaptive law only guarantees that a function of the output error (not the output error) is driven to a small value, it may not be always possible to drive the output error to a small value. Based on simulation results, it is found in the direct case that it is possible to drive the output error to a small value. Unfortunately, this is not feasible in the indirect case.
To my parents, brothers, and sisters.
ACKNOWLEDGMENTS

I would like to thank my advisor, Prof. Kevin M. Passino, for his excellent guidance and generous support during my graduate studies. He has been a great advisor not only on the technical level, but also on the personal one. His excellent guidance has been one of the key elements that helped me succeed. Many words I can say but none can really express my sincere thanks to him.

I would like also to thank Professors Jose B. Cruz Jr. and Stephen Yurkovich for being part of my dissertation committee.

Finally, I want to thank my parents for their unconditional love and support throughout my life. They believed in me and they taught me how to believe in myself. Also, I would like to thank my brothers and sisters for their love and for being good examples to follow.
VITA

November 10, 1972 ............................................. Born - Gaza, Gaza Strip, Palestine

1995 ......................................................... B.S. Electrical Engineering
(Magna Cum Laude)
Texas A&M University
College Station, TX

1997 ......................................................... M.S. Electrical Engineering
The Ohio State University
Columbus, OH

1996-present ................................................ Graduate Research Associate,
The Ohio State University.

PUBLICATIONS

Research Publications


FIELDS OF STUDY

Major Field: Electrical Engineering
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>ii</td>
</tr>
<tr>
<td>Dedication</td>
<td>iv</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>v</td>
</tr>
<tr>
<td>Vita</td>
<td>vi</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xi</td>
</tr>
<tr>
<td>List of Figures</td>
<td>xii</td>
</tr>
<tr>
<td>Chapters:</td>
<td></td>
</tr>
<tr>
<td>1. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Motivation</td>
<td>2</td>
</tr>
<tr>
<td>1.1.1 Adaptive and Optimal Control</td>
<td>2</td>
</tr>
<tr>
<td>1.1.2 Relationships Between Adaptive and Optimal Control</td>
<td>3</td>
</tr>
<tr>
<td>1.1.3 Motivation for Improvement</td>
<td>7</td>
</tr>
<tr>
<td>1.2 Overview of Existing Work in the Literature</td>
<td>8</td>
</tr>
<tr>
<td>1.2.1 Adaptive Control for Continuous Time Systems</td>
<td>8</td>
</tr>
<tr>
<td>1.2.2 Adaptive Control for Discrete Time Systems</td>
<td>14</td>
</tr>
<tr>
<td>1.2.3 Adaptive Optimal Control</td>
<td>16</td>
</tr>
<tr>
<td>1.3 Dissertation Summary</td>
<td>21</td>
</tr>
<tr>
<td>2. Discrete-Time Nonlinear Adaptive Control Systems</td>
<td>24</td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>24</td>
</tr>
<tr>
<td>2.2 Plant Description and Assumptions</td>
<td>24</td>
</tr>
<tr>
<td>2.2.1 Plant Description</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>viii</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Direct Approach</td>
</tr>
<tr>
<td>5.2.4</td>
<td>Indirect Approach</td>
</tr>
<tr>
<td>5.3</td>
<td>Hybrid Adaptive Law: Update and Stability</td>
</tr>
<tr>
<td>5.4</td>
<td>Auto-Tuning the Adaptation Gain</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Auto-Tuning Algorithm</td>
</tr>
<tr>
<td>5.4.2</td>
<td>Stability Analysis</td>
</tr>
<tr>
<td>5.5</td>
<td>Aircraft Wing Rock Example</td>
</tr>
<tr>
<td>5.5.1</td>
<td>Direct Case</td>
</tr>
<tr>
<td>5.5.2</td>
<td>Indirect Case</td>
</tr>
<tr>
<td>5.6</td>
<td>Concluding Remarks</td>
</tr>
<tr>
<td>6.</td>
<td>Conclusions</td>
</tr>
<tr>
<td>6.1</td>
<td>Summary</td>
</tr>
<tr>
<td>6.2</td>
<td>Contributions</td>
</tr>
<tr>
<td>6.3</td>
<td>Future Directions</td>
</tr>
<tr>
<td></td>
<td>Bibliography</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Summary of parameters</td>
<td>114</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>48</td>
</tr>
<tr>
<td>3.2</td>
<td>60</td>
</tr>
<tr>
<td>3.3</td>
<td>73</td>
</tr>
<tr>
<td>3.4</td>
<td>74</td>
</tr>
<tr>
<td>3.5</td>
<td>75</td>
</tr>
<tr>
<td>3.6</td>
<td>76</td>
</tr>
<tr>
<td>3.7</td>
<td>77</td>
</tr>
<tr>
<td>3.8</td>
<td>79</td>
</tr>
<tr>
<td>3.9</td>
<td>81</td>
</tr>
<tr>
<td>3.10</td>
<td>82</td>
</tr>
<tr>
<td>3.11</td>
<td>83</td>
</tr>
<tr>
<td>3.12</td>
<td>84</td>
</tr>
</tbody>
</table>
3.13 Changes in the MCE and MSE as \( \eta \) varies in the indirect adaptive case (Algorithm 1). .................................................. 85

3.14 Changes in the MCE and MSE as \( \eta \) varies in the indirect adaptive case (Algorithm 2). .................................................. 86

4.1 Two-dimensional geometrical presentation. ................................. 93

4.2 Performance of direct adaptive controller when direction is adapted. . 100

4.3 Performance of direct adaptive controller when direction is not adapted. 101

4.4 Variations in MSE and MCE as \( \bar{\sigma} \) varies. ............................... 102

5.1 Steps used for adaptation gain selection. ....................................... 125

5.2 The responses of \( \epsilon \) and \( \eta \) in the direct case. ....................... 138

5.3 The responses of \( \epsilon \) and \( \eta \) in the direct case. ....................... 139

5.4 The parameter vector, \( A \), in the direct case. ............................... 140

5.5 The responses of the aircraft roll angle and aileron input in the direct case for \( \delta = 30 \) ......................................................... 141

5.6 The responses of the aircraft roll angle and aileron input in the direct case for \( \delta = 5000 \) ......................................................... 142

5.7 Changes in the MCE and MSE as \( \eta \) varies in the direct adaptive case. 143

5.8 The responses of \( \epsilon \) and \( \eta \) in the indirect case. ....................... 145

5.9 The parameter vector, \( A \), in the indirect case. ............................... 146

5.10 The responses of the aircraft roll angle and aileron input in the indirect case for \( \delta = 3 \) ......................................................... 147

5.11 The responses of the aircraft roll angle and aileron input in the indirect case for \( \delta = 20 \) ......................................................... 148

5.12 Changes in the MCE and MSE as \( \eta \) varies in the indirect adaptive case. 149

xiii
CHAPTER 1

INTRODUCTION

In this dissertation we focus on improving the performance of direct and indirect adaptive control for nonlinear discrete and continuous-time systems by auto-tuning some of the controller parameters. The adaptation mechanisms of adaptive controllers are usually formulated such that some cost function is optimized. However, improper definition of the cost function can result in a significant deterioration of the closed-loop performance. This represents a major drawback of most adaptive control schemes. The main focus of this dissertation is to address this problem. One approach to improve the performance of adaptive controllers is to attempt to optimize the adaptive law. Successful completion of this step can (loosely speaking) lead to the development of an “optimal adaptive control” algorithm. One way to approach this problem is to simultaneously update some of the parameters in the update law itself. The adaptation gain and the direction of descent for discrete-time systems are updated on-line to optimize some cost function of interest. The auto-tuning is performed in such a way that stability is guaranteed for any value of the adaptation gain within the feasible range. For continuous-time systems, the adaptation gain is updated on-line to minimize the instantaneous control energy when a gradient-based hybrid adaptive law is used for parameter adaptation. Using this update law, some
local results are established and boundedness of the control and output variables is provided.

1.1 Motivation

To establish and to provide motivation of our research, we start by reviewing some of the most relevant control areas to our research. Then, we discuss the relevance of these areas to our research topic.

1.1.1 Adaptive and Optimal Control

The problem of controlling complex nonlinear systems has captured the interest of many researchers due to its important industrial applications. The linear control problem has been studied extensively and relatively complete solutions have been provided for the non-adaptive (for continuous systems as in [1, 2], and for discrete systems as in [3]) and adaptive [4] cases. On the other hand, the nonlinear control problem represents a great challenge to the control designer, especially when one tries to achieve a desired level of performance for partially or completely unknown plants. The level of difficulty of any control problem depends heavily on the amount of prior knowledge we have about the plant we wish to control. Any control problem can be classified according to one of the following three categories: black-box control problem, white-box control problem, and grey-box control problem [5]. In the first case, the plant we wish to control is assumed to be completely unknown. Such a problem is generally very difficult, if not impossible, to solve analytically. In the second case, however, the plant is assumed to be completely known. Solutions for this problem are very well developed for linear systems. However, analytical solutions may be very difficult to find for general classes of nonlinear systems. Lastly, the third
case deals with partially known plants; this explains the choice of the term \textit{grey} since the amount of knowledge we have about the system lies between the amounts of knowledge we have about the systems in the white-box and black-box control problems. Fixed controllers that meet the desired performance for this class of plants (which are most commonly encountered in practice) are difficult to find although much effort is expected in this direction under the general label of “robust control.” Due to this, the control community, since the second half of the 1950s, started to search for a systematic approach for automatic adjustment of the controllers in real time, in order to achieve or to maintain a desired level of performance of the control system when the parameters of the plant dynamic model are unknown and/or change in time \cite{6}. Such a controller is known as an “adaptive controller.” Essentially, in adaptive control \textit{a priori} information is augmented with information that is gathered on-line in order to control the system. In the literature, the terms “adaptive control,” “self-adaptive control,” and “learning and adaptive control” have been used interchangeably.

It is interesting to note that, in the literature, adaptive systems/controllers have been defined in many ways as in \cite{4, 5, 6, 7}. Adaptive systems/controllers can also be defined in terms of optimality principles and optimal design. This approach will be elaborated in the next section.

\subsection{1.1.2 Relationships Between Adaptive and Optimal Control}

The control design problem (for any system, linear or nonlinear) can be thought of as a problem of finding a controller structure and/or parameters that achieve desired closed-loop specifications that are defined by the designer. Such a controller can loosely be called “optimal” with respect to the desired specifications if it is viewed
as the best one according to some performance index. The optimum performance is heavily dependent on the performance index function selected. Note that the performance index is not the only term used in the literature; it is also known as a performance criterion, a performance measure, a penalty function, a profit function, a cost function, and more. For consistency, "cost function" will be used throughout the rest of this document.

Before discussing the relationship between adaptive and optimal control, it is worthwhile to analyze linear feedback design in terms of optimum design, which can in some cases be expressed analytically. Then, we will move to the nonlinear case (that sometimes requires computer-based optimum design algorithms that use real-time cost functions) where for the most part no analytical solutions can be derived (or at least no one has been able to yet for very general nonlinear systems).

**Linear Control Design as an Optimal Control Problem**

In the context of linear control system theory, most of the design specifications encountered in practice can be thought of in terms of cost functions since the designer attempts to find the controller parameters (assuming a known structure) so that the desired performance is met. Good examples of frequency-domain "cost functions" are phase margin, gain margin, percent overshoot, and others. A great deal of work has been done in this area [8, 9]. Note that for a specific design specification, it is often possible to find one or more sets of controller parameters that meet such a specification (i.e., one or more analytical solution). It is then the task of the designer to choose the best of these solutions with respect to other specifications. It is important to note that such analytical solutions are only feasible for linear systems (with small order) and simple cost functions. For (fixed or time-varying) uncertain
nonlinear systems such analytical solutions are difficult to find. Therefore, the need for computer-based optimum design techniques arises. Such techniques require some "time-domain cost functions" (i.e., those that are computed for time signals) that can be used in the linear as well as the nonlinear case. In the next section, a brief discussion on time-domain cost functions will be outlined.

Types of Cost Functions used in Optimal and/or Adaptive Control

Because of the importance of cost functions in adaptive/optimum design, we will briefly discuss some of the commonly used cost functions, some of which will be used in later sections.

One of the most commonly used cost functions is the sum of squared error (SSE) which can be defined in the discrete case as

$$SSE = \sum_{k=1}^{N} e^2(k),$$

where \(e(k)\) is, for example, the closed-loop output error which is summed over a time window of length \(N\). Note that when \(N = 1\), the SSE becomes the instantaneous squared error function. Sometimes the signal is weighted to emphasize the importance of matching at certain times. This cost function is commonly used because of its nice mathematical properties (i.e., a convex function that is usually easy to handle).

Another cost function of interest is the sum of control energy (SCE). Unlike the previous cost function, this one is defined in terms of the control as follows:

$$SCE = \sum_{k=1}^{N} u^2(k),$$

where \(u(k)\) is the control applied to the system. This cost function is used in practice since in many applications it is desirable to achieve an acceptable performance with
the least control possible. There are many other cost functions that are not as widely used as the ones listed above. For a more detailed discussion on types and uses of cost functions, refer to [7] and references within.

Next, we will discuss how adaptive control can be defined in terms of optimum design.

**How Can Adaptive Control be Related to Optimum Design?**

Unlike linear systems, nonlinear systems can have complex and/or time-varying dynamics that make the problem of finding a fixed controller that achieves and maintains an acceptable closed-loop performance very challenging. Therefore, the need for adaptive controllers, that change their parameters on-line (subject to a selected cost function that defines the desired performance) to cope with the problems associated with complex nonlinear systems, arises. As discussed earlier, cost functions are used in the context of optimum design; they define the desired performance of the closed-loop system in such a way that when the optimization problem is solved, a set of controller parameters that makes the closed-loop system behave as desired is obtained. For linear time-invariant systems, this process can be designed so that optimum solution needs to be found only once; such a problem is called static optimization in which an analytical solution is usually feasible to obtain, at least in certain special cases. Here, however, since nonlinear systems are considered, the process of finding a set of parameters that satisfies the desired performance defined by the cost function is carried out over time, not only to obtain but also to maintain an acceptable performance. Therefore, using this line of reasoning the adaptive control problem can be defined as an on-line (or dynamic) optimum design problem. It is important to note that the solution of such a problem can be very challenging to
obtain if more than one cost function is considered at once, and/or a large number of controller parameters are to be found. Therefore, the use of accurate and efficient computationally-based algorithms is essential for such purposes. Two major update routines that are widely used in the literature for online optimum design are the gradient and least squares algorithms. Detailed description and analysis of these algorithms can be found in [10].

### 1.1.3 Motivation for Improvement

Since most systems encountered in practice are nonlinear, nonlinear adaptive controllers are needed. The adaptation mechanisms of such controllers are formulated such that some cost function is minimized. However, improper definition of the cost function can result in a significant deterioration of the closed-loop performance. This represents a major drawback of most adaptive control schemes. The main objective of this dissertation is to address this problem. A step toward further improving the performance of adaptive controllers is to attempt to optimize the adaptive law. Successful completion of this step can (loosely speaking) lead to the development of an "optimal adaptive control" algorithm. One way to approach this problem is to simultaneously update (i.e., optimize in terms of some desired cost function) some of the parameters in the update law itself. This step, if a proper choice is made in selecting a cost function of valuable interest, can be of a significant importance. Good examples on this include updating the parameter(s) that define the gain of adaptation, or attempting to optimize the direction of the update law (if a search method is used). It is these two ideas that are introduced and investigated in this dissertation. We will
pay particular attention to development of such methods to achieve stable closed-loop control.

Next, an overview of standard and optimal adaptive control literature (in terms of cost functions) will be discussed for both linear and nonlinear systems.

1.2 Overview of Existing Work in the Literature

Here, we present an overview of indirect, direct, and the so called optimal adaptive control schemes for linear and nonlinear systems, as well as some remarks on adaptive control.

1.2.1 Adaptive Control for Continuous Time Systems

A great deal of work in the adaptive control literature focuses on continuous-time systems. Most of the adaptive control approaches can be classified under indirect or direct adaptive control that will be overviewed next.

Indirect Adaptive Control

The basic principle of this approach is that an adaptive controller can be designed if a model of the plant is estimated on-line from available real-time input-output measurements. It is called “indirect” since the adaptation of the controller parameters is performed in two steps: First, the plant parameters are estimated on-line. Second, the controller parameters are estimated on-line based on the current estimate of the plant parameters assuming they are the true ones. This assumption is known in the literature as the “certainty equivalence principle.” Once the plant model parameters are estimated (and the certainty equivalence principle is used), the controller parameters are updated to minimize the error between the output of the plant and
the desired reference input, or the output of a reference model in model reference adaptive control (MRAC). In MRAC, the reference model characterizes the desired performance of the closed-loop system.

An overview of the existing work of indirect adaptive control for linear and non-linear systems will be presented next.

**Linear Case**

In the literature, indirect adaptive control was initially introduced in 1958 [11] in relation to digital process control, where the author approximates the process on-line by a linear system (expressed as a linear difference equation). The update of the parameters is based on minimizing the weighted mean squared tracking error. Then, the parameters of the controller (which are also assumed to be linear) is updated to minimize some cost function chosen based on what aspect of system response to be optimized. Then, the authors in [12] used the term "self-tuning" in relation to indirect adaptive control. In this paper, an adaptive minimum variance control (MVC) law for linear discrete time stochastic systems is computed based on the assumption that the model parameter estimates converge. The term MVC is used since the control is found such that the variance of the output error is minimized. Further developments of this approach continued at the end of 1970s. In [13], the authors further discussed closed-loop stability and parameter convergence issues. In [14], a control for a class of multivariable linear stochastic systems is presented. In this paper, asymptotic convergence issues of model parameter estimates (using least squares algorithm) are
discussed. Similar results using pole-zero [15] and pole placement [16] design for updating the model parameters were introduced. Similarly, this algorithm was applied using predictors to estimate the future behavior of the system [17]. Further consideration of the problem to investigate least squares parameter convergence in adaptive control was recently discussed in [18] (and more references are cited within), where an adaptive design procedure that does not require a persistency of excitation condition was developed for deterministic and stochastic linear systems. For further overview and analysis of this technique, refer to [19, 4] and references within.

Since most systems to be controlled are nonlinear, nonlinear approximators (that approximate parts of plant dynamics) are needed. An overview of nonlinear approximation-based indirect adaptive control (that was profoundly impacted by the development of nonlinear function approximation theory) will be presented next.

Nonlinear Case

The advancements in nonlinear function approximation theory (that did not start until the 1980s due to theoretical and computational limitations) made it possible to consider the problem of nonlinear indirect adaptive control at the beginning of this decade. Nonlinear approximators, such as fuzzy systems and neural networks, that possess the universal approximation property, have been widely used for this purpose. This technique started with use of neural networks to control continuous time nonlinear systems [20, 21, 22]. Later, this line of research was widely considered not only using neural networks as nonlinear function approximators to approximate parts of the plant dynamics [23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33], but also using fuzzy
Neural Networks and fuzzy systems can be used for the same purpose; the main difference is in the choice of the approximator structure. This fact is shown in [32], where neural networks and fuzzy systems are used equivalently for adaptive control schemes. Both of fuzzy systems and neural networks can be categorized, however, in terms of the function parameterization used: linear as in [21, 22, 25, 35, 36, 39, 40, 32, 31, 38, 29], and nonlinear as in [23, 24, 27, 28, 30, 33].

In [21], the authors considered the problem of identification and control of affine SISO systems. For identification, they used both linear and nonlinear in the parameter neural networks to approximate the plant dynamics. The control law, however, consists of a certainty equivalence term and a sliding mode term. To update the parameters, a projection algorithm is used; to avoid the singularity problem (i.e., the problem of having a zero estimate of the function multiplied by the control), a compact set that guarantees that the estimate will never be zero is used. Stability analysis shown guarantees uniform ultimate boundedness of the state, and boundedness of the tracking error by the $L_2$ norm of the modeling error.

In [24, 27], the authors used multilayer (nonlinear in the parameter) neural networks to control SISO and MIMO continuous-time systems in the feedback linearizable form with a general relative degree. The objective of the control is to drive the output of the plant to track a desired reference trajectory. The network parameters are updated on-line using a gradient algorithm with a dead zone. Local convergence results, that restrict the initial conditions to be close enough to the true ones, are provided; if the initialization assumption is not satisfied, no convergence guarantees are provided. This fact is shown by example, where closed-loop instability resulted in the case where the initial parameters are not close enough to the actual ones.
In [22], the authors presented an algorithm to control a SISO feedback linearizable continuous-time system so that uniform ultimate boundedness of the tracking error to a neighborhood of zero is established. The control law consists a negative feedback term including a weighted combination of tracking errors of the states, an adaptive term, and a sliding mode term.

Similar ideas are used in [35] to design an indirect adaptive control for the same plant structure using fuzzy systems; this again shows the similarity between neural networks and fuzzy systems. To establish asymptotic stability, the authors used a control law consisting of an adaptive term, a proportional term, and two sliding mode terms; the first term is used to keep the state within a compact set, and the second one is used to compensate for the approximation error.

The first global stability results are provided in [28], where the authors considered the class of SISO state-feedback linearizable continuous-time systems. It has been shown that no initial conditions are needed on the parameters (i.e., they do not have to belong to a compact set). For this purpose, a multilayer neural networks based-controller that feedback linearizes the system is used. A control law consisting of a sliding mode term and a certainty equivalence term is used to provide ultimate boundedness of the states and convergence of the tracking error to a neighborhood of zero. Similar results are presented in [36], where fuzzy systems are used to generate the adaptive part of the control law, and a sliding mode term is used to provide the robustification term.

The authors in [32] presented indirect and direct adaptive control algorithms using linearly parameterized fuzzy systems or neural networks to control SISO affine systems with guaranteed convergence of the tracking error to zero in both cases. For the
indirect case, the control law consists of three terms: a bounding control term to ensure boundedness of the output and states, a sliding mode term to compensate for approximation error, and a certainty equivalence term for which it is assumed that the current estimate of the plant parameters are the actual ones. In the direct case, however, an adaptive control term is used instead of the certainty equivalence term. Unique features of the algorithms presented in this paper include the utilization of prior knowledge (about the plant dynamics in the indirect case and the controller dynamics in the direct case) to specify the control and improve the performance, the incorporation of the inverse model dynamics in the direct case to enhance the overall performance, and the applicability of these algorithms to systems containing zero dynamics and/or state dependent input gain. Next, direct adaptive control will be overviewed.

**Direct Adaptive Control**

Direct adaptive control, in which an approximator attempts to directly approximate the ideal controller, was not considered as widely as indirect adaptive control. Direct adaptive control was first introduced in 1958 [41] in relation to the use of a model reference adaptive system for aircraft control. This shows that MRAC and direct adaptive control share some common design features in the sense that the controller parameters in both cases are directly updated. Unique results on this topic are presented in [32] which were discussed above.

In the current and preceding sections, an overview of some of the adaptive control schemes for continuous time systems have been discussed. In the next section, similar discussion for discrete time adaptive control will be presented.
1.2.2 Adaptive Control for Discrete Time Systems

Adaptive control for discrete time systems has not been as widely studied as adaptive control for continuous time systems. Here, an overview of this approach will be presented; then, due to the great relevance of discrete time direct and indirect adaptive control to the context of this dissertation, more details will be provided on these approaches.

General Overview

In the field of adaptive control for discrete-time systems, most of the work has been focused on linear systems; relatively few researchers have considered discrete time adaptive control for nonlinear systems. In the linear case, adaptive controllers have been designed (as shown in [42, 4, 19]) in the context of linear quadratic regulator (LQR), receding horizon control (RHC), and other linear adaptive control techniques based on linear control theory. Note that LQR and RHC are model-based controllers (i.e., the plant model is used as part of the controller) that optimize some cost function (usually quadratic) to obtain the control law. Therefore, optimal design is used on-line for each updated linear plant model. The adaptation of the plant model parameters (assuming a fixed model structure) is usually performed using some projection or search methods. Note that, once the model is updated, many standard linear control techniques can be used to obtain an adaptive control law.

Among the little work done for nonlinear systems, the authors in [43] designed an indirect adaptive controller for input-output discrete time systems (which, as shown in the paper, can be transformed into feedback linearizable systems) using nonlinear in the parameter neural networks to approximate the plant dynamics. They established
asymptotic convergence of the tracking error to a neighborhood of zero by assuming that the parameters of the neural network are close to the actual ones.

The authors in [44] designed an indirect adaptive controller (using linear in the parameter neural network) for discrete-time MIMO non-square feedback linearizable systems with bounded state disturbance, and without input gain nor zero dynamics. Regarding stability results, uniform ultimate boundedness of both parameters and tracking error are established without assuming initialization conditions on the parameters of the network. Similar results are provided in [45] under the same assumption but using multilayer (i.e., nonlinear in the parameter) neural networks. In [46], the authors consider MIMO non-square discrete-time dynamical systems which are assumed to be controllable and its state vector is available for measurement. They use multilayer neural networks to design a model reference indirect adaptive controller for which no persistency of excitation condition is needed, linearity in the parameters is not required, and the certainty equivalence principle is not used. Uniform ultimate boundedness of the tracking error as well as the network weight estimates (which are updated on-line based on delta rule) is achieved.

A different approach is presented in [47], where the authors presented a direct adaptive control for stochastic nonlinear systems in the prediction form (a function of input-and-output regressor vector), where the output is noise corrupted. The control is based on the estimation of the inverse model of the plant using multilayer neural networks. Only local stability results are provided, and no upper bound on the tracking error is specified.
Another different approach is presented in [48], where the control design is performed in two steps: the identification step in which a set of basis functions is found, and control step in which the basis functions are used to find the control law.

The authors in [49, 50] presented a methodology to design direct [49] and indirect [50] adaptive controllers for a class of discrete-time nonlinear systems. The control law presented is based on on-line function approximation (of the controller dynamics in the direct case and the plant dynamics in the indirect case) using Takagi-Sugeno fuzzy systems. Such controllers provide asymptotic tracking of the reference signal to an ε-neighborhood of zero.

Next, an overview of the adaptive optimal control literature will be presented.

1.2.3 Adaptive Optimal Control

Adaptive control has been referred to as "optimal" when the design approach used for the certainty equivalence principle use an optimal control design methodology such as linear quadratic (LQ) control design. Linear quadratic control is an optimal control approach in which the control law is obtained by minimizing a quadratic cost function. Two major LQ-based regulation/tracking control schemes are linear quadratic Gaussian control (LQGC) and model predictive control (MPC). These approaches have been investigated extensively for deterministic and stochastic linear processes [51, 42]. Note, however, that no satisfying advancements have been made in the nonlinear case since no optimality guarantees are provided when the quadratic cost function is solved subject to a nonlinear system. Next, we will briefly present the LQ-based optimal control problem (for which the adaptive version has been studied) as well as an overview of the recent advancements in adaptive optimal control.
Adaptive Linear Quadratic Gaussian Control

A scalar form of the instantaneous quadratic cost function (in discrete form for simplicity) can be written as

\[ J(u) = Q(y(k + 1) - r(k + 1))^2 + Ru^2(k), \quad (1.3) \]

where \( y(k) \) is the output, \( u(k) \) is the input, \( r(k) \) is the reference input, and both of \( Q \) and \( R \) are weightings for the output and input, respectively. The minimization of the first part of the cost function achieves the regulation or tracking objective based on whether the reference signal is zero or non-zero, respectively. For the second part of the cost function, we try to find the smallest control that achieves the desired performance. The parameters \( Q \) and \( R \) are chosen based on the trade-off between the desired performance and admissible control. Note that (1.3) is expressed over a single step; it can also be formulated over an infinite or finite horizon. It is interesting to note that the objective function (1.3) was not the first one to be considered for LQ control design; various cost functions have been introduced during the evolution of research on the cost function (1.3). For instance, the first LQ-based controller was obtained, as shown in [52, 12, 53], by minimizing a cost function that does not include the input. Such a cost function has several drawbacks. For instance, a very large control input may be needed in some cases, such as non-minimum phase systems for which a large control is generally needed to cancel the unstable zeros of the system. These problems have been solved (as presented in [13]) by including the control as a part of the cost function. For a detailed listing and analysis of many quadratic cost functions introduced in the literature, refer to [51].
One of the most important LQ-based optimal control problems is the linear quadratic regulator (LQR) in which the controller finds a unique optimal control (based on the defined cost function) that regulates the output of the plant (i.e., drives it to zero). This problem is very well studied and an analytical solution is derived as shown in [51, 42] and some references within. It is important to note that in the LQ design, it is assumed that the states of the systems are available for measurement. Unfortunately, this is not usually the case. Therefore, there is a need to estimate the current states of the system on-line; such a step can be achieved using a Kalman filter. The joint problem of LQ feedback control and Kalman-based linear estimation is called the LQG control problem. The term Gaussian is used here since this technique is usually applied to linear stochastic systems (linear systems with a disturbance that is assumed to be a Gaussian noise of zero mean). This technique deals with linear systems whose parameters are assumed to be fixed. For unknown or time-varying models, adaptive version of LQG control design is needed.

For discrete-time linear systems, the adaptive LQG control problem has been studied extensively, especially for autoregressive moving average with exogenous input (ARMAX) models. The main idea used to develop the adaptive control law is to use some update algorithm, such as least squares or gradient, to estimate the parameters of the linear model. The major concerns here are to ensure strong consistency of the model estimates and acceptable adaptive tracking. Researchers initially approached the two problems individually: to solve the first problem, the authors in [54, 55, 56, 57, 58] used an extended least squares method, where as the authors in [59, 60] used a stochastic gradient method to solve the second problem. The question that was not yet answered at that point is whether there is a method to solve the two...
problems simultaneously. This question was answered in [61, 62], where the authors used a weighted least squares algorithm to guarantee both consistency of estimates and asymptotic minimality of tracking error. To relax the persistency of excitation assumption in the adaptive optimal control problem, it has always been assumed that there exists an external excitation signal, or the system to be estimated is free from disturbance. In [13], however, it was shown that no persistency of excitation assumption is needed to design least squares based adaptive optimal controllers. They achieved this result using a decomposition of an estimate space for the deterministic case, or using an averaging technique for the stochastic case.

The adaptive LQG control problem has not been studied extensively for continuous time systems [63, 64] as has been done for discrete time systems. In the continuous case, the problems of ensuring stability and optimality have been very difficult to solve in general. Usually the certainty equivalence principle, the excitation condition, or the weighted least squares algorithm is used to ensure consistency of the least squares parameter estimates. The authors in [65] have shown that a complete solution to this problem can be provided using only the controllability and observability assumptions where a weighted least squares algorithm is used to guarantee consistency of parameter estimates. Next, an overview of the adaptive model predictive control will be presented.

**Adaptive Model Predictive Control**

All of the above adaptive control approaches incorporate only available real-time plant information in the adaptation mechanism. Thus, one can classify them as passively adaptive control schemes [66]. Unlike the passively adaptive controllers, “active” adaptive controllers can utilize predictions about future plant behavior in
the adaptation process. A good example of such a scheme is adaptive model predictive controllers (AMPC). Such controllers are the adaptive extension of model predictive control (MPC) which is also known as receding horizon control (RHC). MPC algorithms (a detailed survey of which can be found in [67]) use predictors to predict future behavior of the plant over a finite or infinite horizon, and then find the control law (which is a sequence of control commands) that minimizes the error between the predicted output and the desired reference trajectory. The first input command (which is the input at the current time) is applied to the system and the procedure is repeated at every time step. The methodology of finding the control law used by this approach is based on optimizing some cost function (usually a quadratic cost function due to its useful properties); therefore, MPC algorithms can be thought of as optimal control techniques with respect to the selected cost function. Due to its successful results in several applications, closed-loop stability and performance of MPC have been analyzed for deterministic linear and nonlinear systems. Stability of MPC for linear systems was derived under certain assumptions in [68, 69, 70, 71, 51], and for nonlinear systems in [72, 73, 74, 75, 76, 77]. Lately, special attention has been given by many researchers to nonlinear MPC, particularly fuzzy model predictive control (FMPC). The authors in [78, 79] presented a fuzzy-model based long-range predictive control algorithm where future plant's output is predicted using a relational or simplified relational fuzzy model. In [80], the authors presented a fuzzy control scheme similar to Dynamic Matrix Control (DMC) where they introduced a new way of constructing the dynamic matrix based on the fuzzy model and the fuzzy inverse model. Also, the authors in [81] presented a relational model-based predictive control algorithm in which the control command is a linear combination
of steady state control and the one-step-ahead or dead-beat control. Recently, the authors in [82, 83] presented several fuzzy model predictive control schemes using TSFS where the consequents are linear models.

In MPC design, the model used for prediction is assumed to be fixed; adaptive versions of these algorithms (AMPC) can be obtained by updating the predictor online to cope with the nonlinear and/or time varying components of the plant (as shown in [51] for linear MPC, and in [83] for FMPC), where plant model parameters can be updated using standard projection schemes, such as RLS or a gradient algorithm.

It is very important to note that all of the algorithms discussed in this section deal with adapting linear optimal control techniques. When applying these algorithms it is implicitly assumed that the system we wish to control is linear time-varying or time-invariant (for which no adaptation is needed). Therefore, all these techniques should be classified as "adaptive optimal control," not "optimal adaptive control" schemes as many authors claim.

Next, we will provide a summary of how the dissertation is organized.

1.3 Dissertation Summary

Each of the following chapters is devoted to a specific aspect of the main topic of this dissertation. Chapter 2 deals with direct and indirect adaptive control for a class of nonlinear discrete-time systems. It provides the basic foundation to be used in the successive two chapters. We will first present the system we consider for control along with its assumptions. Then, we will discuss the gradient update law that is used for parameter adaptation. We will also discuss direct and indirect adaptive control schemes, and derive their output error equations. Then, we will establish (based on
the results obtained in [43]) the stability results for both direct and indirect adaptive
control schemes. The stability results established in this chapter are considered local
with respect to the parameters, but not with respect to the state.

Chapter 3 deals with auto-tuning the adaptation gain for the discrete-time system
presented in the Chapter 2. Two algorithms for auto-tuning the adaptation gain are
presented. In the first algorithm, the adaptation gain is selected to minimize the
instantaneous control energy which is very desirable in many applications. In the
second algorithm, however, the adaptation gain selected is the one that produces a
control that is found by optimizing some criterion. In both cases, the adaptive law
is updated not only to minimize the squared output error (as we usually do when we
use the gradient update law), but also minimize some other cost function of interest.

Then, we apply the presented algorithms (for both direct and indirect cases) to a
surge tank example.

Considering the discrete-time system presented in the Chapter 2, in Chapter 4
we present an algorithm to auto-tune the direction of descent in the update law for
both direct and indirect adaptive schemes. In this algorithm, the direction of descent
is obtained by minimizing the instantaneous control energy. We will also show that
 updating the adaptation gain (presented in Chapter 2 and found in [84, 85]) can be
viewed as a special case of updating the direction of descent. Finally, we will illustrate
the performance of this algorithm via a surge tank example.

Chapter 5 deals with a class of continuous-time nonlinear systems. In this chapter,
a gradient-based hybrid adaptive law is used for parameter adaptation. Using this
update law, we will establish some local results and prove boundedness of the control
and output variables. Then, we will present an algorithm to auto-tune the adaptation
gain of the hybrid adaptive law to minimize the instantaneous control energy. We will also implement this algorithm on a wing rock regulation example. Based on the results of this example, some comparisons will be made between direct and indirect adaptive control schemes.

Chapter 6 gives an overview of the results in the dissertation, and it gives a summary of the strengths and weaknesses of the presented algorithms. Finally, it provides an outline of possible future research directions.
CHAPTER 2

DISCRETE-TIME NONLINEAR ADAPTIVE CONTROL SYSTEMS

2.1 Introduction

In this chapter, we discuss indirect (following [43]) and direct adaptive control for a class of feedback linearizable discrete-time nonlinear systems. First, in Section 2.2 we start by describing the system we consider for control, along with its assumptions. Then, in Section 2.3 both direct and indirect adaptive control techniques are briefly discussed, and the output error equation is derived for both cases. In Section 2.4, the gradient update law used for parameter adaptation is derived. Then, in Section 2.5 we present the stability results for both direct and indirect schemes. Finally, Section 2.6 gives the concluding remarks.

2.2 Plant Description and Assumptions

Here, we start by describing the system we consider for control.
2.2.1 Plant Description

This section closely follows the development in [43]; hence we consider the single-input single-output discrete-time system described by

\[ y(k + 1) = f_0(x(k)) + g_0(x(k))u(k - d + 1) \]  

(2.1)

where \( f_0(\cdot) \) and \( g_0(\cdot) \) are unknown smooth functions, \( x(k) \) is a vector of past inputs and outputs \([y(k - n + 1), \ldots, y(k), u(k - m - d + 1), \ldots, u(k - d)]^\top\). where \( m \leq n \). \( y \) is the output, \( u \) is the input, and \( d \) is the time delay (relative degree) of the system. It has been shown in [43] that a state space representation of the system (2.1) can be written as

\[
\begin{align*}
x_i(k + 1) &= x_{i+1}(k), & \text{for } i &= 1, 2, \ldots, n - 1 \\
x_n(k + 1) &= f_0(x(k)) + g_0(x(k))x_{n+m+1}(k) \\
x_{n+i}(k + 1) &= x_{n+i+1}(k), & \text{for } i &= 1, 2, \ldots, m + d - 2 \\
x_{n+m+d-1}(k + 1) &= u(k) \\
y(k) &= x_n(k)
\end{align*}
\]  

(2.2)

where

\[
\begin{align*}
x_i(k) &= y(k - n + i), & \text{for } i &= 1, 2, \ldots, n \\
x_{n+i}(k) &= u(k - m - d + i), & \text{for } i &= 1, 2, \ldots, m + d - 1.
\end{align*}
\]

Note that when \( d = 1 \), the second equation of (2.2) becomes

\[ x_n(k + 1) = f_0(x(k)) + g_0(x(k))u(k) \]
so we can find a causal state feedback control to cancel the nonlinearity if it is known.

However, when \( d > 1 \), extra work is needed. Note that

\[
x_n(k + 2) = f_0(x(k + 1)) + g_0(x(k + 1))x_{n+m+1}(k + 1)
\]  

(2.3)

Replacing the \( x(k + 1) \) in (2.3) by the right hand side of (2.2), we get

\[
x_n(k + 2) = f_1(x(k)) + g_1(x(k))x_{n+m+2}(k)
\]  

(2.4)

Applying the same procedure, we get

\[
x_n(k + 3) = f_2(x(k)) + g_2(x(k))x_{n+m+3}(k)
\]

\[
\vdots
\]

\[
x_n(k + d - 1) = f_{d-2}(x(k)) + g_{d-2}(x(k))x_{n+m+d-1}(k).
\]

Consider the state transformation

\[
z(k) = \\
\begin{bmatrix}
z_{11}(k) \\
z_{12}(k) \\
\vdots \\
z_{1n}(k) \\
z_{1,n+1}(k) \\
z_{1,n+2}(k) \\
\vdots \\
z_{1,n+d-1}(k) \\
z_{21}(k) \\
z_{22}(k) \\
\vdots \\
z_{2m}(k) \\
z_{m+1}(k) \\
z_{m+2}(k) \\
\vdots \\
z_{n+m}(k)
\end{bmatrix} = \\
\begin{bmatrix}
x_1(k) \\
x_2(k) \\
\vdots \\
x_n(k) \\
x_{n+1}(k) \\
x_{n+2}(k) \\
\vdots \\
x_{n+k}(k) \\
x_{n+m}(k)
\end{bmatrix} = T(x(k)) = \\
\begin{bmatrix}
y(k - n + 1) \\
y(k - n + 2) \\
\vdots \\
y(k) \\
f_0(k) + g_0(k)u(k) \\
f_1(k) + g_1(k)u(k - d + 2) \\
\vdots \\
f_{d-2}(k) + g_{d-2}(k)u(k - 1) \\
u(k - m - d + 1) \\
u(k - m - d + 2) \\
\vdots \\
u(k - d)
\end{bmatrix}
\]

(2.5)

Assuming that \( g_0(x), g_1(x), \ldots, g_{d-2}(x) \) are bounded away from zero, then it can be shown that \( T(x(k)) \) is invertible (i.e., \( x = T^{-1}(z) \)). Applying the state transformation \( T(x(k)) \) to (2.2), in [43] the authors obtain

\[
z_{l1}(k + 1) = z_{l,i+1}(k), \quad \text{for } i = 1, 2, \ldots, n + d - 2
\]
\[ \begin{align*}
  z_{1,n+d-1}(k + 1) &= F(z(k)) + G(z(k))u(k) \\
  z_{2i}(k + 1) &= z_{2,i+1}(k), \quad \text{for } i = 1, 2, \ldots, m - 1 \\
  z_{2m}(k + 1) &= u(k - d + 1) \\
  y(k) &= z_{1n}(k) 
\end{align*} \]  

(2.6)

where

\[ \begin{align*}
  F(z(k)) &= f_{d-1}(x(k)) = f_{d-1}(T^{-1}(z(k))) \\
  G(z(k)) &= g_{d-1}(x(k)) = g_{d-1}(T^{-1}(z(k))) 
\end{align*} \]  

(2.7)

It is known that for the class of systems (2.1), there exists an ideal controller \((u^*(k))\) that drives the output of the system to track a known reference trajectory after \(d\) steps. Such a controller is defined as

\[ u^*(k) = \frac{r(k) - F(z(k))}{G(z(k))}. \]  

(2.8)

**Definition 1**: The zero dynamics [86] are defined as the unobservable dynamics when the control (2.8) is used; that is

\[ \begin{align*}
  z_{1i}(k + 1) &= z_{1,i+1}(k), \quad \text{for } i = 1, 2, \ldots, n - 2 \\
  z_{1,n-1}(k + 1) &= 0 \\
  z_{2i}(k + 1) &= z_{2,i+1}(k), \quad \text{for } i = 1, 2, \ldots, m - 1 \\
  z_{2m}(k + 1) &= \left. \frac{-f_0(T^{-1}(z(k)))}{g_0(T^{-1}(z(k)))} \right|_{z_{1i}=0,i=n-1,...,n+d-1} 
\end{align*} \]  

(2.9)

Hence, the zero dynamics can be defined [43] as the internal dynamics of the system when the reference command \(r(k)\) and the plant output \(y(k)\) are constrained to be identically zero. Since the dynamics associated with \(z_{1i}, i = 1, \ldots, n - 1\) are always
stable, the system is defined to be minimum phase if

\[
\begin{align*}
  z_{2i}(k + 1) &= z_{2,i+1}(k), & \text{for } i = 1, 2, \ldots, m - 1, \\
  z_{2m}(k + 1) &= \frac{-f_0(T^{-1}(0, z_2(k)))}{g_0(T^{-1}(0, z_2(k)))}
\end{align*}
\]

(2.10)

has an asymptotically stable equilibrium point, \( C = [c, c, \ldots, c]^T \).

This helps establish the class of systems we consider. Since we follow the development in [43] and hence use the same class of systems, the reader will be able to more readily identify the contribution of this work relative to that in [43] and the related works (e.g., in [87, 88, 89]).

### 2.2.2 Plant Assumptions

Here, we state our plant assumptions, which are the same as the ones used in [43].

**Assumption 1:** \( g_0(x), \ldots, g_{d-1}(x) \) are bounded away from zero over \( S_x \) (a known compact subset of \( \mathcal{R}^{n+m+d-1} \)), that is

\[
0 < \theta_0 \leq |g_i(x)|, \quad \forall x \in S_x.
\]

(2.11)

**Assumption 2:** (The Minimum Phase Assumption) The change of variables \( e_{2i} = z_{2i} - c \) transforms (2.10) into

\[
\begin{align*}
  e_{2i}(k + 1) &= e_{2,i+1}(k), & \text{for } i = 1, 2, \ldots, m - 1, \\
  e_{2m}(k + 1) &= \frac{-f_0(T^{-1}(0, z_2(k)))}{g_0(T^{-1}(0, z_2(k)))} - c.
\end{align*}
\]

(2.12)

It is assumed that the origin of the zero dynamics (2.12) is globally exponentially stable, and there is a Lyapunov function function \( V_2(e_2) \) that satisfies

\[
c_1|e_2(k)|^2 \leq V_2(e_2(k)) \leq c_2|e_2(k)|^2,
\]

28
\[ V_2(e_2(k + 1)) - V_2(e_2(k)) \leq -c_3|e_2(k)|^2. \]

and

\[ \left| \frac{\partial V_2(e_2)}{\partial e_2} \right| \leq L|e_2| \quad (2.13) \]

in some ball \( B_{q_2} \subset \mathbb{R}^m \) (where \( B_{q_2} = \{ e_2 \in \mathbb{R}^m | |e_2| < q_2 \} \), for some \( c_1, c_i, \) and \( c_3 > 0. \)

**Assumption 3:** Given a positive constant \( \varepsilon \) and a compact set \( S_x \subset \mathbb{R}^{n+m+d-1} \), there exists a parameter vector, \( A \), such that \( \hat{h} \) approximates a continuous function \( h \) with accuracy \( \varepsilon \) over \( S_x \), that is

\[ \max |\hat{h}(x, A) - h(x)| \leq \varepsilon, \quad \forall x \in S_x. \quad (2.14) \]

This assumption implies that there exists an approximator (which possesses the universal approximation property) that can approximate a continuous function to any desired degree of accuracy over \( S_x \). Here, a linear in the parameter Takagi-Sugeno fuzzy system (TSFS) is used for function approximation. Clearly linear in the parameter neural networks can also be used.

### 2.3 Direct and Indirect Adaptive Control

Here, we briefly discussed both direct and indirect adaptive control techniques and derive the output error equation for both cases.

#### 2.3.1 Direct Approach

A direct adaptive controller that seeks to drive the system to track a known reference input \( r(k) \) uses an approximator that attempts to approximate the ideal
controller dynamics \((u^*, \text{that we assume to exist})\). Let us consider the subclass of systems \((2.1)\) that can be written as \[y(k+d) = f_{d-1}(x(k)) - g_{d-1}(x(k))u(k).\] (2.15)

It is known that there exists a feedback control \((2.8)\) (that we attempt to approximate) that linearizes the system \((2.1)\) such that \(r(k)\) appears as the desired output \(d\) steps later. Assume that the control \((2.8)\) can be expressed as

\[u^*(k) = A_u^* \zeta(x(k), r(k)) + u_k(k) + \tilde{u}(k).\] (2.16)

where \(\tilde{u}(k)\) (which is a function of the approximation error) will be defined later, and

\[A_u^* = \arg \min_{A_u \in \Omega_u} \left[ \sup_{x \in S_x, r \in S_r} |A_u^T \zeta(x, r) - u_u(x, r)| \right].\] (2.17)

\(u_k(k)\) is the known part of the ideal control, \(\zeta(x, r)\) is the partial derivative of the approximator output with respect to the parameter vector, \(\Omega_u\) is a convex compact set containing all admissible approximator parameters, and \(S_r\) is the bounded space in which the reference input may vary. Note that \(\Omega_u\) may contain more than one \(A_u^*\) that best approximates \(u^*(k)\). We assume that “arg” in equation (2.17) simply picks one of these \(A_u^*\). The ideal control \(u^*(k)\) is unknown since the ideal parameter vector is not available. However, we assume that the ideal control can be approximated by a linear in the parameter approximator that can be written as

\[u(k) = A_u^T \zeta(x(k), r(k)) + u_k(k),\] (2.18)

where \(A_u(k)\) (an approximation of the ideal parameter vector \(A_u^*\)) will be updated online using a gradient method with projection to guarantee that \(A_u(k) \in \Omega_u\). Define the approximator parameter error as \(\phi(k) = A_u(k) - A_u^*\), it can be shown that the
output error dynamics can be expressed as

\[ e(k + 1) = -\theta(x(k - d + 1))\phi^T(k)\zeta(x(k - d + 1), r(k)) + \tilde{v}(k), \]  

(2.19)

where \( \theta(x(k)) = g_{d-1}(x(k)) \) and \( \tilde{v}(k) \) is function of the approximation error. Here, it is assumed that \( \theta(x(k)) \) is defined such that \( 0 < \theta_0 \leq \theta(x(k)) \leq \theta_1 \), where \( \theta_0 \) and \( \theta_1 \) are known constants related to the plant dynamics.

### 2.3.2 Indirect Approach

Unlike the direct approach, in the indirect approach we approximate the plant dynamics, then the feedback controller uses these estimates of the plant dynamics. As in the direct case, let us consider the subclass of systems (2.1) which can be written as

\[ y(k + d) = f_{d-1}(x(k)) + g_{d-1}(x(k))u(k) \]

\[ = f_u(x(k)) + f_k(x(k)) + [g_u(x(k)) + g_k(x(k))]u(k) \]

(2.20)

where \( f_k(\cdot) \) and \( g_k(\cdot) \) are the known parts of the dynamics, and \( f_u(\cdot) \) and \( g_u(\cdot) \) are the unknown parts of the dynamics (in what follows we can consider the case where \( f_k \equiv g_k \equiv 0 \)). We know (as we mentioned in the direct case) that there exists a state feedback controller

\[ u^*(k) = \frac{-f_{d-1}(x(k)) + r(k)}{g_{d-1}(x(k))}, \]

(2.21)

which linearizes the system (2.20) such that \( r(k) \) appears as the output after \( d \) steps. It is assumed that the unknown plant dynamics \( f_{d-1}(x(k)) \) and \( g_{d-1}(x(k)) \) can be estimated using approximators defined as

\[ f_u(x(k)) = A_f^T \zeta_f(x(k)) + \tilde{v}_f(k) \]

(2.22)

\[ g_u(x(k)) = A_g^T \zeta_g(x(k)) + \tilde{v}_g(k). \]

(2.23)
where $\tilde{u}_f(k)$ and $\tilde{u}_g(k)$ are functions of the approximation errors. The ideal parameter vectors are defined as

$$A_f^* = \arg \min_{A_f \in \Omega_f} \left[ \sup_{x \in S_x} |A_f^\top \zeta_f(x) - f_u(x)| \right]$$

(2.24)

and

$$A_g^* = \arg \min_{A_g \in \Omega_g} \left[ \sup_{x \in S_x} |A_g^\top \zeta_g(x) - g_u(x)| \right],$$

(2.25)

where $\Omega_f$ and $\Omega_g$ are the convex compact spaces containing the feasible parameter sets for $A_f^*$ and $A_g^*$, respectively.

Using the certainty equivalence approach, the control law is defined as

$$u(k) = \frac{-\hat{f}_{d-1}(x(k)) + r(k)}{\hat{g}_{d-1}(x(k))}.$$  

(2.26)

where $\hat{f}_{d-1}(x(k))$ and $\hat{g}_{d-1}(x(k))$ are estimates of $f_{d-1}(x(k))$ and $g_{d-1}(x(k))$, respectively, defined as

$$\hat{f}_{d-1}(x(k)) = A_f^\top \zeta_f(x(k)) + f_k(k),$$

(2.27)

$$\hat{g}_{d-1}(x(k)) = A_g^\top \zeta_g(x(k)) + g_k(k).$$

(2.28)

A projection algorithm may be used to ensure that $\hat{g}_{d-1}(x(k)) \geq \theta_0 > 0$ so that the control signal is well defined, and to ensure that $A_f(k) \in \Omega_f$ and $A_g(k) \in \Omega_g$ for all $k$. The parameter errors for the indirect adaptive controller are defined as $\phi_f(k) = A_f(k) - A_f^*$ and $\phi_g(k) = A_g(k) - A_g^*$. The output error for the indirect control system (as shown in [43]) can be expressed as

$$e(k+1) = \phi^\top(k)\zeta(x(k-d+1), u(k-d+1)) + \tilde{v}(k),$$

(2.29)

where $\phi = [\phi_f^\top, \phi_g^\top]^\top$, $A(k) = \left[ A_f^\top(k), A_g^\top(k) \right]^\top$, $\tilde{v}(k)$ is function of the approximation error, and $\zeta(x(k-d+1), u(k-d+1)) = \left[ \zeta_f^\top(x(k-d+1)), \zeta_g^\top(x(k-d+1)) u(k-d+1) \right]^\top$. 

32
In summary, the error equation for both direct and indirect cases can be written as

$$e(k + 1) = -\kappa \theta(x(k - d + 1))\phi^T(k)\zeta(x(k - d + 1)) + \bar{v}(k) \quad (2.30)$$

where $\theta(x(k - d + 1)) = 1$ in the indirect case, and $\kappa = 1$ and $-1$ in the direct and indirect cases, respectively. For simplicity, we will write (2.30) as

$$e(k + 1) = -\kappa \theta(k - d + 1)\phi^T(k)\zeta(k - d + 1) + \bar{v}(k) \quad (2.31)$$

In the next section, a brief description of the general gradient-based parameter update law for adaptive control will be given. Then, stability and parameter convergence issues will be discussed.

### 2.4 Gradient Method: Update and Stability

In this section, we derive a (normalized) gradient update law that seeks to minimize the squared tracking error. This section includes some of the adaptation and stability results for gradient update laws presented in [90].

Consider the cost function

$$J(A, \eta) = J_e(A) + J_u(\eta) \quad (2.32)$$

where $J_e(A) = e^2(k)$, $J_u(\eta) = u^2(k)$, $e(k)$ is the instantaneous tracking error, and $u(k)$ is the instantaneous control, and $\eta$ (as we will define later) is the adaptation gain. In this section we take the standard approach in adaptive control where we will only consider constructing the update routine based on minimizing the first part of the cost function (i.e., $J_e(A)$). Minimizing the second part of the cost function will be considered in later sections. The gradient update law is defined as

$$\Delta(k) = \Delta(k - 1) + \bar{\lambda}(k)D(k) \quad (2.33)$$
where $\bar{\lambda}(k) > 0$ is the step size, and $D(k)$ is the direction of descent. Since to achieve descent we want $\frac{\partial e}{\partial A} D(k) < 0$, one choice is to pick $D(k) = -\frac{\partial e}{\partial A}$. That is $D(k) = -2e(k) \frac{\partial \hat{e}(k)}{\partial A}$. Using the definition of output error (2.31) presented in the previous section, it can be shown that the direction of descent can be written as

$$D(k) = 2\kappa e(k) \theta(k - d) \zeta(k - d).$$

(2.34)

For this choice of the descent direction, along with picking the step size based on the normalized gradient approach, it can be shown that the parameter update law $A(k)$ can be written as

$$A(k) = A(k - 1) + \frac{\kappa \eta \zeta(k - d)}{1 + \gamma |\zeta(k - d)|^2} e(k)$$

(2.35)

where $\eta$ is the adaptation gain, and $\gamma$ is a positive constant ($\gamma > 0$). Here, $e(k)$ (which is the representation of the output error $e(k)$ in terms of a continuous dead zone of finite size $\epsilon > 0$) is defined as

$$e_i(e(k), \epsilon) = \begin{cases} 
  e(k) - \epsilon & \text{if } e(k) > \epsilon \\
  0 & \text{if } |e(k)| \leq \epsilon \\
  e(k) + \epsilon & \text{if } e(k) < -\epsilon 
\end{cases}$$

(2.36)

It is assumed that $\eta$ is chosen such that

$$0 < \eta < \frac{2\gamma}{\theta_1}.$$  

(2.37)

It is known that $A(k-1)$ is the old value of the parameter vector, and $\frac{1}{2(1+\gamma |\zeta(k-d)|^2)\theta(k-d)}$ and $\frac{1}{2(1+\gamma |\zeta(k-d)|^2)}$ are the choices of step size $\bar{\lambda}(k)$ that normalize the gradient for the direct and indirect cases respectively. From (2.31), $e(k)$ can be written as $e(k) = \bar{e}(k) + \tilde{v}(k)$, where $\bar{e}(k) = -\kappa \theta(k - d) \phi^\top(k - d) \zeta(k - d)$. It can be shown (according to a lemma that is presented in [90]) that if $e(k) = \bar{e}(k) + \tilde{v}(k)$, where $0 \leq |\tilde{v}(k)| < \epsilon$, and $\epsilon > 0$, then $e_i(e(k), \epsilon) = \pi(k) \bar{e}(k)$, where $0 \leq \pi(k) < 1$. Based on
this fact, $e_\epsilon(k)$ can be written as

$$
e_\epsilon(k) = e_\epsilon(\bar{e}(k), \epsilon) = \pi(k)\bar{e}(k) = -\kappa\pi(k)\theta(k-d)\phi^\top(k-d)\zeta(k-d). \quad (2.38)$$

In the next section, we will present some stability results for both direct and indirect cases by following the line of analysis presented in [43] for the indirect case.

### 2.5 Stability Analysis

In this section, we present stability and convergence results for the system (2.1) (for both direct and indirect cases) similar to the ones presented in [43] for the indirect case; note, however, that this is not the objective of the work in this document. Our objective, as we will show later, is to improve the closed-loop performance by auto-tuning some of the parameters in the adaptive law (e.g., the adaptation gain and the direction of descent). We include this section only for completeness and clarity, and since our results build directly on this theory. For detailed analysis about parts of the proof that we use from [43], the reader may need to refer to [43] whenever necessary. Next, stability and convergence results will be presented.

#### 2.5.1 Direct Adaptive Control

The authors in [43] have derived local convergence results for the indirect adaptive control case. Here, we will derive similar results for the direct case as well.

**Theorem 1:** Suppose $|r(k)| \leq d_1$ for all $k \geq 0$. Given any constant $\rho > 0$ and any small constant $\epsilon > 0$, there exist positive constants $\rho_1 = \rho_1(\rho, d_1)$, $\rho_2 = \rho_2(\rho, d_1)$,
\( \varepsilon^* = \varepsilon^*(q, \varepsilon, d_1) \), and \( \delta^* = \delta^*(q, \varepsilon, d_1) \) such that if Assumptions 1 and 3 are satisfied on \( S_x \supset B_{\varepsilon^*} \) with \( \varepsilon < \varepsilon^* \), Assumption 2 is satisfied on \( B_{\delta^*} \), \( |x(0)| \leq \theta \), and \( |\phi(0)| \leq \delta < \delta^* \), then using the direct adaptive control law (2.18), we will ensure that

1. \( |\phi(k)| \) will be monotonically nonincreasing, and \( |\phi(k) - \phi(k - 1)| \) will converge to zero.

2. The tracking error between the plant output and the reference command will converge to a ball of radius \( \varepsilon \) centered at the origin.

Proof: This proof proceeds in several steps in a fashion similar to the one presented in [43].

Step 1: We know that the dynamics of \( z_1 \) are

\[
    z_{1i}(k + 1) = z_{1,i+1}(k), \quad \text{for } i = 1, 2, \ldots, n + d - 2
\]

\[
    z_{1,n+d-i}(k + 1) = F(z(k)) + G(z(k))u(k)
\]

where \( u(k) \) (which is the approximate of the ideal control (2.8)) is defined by (2.18).

Adding and subtracting \( r(k) \) from the second equation of (2.39), we get

\[
    z_{1,n+d-i}(k + 1) = r(k) - [r(k) - F(z(k))] + G(z(k))u(k)
\]

Since \( u^* = \frac{r(k) - F(z(k))}{G(z(k))} \), (2.40) becomes

\[
    z_{1,n+d-i}(k + 1) = r(k) - \{G(z(k)) [u^* - u(k)]\}
\]

Define \( e_{1i}(k) = z_{1i}(k) - r(k - n - d + i) \), then (2.39) can be represented as

\[
    e_{1i}(k + 1) = e_{1,i+1}(k), \quad \text{for } i = 1, 2, \ldots, n + d - 2
\]

\[
    e_{1,n+d-i}(k + 1) = -\{G(z(k)) [u^* - u(k)]\}
\]
Note here that \( G(z(k)) = g_{d-1}(k) = \theta(x(k)) \), where \( 0 < \theta_0 \leq \theta(x(k)) \leq \theta_1 \). With the transformation \( e_{2i} = z_{2i} - c \), the dynamics of \( z_2 \) become

\[
e_{2i}(k + 1) = e_{1,i+1}(k), \quad \text{for } i = 1, 2, \ldots, m - 1
\]

\[
e_{2m}(k + 1) = u(k - d + 1) - c.
\] (2.43)

Equations (2.42) and (2.43) together form the new closed-loop state space representation. Let

\[
e_1(k) = [e_{11}(k), e_{12}(k), \ldots, e_{1n+d-1}(k)]^T.
\]

\[
e_2(k) = [e_{21}(k), e_{22}(k), \ldots, e_{2m}(k)]^T. \quad (2.44)
\]

and

\[
\Pi(k) = [r(k - n - d + 1), r(k - n - d + 2), \ldots, r(k - 1)]^T. \quad (2.45)
\]

**Step 2:** Consider the set

\[
I_e = \{ \begin{pmatrix} e_1 \\ e_2 \end{pmatrix} | |e_1| \leq \mu_1, |e_2| \leq \mu_2 \} \quad (2.46)
\]

where the positive constants \( \mu_1 \) and \( \mu_2 \) will be selected later. Let us start by picking them so that for all \( |x(0)| < \varrho \), \( e(0) \) will be in the interior of \( I_e \). We know that \( z(k) = e(k) + [\Pi(k), C]^T \) and \( x(k) = T^{-1}[z(k)] \). Hence, assuming that Assumptions 1 and 3 hold on a compact set \( S_x \) containing \( B_{\varrho_1} \), we conclude that for all \( e \in I_e \), \( x \) will remain inside a ball \( B_{\varrho_1} \), where \( \varrho_1 \) depends on \( \mu_1, \mu_2, d_1 \), and \( |C| \).

Consider the set

\[
I_A = \{ \phi | \phi \| \leq \delta \}. \quad (2.47)
\]
The objective of this step is show that if $e(k)$ remains in $I_e$, $I_\delta$ will be a positively invariant set when $\varepsilon$ and $\delta$ are sufficiently small. We have shown earlier that the output error can be defined as

$$e(k + 1) = \phi^T(k)\zeta(x(k - d + 1), r(k)) + \tilde{\nu}(k)$$

(2.48)

Since $x(k)$ is bounded, there exist $c_3$ and $c_4$ depending on $\mu_1$ and $\mu_2$ such that

$$|\tilde{\nu}(k)| \leq c_3|\phi|^2 + c_4\varepsilon.$$  

(2.49)

Assume that $\varepsilon$ and $\delta$ are sufficiently small such that

$$|\tilde{\nu}(k)| \leq M < \varepsilon$$  

(2.50)

where $\varepsilon$ is used in (2.36). Based on our definition of the parameter error ($\phi(k) = A_u(k) - A_u^*$), $\phi(k)$ can be expressed as

$$\phi(k) = \phi(k - 1) + \frac{\kappa_\eta\zeta(k - d)}{1 + \gamma|\zeta(k - d)|^2\varepsilon(k)} e(k).$$

(2.51)

Consider the Lyapunov-like function $V(k) = \phi^T(k)\phi(k)$. We will consider the case where $e(k)$ is within the dead zone separate from the case where $e(k)$ is outside the dead zone. First, consider the case where $e(k)$ is inside the dead zone. In this case, $e(k) = 0$ so $\phi(k) = \phi(k - 1)$ and $V(k) - V(k - 1) = 0$. With the error outside the dead zone, and for some $\eta$ and $0 < \pi(k) < 1$, we have

$$V(k) - V(k - 1) = \phi^T(k)\phi(k) - \phi^T(k - 1)\phi(k - 1)$$

$$= \left(\phi(k - 1) + \frac{\kappa_\eta\zeta(k - d)}{1 + \gamma|\zeta(k - d)|^2\varepsilon(k)} e(k)\right)^T$$

$$\left(\phi(k - 1) + \frac{\kappa_\eta\zeta(k - d)}{1 + \gamma|\zeta(k - d)|^2\varepsilon(k)} e(k)\right) - \phi^T(k - 1)\phi(k - 1)$$

38
so that with Equation (2.38)

\[ V(k) - V(k-1) = \eta \left[ \frac{-2}{\pi(k)\theta(k-d)} + \frac{\eta|\zeta(k-d)|^2}{1 + \gamma|\zeta(k-k)|^2} \right] \frac{e^2_k(k)}{1 + \gamma|\zeta(k-d)|^2} \] (2.52)

Since \( 0 < \pi(k) < 1 \), we get

\[ V(k) - V(k-1) \leq \eta \left[ \frac{-2}{\theta_1} + \frac{\eta|\zeta(k-d)|^2}{1 + \gamma|\zeta(k-d)|^2} \right] \frac{e^2_k(k)}{1 + \gamma|\zeta(k-d)|^2} \] (2.53)

For \((V(k) - V(k-1)) \leq 0\), we need

\[ 0 < \frac{2}{\theta_1} - \frac{\eta|\zeta(k-d)|^2}{1 + \gamma|\zeta(k-d)|^2} \] (2.54)

One way to ensure this is to pick \( \eta \) such that

\[ 0 < \eta < \frac{2\gamma}{\theta_1} = \frac{2\gamma|\zeta(k-d)|^2}{\theta_1|\zeta(k-d)|^2} < \frac{2[1 + \gamma|\zeta(k-d)|^2]}{\theta_1|\zeta(k-d)|^2} \] (2.55)

For this choice of the adaptation gain, we guarantee that \( V'(k) - V'(k-1) \leq 0 \). This shows that \( I_A \) is a positively invariant set.

Step 3: In this step, we study the stability of the dynamics associated with \( e_1 \). The dynamics of \( e_1 \) can be written as

\[ e_1(k+1) = A_1 e_1(k) + B_1 \{ \cdot \}_0 \] (2.56)

where \( A_1 \) and \( B_1 \) are equivalent to \( A \) and \( B \) in [43], respectively. For \( e \in I_e \), there exist constants \( c_5 \) and \( c_6 \) depending on \( \mu_1 \) and \( \mu_2 \) such that

\[ \{G(z(k))[u^* - u(k)]\} \leq \theta_1(c_5\delta + c_6\varepsilon) \] (2.57)

Proceeding with this step as in [43], we can show that given \( \mu_3 > 0 \), if \( \delta \) and \( \varepsilon \) are small enough such that

\[ \frac{c_6}{\lambda} \left[ \theta_1(c_5\delta + c_6\varepsilon) \right]^2 < \mu_3 \] (2.58)
where $c_9 > 0$ and $\lambda = \frac{\lambda_{\text{min}}(Q)}{2\lambda_{\text{max}}(P)}$ ($P$ and $Q$ are positive definite matrices used in the Lyapunov stability condition $A_1^TPA_1 - P = -Q$), then $\{V_1(e_1) = e_1^T(k)Pe_1(k) \leq \mu_3\}$ will be a positively invariant set. By choosing $\mu_3$ large enough so that $|e_1(0)| \leq \sqrt{\frac{\mu_3}{\lambda_{\text{max}}(P)}}$, we can be sure that $\{e_1 \in \mathcal{R}^{n+d-1} : V_1(e_1) \leq \mu_3\}$. Moreover, by choosing $\mu_1 \geq \sqrt{\frac{\mu_3}{\lambda_{\text{max}}(P)}}$, we can make $\{e_1 \in \mathcal{R}^{n+d-1} : \{V_1(e_1) \leq \mu_3\} \subset \{|e_1| \leq \mu_1\}\}$.

**Step 4:** In this step, we study the stability of the dynamics associated with $e_2$. Again, we will not repeat this step as well (we will only summarize the conclusion) since it is identical to the corresponding one in [43]. It can be shown that if $\mu_2$ is chosen large enough, there will be a positively invariant set $\{e_2 \in \mathcal{R}^m : \{V_2(e_2) \leq \mu_4\} \subset \{|e_2| \leq \mu_2\}\}$. Hence, by choosing $\mu_4$ large enough we can be sure that $\{e_2 \in \mathcal{R}^m : e_2(0) \in \{V_2(e_2) \leq \mu_4\}\}$ [43].

**Step 5:** As in [43], combine steps 2, 3, and 4 to conclude that as long $e(k)$ remains in $I_e$, the sets $I_A = \{\phi : \phi \leq \delta\}$, $\{e_1 \in \mathcal{R}^{n+d-1} : V_1(e_1) \leq \mu_3\}$, and $\{e_2 \in \mathcal{R}^m : V_2(e_2) \leq \mu_4\}$ are positively invariant sets for sufficiently small $\varepsilon$ and $\delta$. Since $e(0) \in \{V_1(e_1) \leq \mu_3\} \times \{V_2(e_2) \leq \mu_4\} \subset I_e$, $e(k)$ will remain in $I_e$ for all $k \geq 0$. This also implies that our conclusions are valid for all $k \geq 0$.

**Step 6:** Since $V(k) - V(k - 1) \leq 0$ is valid for all $k \geq 0$, we can say that $\phi^T(k)\phi(k)$ is monotonically nonincreasing and

$$\lim_{k \to \infty} \phi^T(k)\phi(k) = C_1,$$  \hspace{1cm} (2.59)

40
where \( C_1 \) is a constant. Using (5.74) in (2.51), it can be shown that
\[
\lim_{k \to \infty} \frac{\kappa \eta \zeta(k-d)}{1 + \gamma |\zeta(k-d)|^2} e_\epsilon(k) = 0.
\] (2.60)

Using (2.60) and (2.35), it can be shown that
\[
\lim_{k \to \infty} A(k) - A(k - 1) = 0.
\] (2.61)

This also implies that
\[
\lim_{k \to \infty} \sigma(k) - \sigma(k - 1) = 0.
\] (2.62)

establishing the first part part of the theorem. Also, the authors of [43] came up with another result showing that uniform boundedness of \( e(k) \) will ensure uniform boundedness of \( u(k) \).

**Step 7:** In this last step, we will guarantee convergence of the plant output error to an \( \epsilon \)-neighborhood of zero. Since \( x(k) \) and \( u(k) \) are both bounded, it can be shown that \( \zeta(k-d) \) is also bounded. Using this fact in (2.60) (and the fact that \( \eta \) is also bounded), we can show that
\[
\lim_{k \to \infty} e_\epsilon(k) = 0.
\] (2.63)

establishing the second part part of the theorem.

**2.5.2 Indirect Adaptive Control**

The authors in [43] have derived local convergence results for indirect adaptive control. Here, we will follow their proof.

**Theorem 2:** Suppose \( |r(k)| \leq d_1 \) for all \( k \geq 0 \). Given any constant \( \varrho > 0 \) and any small constant \( \epsilon > 0 \), there exist positive constants \( \varrho_1 = \varrho_1(\varrho, d_1) \), \( \varrho_2 = \varrho_2(\varrho, d_1) \),
\( \varepsilon^* = \varepsilon^*(q, \varepsilon, d_1), \) and \( \delta^* = \delta^*(q, \varepsilon, d_1) \) such that if Assumptions 1 and 3 are satisfied on \( S_x \supset B_{\delta_1} \) with \( \varepsilon < \varepsilon^* \), Assumption 2 is satisfied on \( B_{\varepsilon_2}, |x(0)| \leq \varepsilon, \) and \( |\phi(0)| \leq \delta \), then using the indirect adaptive control law (2.26), we will ensure that

1. \( |\phi(k)| \) will be monotonically nonincreasing, and \( |\phi(k) - \phi(k - 1)| \) will converge to zero.

2. The tracking error between the plant output and the reference command will converge to a ball of radius \( \varepsilon \) centered at the origin.

Proof: As in the direct case, this proof proceeds in several steps.

Step 1: Consider the the dynamics of \( z_1 \) expressed by (2.39). The second equation of (2.39) can be written [43] as

\[
\begin{align*}
    z_{1,n+d-1}(k+1) & = r(k) + \{\cdot\}_0, \\
    e_{1,n+d-1}(k+1) & = r(k) + \{\cdot\}_0
\end{align*}
\]

where \( \{\cdot\}_0 = \left[F(z(k)) - \hat{F}(z(k))\right] + \left[G(z(k)) - \hat{G}(z(k))\right] u(k) \) (here, \( \hat{F}(z(k)) \) and \( \hat{G}(z(k)) \) are estimates of the functions \( F(z(k)) \) and \( G(z(k)) \), respectively). Define \( e_{1i}(k) = z_{1i}(k) - r(k - n - d + i) \), then (2.39) can be represented (as in the direct case) by

\[
\begin{align*}
    e_{1i}(k+1) & = e_{1,i+1}(k), \quad \text{for } i = 1, 2, \ldots, n + d - 2 \\
    e_{1,n+d-1}(k+1) & = r(k) + \{\cdot\}_0
\end{align*}
\]

With the transformation \( e_{2i} = z_{2i} - c \), the dynamics of \( z_2 \) can be transformed into (2.43). As in the direct case, the vectors \( e_1(k) \) and \( e_1(k) \) are defined by (2.44), and \( \Pi(k) \) is defined by (2.45).
Step 2: Consider the sets $I_e$ and $I_A$ as defined in the direct case, where Assumptions 1 and 3 hold on a compact set $S_x$ containing $B_{g1}$. The objective of this step is to show that if $e(k)$ remains in $I_e$, $I_A$ will be a positively invariant set when $\varepsilon$ and $\delta$ are sufficiently small. We have shown earlier that the output error can be defined as

$$e(k + 1) = -\phi^T(k)\zeta(x(k - d + 1), r(k)) + \nu(k)$$  \hspace{1cm} (2.66)$$

Since $x(k)$ is bounded, there exist $c_3$ and $c_4$ depending on $\mu_1$ and $\mu_2$ such that

$$|\nu(k)| \leq c_3|\phi(k)|^2 + c_4\varepsilon.$$  \hspace{1cm} (2.67)$$

Assume that $\varepsilon$ and $\delta$ are sufficiently small such that

$$|\nu(k)| \leq M < \varepsilon$$  \hspace{1cm} (2.68)$$

where $\varepsilon$ is used in (2.36). Based on our definition of the parameter error ($\phi(k) = A_u(k) - A_u^*$), $\phi(k)$ can be expressed as

$$\phi(k) = \phi(k - 1) + \frac{\kappa_\eta\zeta(k - d)}{1 + \gamma|\zeta(k - d)|^2}e_\varepsilon(k).$$  \hspace{1cm} (2.69)$$

Consider the Lyapunov-like function $V(k) = \phi^T(k)\phi(k)$. We will consider the case where $e(k)$ is within the dead zone separate from the case where $e(k)$ is outside the dead zone. First, consider the case where $e(k)$ is inside the dead zone. Then $e_\varepsilon(k) = 0$ so $\phi(k) = \phi(k - 1)$ and $V(k) - V(k - 1) = 0$. With the error outside the dead zone, and for some $\eta$ and $0 < \pi(k) < 1$, we have

$$V(k) - V(k - 1) = \phi^T(k)\phi(k) - \phi^T(k - 1)\phi(k - 1)$$

$$= \left(\phi(k - 1) + \frac{\kappa_\eta\zeta(k - d)}{1 + \gamma|\zeta(k - d)|^2}e_\varepsilon(k)\right)^T$$

$$\left(\phi(k - 1) + \frac{\kappa_\eta\zeta(k - d)}{1 + \gamma|\zeta(k - d)|^2}e_\varepsilon(k)\right) - \phi^T(k - 1)\phi(k - 1)$$

43
so that with Equation (2.38)

\[ V(k) - V(k-1) = \eta \left[ \frac{-2}{\pi(k)} + \frac{\eta|\zeta(k-d)|^2}{1 + \gamma|\zeta(k-d)|^2} \right] \frac{e_i^2(k)}{1 + \gamma|\zeta(k-d)|^2} \]  

(2.70)

Since \( 0 < \pi(k) < 1 \), we get

\[ V(k) - V(k-1) \leq \eta \left[ -2 + \frac{\eta|\zeta(k-d)|^2}{1 + \gamma|\zeta(k-d)|^2} \right] \frac{e_i^2(k)}{1 + \gamma|\zeta(k-d)|^2} \]  

(2.71)

For \( (V(k) - V(k-1)) \leq 0 \), we need

\[ 0 < 2 - \frac{\eta|\zeta(k-d)|^2}{1 + \gamma|\zeta(k-d)|^2} \]  

(2.72)

One way to ensure this is to pick \( \eta \) such that

\[ 0 < \eta < 2\gamma = \frac{2\gamma|\zeta(k-d)|^2}{|\zeta(k-d)|^2} < \frac{2[1 + \gamma|\zeta(k-d)|^2]}{|\zeta(k-d)|^2} \]  

(2.73)

For this choice of the adaptation gain, we guarantee that \( V(k) - V(k-d) \leq 0 \). This shows that \( \theta \) is a positively invariant set.

**Steps 3, 4, and 5:**

Rewrite the dynamics of \( e_1 \) as

\[ e_1(k+1) = A_1 e_1(k) + B_1 \{ \cdot \}_0 \]  

(2.74)

where \( A_1 \) and \( B_1 \) are equivalent to \( A \) and \( B \) in [43], respectively. These three steps are identical to the one presented in [43], and have been summarized earlier in the direct case; hence we will not repeat for briefness.

**Steps 6 and 7**

These two steps are the same as the one presented in the direct case; hence we also will not repeat for briefness.

\[ \Box \]
2.6 Conclusions

In this chapter, we have discussed indirect and direct adaptive control for a class of feedback linearizable discrete-time nonlinear systems. We presented the stability results for both direct and indirect schemes. These results are considered local with respect to the parameters, but not with respect to the state. Note that our objective in this chapter was not to establish global stability results; we were mainly focused on establishing the basic foundation that will be used in the next two chapters to establish the auto-tuning results of the adaptation gain and the direction of descent.
3.1 Introduction

As with most gradient algorithms, the gradient update law presented in the previous chapter relies on the following idea. Starting with an initial value for the parameter vector, the gradient algorithm changes (updates) this vector by adding to it another vector having a magnitude (that depends on the adaptation gain and the magnitude of the output error) and a direction of descent. We can think of this as searching for the ideal parameter vector.

In [43, 90], the adaptation gain is a fixed parameter (selected \textit{a priori}). Here, however, we argue that the adaptation gain can be selected (adapted) on-line to minimize $J_u(\eta) = u^2(k)$ that is defined in the second part of the cost function (2.32). It is important to mention that our objective here is to search for an "optimal" $\eta(k)$ (that we will call $\eta^{\text{opt}}(k)$). This step is crucial to find the "optimal" parameter vector $A^{\text{opt}}(k)$, and hence what we will call the "optimal" control, $u^{\text{opt}}(k)$. The term \textit{optimal} is used here since the adaptation gain (as shown below) will be selected in two ways. In the first one, the adaptation gain is selected to minimize the instantaneous control
energy $J_u(\eta)$. In the second algorithm, the adaptation gain that is selected is the one that produces a control that is found by optimizing some criterion. Note that the optimization over an infinite horizon in the first case is infeasible since it is analytically intractable and computationally very expensive. In the second case, minimizing the control energy subject to a nonlinear uncertain system over an infinite horizon is very difficult, if not impossible.

This chapter is organized as follows. In Section 3.2, two algorithms that are used to auto-tune the adaptation gain for both direct and indirect adaptive control schemes will be presented. Then, in Section 3.3, two algorithms that are used to auto-tune the adaptation gain for both direct and indirect adaptive control schemes will be presented. In Section 3.4, simulation results will be presented. Finally, in Section 3.5 we provide some concluding remarks.

3.2 Direct Adaptive Control

Here, we present two algorithms to auto-tune the adaptation gain for direct adaptive control systems.

3.2.1 Algorithm 1

The adaptation gain tuning algorithm proceeds according to the following steps (shown in Figure 3.1):

1. Find a range on $\eta(k)$ (i.e., $\eta(k) \in [\eta_{\text{min}}(k), \eta_{\text{max}}(k)]$), such that the tracking error is forced to be within an $\epsilon$-neighborhood of zero no matter which $\eta(k)$ in this range is used.
2. Find the new adaptation gain ($\eta^{opt}(k)$) that minimizes the instantaneous control energy $J_u(k) = u^2(k)$.

3. Using $\eta^{opt}(k)$, find the new parameter vector $A^{opt}(k)$ and hence the new control $u^{opt}(k)$.

\[ J_u(k) = \sum u(k)^2 \]

\[ A^{opt}(k) = \begin{cases} A_u(k) & \text{if } \eta^{opt}(k) > 0 \\ A_u(k) - A^* & \text{if } \eta^{opt}(k) < 0 \end{cases} \]

\[ u^{opt}(k) = \frac{A^{opt}(k) - A_u(k)}{\eta^{opt}(k)} \]

\[ \eta^{opt}(k) = \min \{ \eta_{\min}, \max \{ \eta_{\min}, \eta(k), \eta_{\max} \} \} \]

\[ \text{Figure 3.1: Steps used for adaptation gain selection.} \]

**Finding a Feasible Range on $\eta(k)$**

Recall that the discrete-time parameter update law is given by (2.35). Based on our definition of the parameter error ($\phi(k) = A_u(k) - A^*$), $\phi(k)$ can be expressed as

\[ \phi(k) = \phi(k-1) + \frac{\kappa \eta(k) (k-d)}{1 + \gamma |\zeta(k-d)|^2} e_\varepsilon(k) \]

(3.1)

Consider the Lyapunov-like function $V(k) = \phi^T(k)\phi(k)$. We will consider the case where $e(k)$ is within the dead zone separate from the case where $e(k)$ is outside the dead zone. First, consider the case where $e(k)$ is inside the dead zone. Then $e_\varepsilon(k) = 0$ so $\phi(k) = \phi(k-1)$ according to (3.1) and $V(k) - V(k-1) = 0$. With the error outside the dead zone, and for some $\eta(k)$ and $0 < \pi(k) < 1$, we have

\[ V(k) - V(k-1) = \phi^T(k)\phi(k) - \phi^T(k-1)\phi(k-1) \]
\[
\begin{align*}
V(k) - V(k-1) &= \eta(k) \left[ \frac{-2}{\pi(k)\theta(k-d)} + \frac{\eta(k)|\zeta(k-d)|^2}{1 + \gamma|\zeta(k-d)|^2} \right] \frac{e_z^2(k)}{1 + \gamma|\zeta(k-d)|^2} \\
\end{align*}
\] (3.2)

Since \(0 < \pi(k) < 1\) and \(0 < \theta_0 \leq \theta(k) \leq \theta_1\), we get
\[
\begin{align*}
V(k) - V(k-1) &\leq \eta(k) \left[ \frac{-2}{\theta_1} + \frac{\eta(k)|\zeta(k-d)|^2}{1 + \gamma|\zeta(k-d)|^2} \right] \frac{e_z^2(k)}{1 + \gamma|\zeta(k-d)|^2} \\
\end{align*}
\] (3.3)

For \((V(k) - V(k-1)) \leq 0\), we need
\[
\begin{align*}
0 < \frac{2}{\theta_1} - \frac{\eta(k)|\zeta(k-d)|^2}{1 + \gamma|\zeta(k-d)|^2} \\
\end{align*}
\] (3.4)

One way to ensure this is to use \(\eta(k)\), where
\[
0 < \eta(k) < \frac{2 [1 + \gamma|\zeta(k-d)|^2]}{\theta_1|\zeta(k-d)|^2} = \bar{\eta}(k) \\
\] (3.5)

where \(\bar{\eta}(k)\) is defined as an upper bound on \(\eta(k)\). Define \(\eta(k) = \alpha(k)\bar{\eta}(k)\) and

\[
\eta_{\min}(k) = \alpha_1\bar{\eta}(k) \leq \eta(k) \leq \alpha_2\bar{\eta}(k) = \eta_{\max}(k), \\
\] (3.6)

where \(0 < \alpha_1 \leq \alpha(k) \leq \alpha_2 < 1\) for fixed constants \(\alpha_1\) and \(\alpha_2\). Later, we will define \(\alpha_1\) and \(\alpha_2\) in terms of known plant information. From the above analysis, stability (i.e., \((V(k) - V(k-1)) < 0\)) can be guaranteed, as shown in Theorem 4 below, for any choice of \(\alpha(k)\) such that inequality (3.6) holds.

Next, we study how to choose \(\alpha_1\) and \(\alpha_2\) such that \(\alpha(k)\) satisfies the above constraints. Note that if we can find such an \(\alpha(k)\), we will be able to derive a range of possible values of \(\eta(k)\) that ensures stability. Since \(\pi(k) < 1\), Equation (3.2) becomes

\[
\begin{align*}
V(k) - V(k-1) &\leq \eta(k) \left[ \frac{-2}{\theta(k-d)} + \frac{\eta(k)|\zeta(k-d)|^2}{1 + \gamma|\zeta(k-d)|^2} \right] \frac{e_z^2(k)}{1 + \gamma|\zeta(k-d)|^2} \\
\end{align*}
\] (3.7)
Suppose we want the decrease of \( (V'(A:) - V(k - 1)) \) to be influenced by a parameter \( \rho > 0 \) (\( \rho \) is a design parameter we can pick but below we will derive some constraints on its choice), then \( \eta(k) \) that meets this requirement can be expressed as

\[
0 < \rho = \frac{2}{\theta(k-d)} - \frac{\eta(k)|\zeta(k-d)|^2}{1 + \gamma|\zeta(k-d)|^2}
\]  
(3.8)

Then, \( \eta(k) \) can be written as

\[
\eta(k) = \frac{[2 - \rho \theta(k-d)][1 + \gamma|\zeta(k-d)|^2]}{\theta(k-d)|\zeta(k-d)|^2}
\]  
(3.9)

Since \( \eta(k) = \alpha(k)\bar{\eta}(k) \), then

\[
\alpha(k) = \frac{\theta_1[2 - \rho \theta(k-d)]}{2\theta(k-d)}
\]  
(3.10)

Notice that since \( 0 < \theta_0 \leq \theta(k) \leq \theta_1 \), the smallest \( \alpha(k) \) will be is \( \alpha_{\text{min}} \) where

\[
\alpha_{\text{min}} = \frac{\theta_1(2 - \rho \theta_1)}{2\theta_1} = 1 - \frac{1}{2}\rho \theta_1
\]  
(3.11)

We will pick \( \alpha_1 \) such that

\[
0 < \alpha_1 \leq 1 - \frac{1}{2}\rho \theta_1
\]  
(3.12)

and

\[
\rho < \frac{2}{\theta_1}
\]  
(3.13)

since we want to guarantee that \( \alpha_{\text{min}} > 0 \). Note also that the largest \( \alpha(k) \) will be is \( \alpha_{\text{max}} \) where

\[
\alpha_{\text{max}} = \frac{\theta_1(2 - \rho \theta_0)}{2\theta_0}
\]  
(3.14)

and we pick \( \alpha_2 \) such that \( \alpha_{\text{max}} \leq \alpha_2 < 1 \) so

\[
\frac{\theta_1(2 - \rho \theta_0)}{2\theta_0} \leq \alpha_2 < 1
\]  
(3.15)
Note that for the above choices, we know that $0 < \alpha_1 < \alpha_2 < 1$ because

$$1 > \alpha_2 \geq \frac{\theta_1(2 - \rho \theta_0)}{2\theta_0} > \frac{2 - \rho \theta_0}{2} > \frac{2 - \rho \theta_1}{2} \geq \alpha_1 > 0 \quad (3.16)$$

With this we know that there will be a range of possible $\alpha(k)$ values. At this stage of our analysis, it is important to study the feasibility of our choices of $\alpha_1$ and $\alpha_2$ (in terms of plant dynamics). One way to make such a study is to see how big $\alpha_{2\text{min}} - \alpha_{1\text{max}}$ is, where $\alpha_{2\text{min}}$ and $\alpha_{1\text{max}}$ are the minimum and maximum values that $\alpha_2$ and $\alpha_1$ can have.

$$\alpha_{2\text{min}} - \alpha_{1\text{max}} = \frac{\theta_1(2 - \rho \theta_0)}{2\theta_0} - \frac{\theta_0(2 - \rho \theta_1)}{2\theta_0} = \frac{\theta_1 - \theta_0}{\theta_0} \quad (3.17)$$

Therefore, for this approximation to apply, we need

$$0 < \frac{\theta_1 - \theta_0}{\theta_0} < 1 \quad (3.18)$$

which means that $0 < (\theta_1 - \theta_0) < \theta_0$ since $0 < \alpha_2 - \alpha_1 < 1$. Hence, this is very suitable for applications where we want to find as small of $\theta_1$ as possible and as big of $\theta_0$ as possible. Note that above we picked $\rho$ and found the resulting values of $\alpha_1$ and $\alpha_2$. We can also start by picking $\alpha_1$ and $\alpha_2$ such that

$$0 < \alpha_1 < \alpha_2 < 1 \quad (3.19)$$

with

$$\rho < \frac{2}{\theta_1} \quad (3.20)$$

then pick $\rho$ such that

$$2\alpha_1 \leq (2 - \rho \theta_1) \quad (3.21)$$

which can be written as

$$\rho \leq \frac{2 - 2\alpha_1}{\theta_1} \quad (3.22)$$
and also so that
\[ 2\alpha_2\theta_0 \geq \theta_1 (2 - \rho\theta_0) \]  \hspace{1cm} (3.23)

or equivalently
\[ \rho \geq \frac{2\theta_1 - 2\theta_0\alpha_2}{\theta_0\theta_1} \]  \hspace{1cm} (3.24)

Therefore, \( \rho \) should be chosen such that
\[ \frac{2\theta_1 - 2\theta_0\alpha_2}{\theta_0\theta_1} \leq \rho \leq \frac{2 - 2\alpha_1}{\theta_1} \]  \hspace{1cm} (3.25)

and also
\[ 0 < \rho < \frac{2}{\theta_1} \]  \hspace{1cm} (3.26)

Now, it is important to study the feasibility of our choice of \( \rho \). We can do that by finding the constraints that guarantee two things: First, the upper bound of \( \rho \) is always greater than the lower bound. Second, \( \rho > 0 \). To check the first condition, we want to make sure that equation (3.25) holds. That is,
\[
2\theta_1 - 2\theta_0\alpha_2 < 2\theta_0 - 2\theta_0\alpha_1 \\
2(\theta_1 - \theta_0) < 2\theta_0(\alpha_2 - \alpha_1) \\
\frac{\theta_1 - \theta_0}{\theta_0} < (\alpha_2 - \alpha_1)
\]

So it is clear that for the upper bound of \( \rho \) to be always greater than the lower bound, we need
\[ \frac{\theta_1}{\theta_0} - 1 < (\alpha_2 - \alpha_1) \]  \hspace{1cm} (3.27)

Also, to make sure that \( \rho > 0 \), it can be shown that we need the lower bound of equation (3.25) be greater than zero, or equivalently the following inequality be satisfied
\[ \alpha_2 < \frac{\theta_1}{\theta_0} \]  \hspace{1cm} (3.28)
Next, we will show that using the gradient update law (whose adaptation gain is adapted to satisfy the above requirements) along with the continuous dead zone, the output error is forced to stay within an $\epsilon$-neighborhood of zero.

**Finding the New Adaptation Gain $\eta^{opt}(k)$ via Minimizing the Instantaneous Control Energy**

Here, the new adaptation gain is obtained by minimizing the second part of the cost function (2.32) which can be expressed as

$$\min \quad J_u(\eta) = u^2(k)$$

such that $\eta_{\text{min}}(k) \leq \eta(k) \leq \eta_{\text{max}}(k)$. The control defined in (2.18) (assuming $u_k(k) = 0$, we have no prior information about the ideal control) can be written as

$$u(k) = A^T(k - 1)\zeta(k) + \frac{\kappa \eta \zeta(k - d)^T \zeta(k)}{1 + \gamma |\zeta(k - d)|^2} e_\epsilon(k).$$

(3.30)

Using Equation (3.30), it can be shown that $u^2(k)$ can be written as

$$u^2(k) = T_1(k)\eta^2(k) + T_2(k)\eta(k) + T_3(k)$$

(3.31)

where

$$T_1(k) = \frac{2 e_\epsilon^2(k) \left[ \zeta(k - d)^T \zeta(k) \right]^2}{\left[ 1 + \gamma |\zeta(k - d)|^2 \right]^2},$$

$$T_2(k) = \frac{2 A^T(k - 1)\zeta(k)e_\epsilon \zeta(k - d)^T \zeta(k)}{1 + \gamma |\zeta(k - d)|^2},$$

and

$$T_3(k) = \left[ A^T(k - 1)\zeta(k) \right]^2.$$

Since $T_3(k)$ is independent of $\eta(k)$ it can be omitted. Since $u^2(k)$ expressed in Equation (3.31) is in quadratic form, the cost function (3.29) can be minimized as a quadratic programming problem with linear inequality constraint ($\eta_{\text{min}}(k) \leq \eta(k) \leq \eta_{\text{max}}(k)$).
Since \( T_i(k) \) is positive definite, this problem is known to have a unique global minimum, \( \eta^{opt}(k) \), which is used to find the new parameter vector and hence the new control. Now, this adaptation gain can be used in the update routine of the controller’s parameter vector as shown next.

**Finding the New Parameter Vector \( A^{opt}(k) \) and the New Control \( u^{opt}(k) \)**

The new adaptation gain \( \eta^{opt}(k) \) can be used to find the new parameter vector \( A^{opt}(k) \) as follows

\[
A^{opt}(k) = A^{opt}(k - 1) + \frac{\eta^{opt}(k)\zeta(k - d)}{1 + \gamma|\zeta(k - d)|^2} e(k).
\]  

(3.32)

This new parameter vector of the controller is used to find the new control as

\[
u^{opt}(k) = A^{opt}(k)^T \zeta(k)
\]

(3.33)

which is the control to be input to the system.

It is important at this point to show that \( u^{opt}(k) \) (that is found using \( \eta^{opt}(k) \in [\eta_{min}, \eta_{max}] \)) lies inside the feasible control range \( [u_{min}(k), u_{max}(k)] \). This is shown in the next Theorem.

**Theorem 3:** Given that the new adaptation gain \( \eta^{opt}(k) \) is defined such that 0 < \( \eta_{min}(k) \leq \eta^{opt}(k) \leq \eta_{max}(k) \), the direct adaptive control law \( u^{opt}(k) \) that is obtained using \( \eta^{opt}(k) \) lies inside the feasible control range \( [u_{min}(k), u_{max}(k)] \).

**Proof:** Since the control is defined as \( u(k) = A^T(k)\zeta(k) \), it can be shown that the control can be written as

\[
u(k) = N_1(k) + \eta(k)N_2(k)
\]

(3.34)
where \( N_1(k) = A^\top(k-1)\zeta(k) \) and \( N_2(k) = \frac{\zeta(k-d)\zeta(k)e_s(k)}{1+\eta|\zeta(k-d)|^2} \). We know from the previous sections that \( \eta_{\min}(k) \) and \( \eta_{\max}(k) \) do not necessarily produce \( A_{\min}(k) \) and \( A_{\max}(k) \), respectively. Hence, \( u_{\max}(k) \) and \( u_{\min}(k) \) are defined as

\[
u_{\max}(k) = \begin{cases} N_1(k) + \eta_{\max}(k)N_2(k) & \text{if } \eta_{\max}(k) \text{ produces } A_{\max}(k) \\ N_1(k) + \eta_{\min}(k)N_2(k) & \text{if } \eta_{\min}(k) \text{ produces } A_{\max}(k) \end{cases}
\]

(3.35)

and

\[
u_{\min}(k) = \begin{cases} N_1(k) + \eta_{\min}(k)N_2(k) & \text{if } \eta_{\min}(k) \text{ produces } A_{\min}(k) \\ N_1(k) + \eta_{\max}(k)N_2(k) & \text{if } \eta_{\max}(k) \text{ produces } A_{\min}(k) \end{cases}
\]

(3.36)

It is clear at this point that we have two cases. In the first case, \( A_{\max}(k) \) and \( A_{\min}(k) \) are found using \( \eta_{\max}(k) \) and \( \eta_{\min}(k) \), respectively. In the second case, however, \( A_{\max}(k) \) and \( A_{\min}(k) \) are found using \( \eta_{\min}(k) \) and \( \eta_{\max}(k) \), respectively. It can be easily shown that \( N_2(k) \) is positive in the first case and negative in the second. To see this, note that in the first case we know that

\[
u_{\min} \leq \nu_{\max}
\]

(3.37)

or equivalently

\[
N_1(k) + \eta_{\min}(k)N_2(k) \leq N_1(k) + \eta_{\max}(k)N_2(k).
\]

(3.38)

Subtracting \( N_1(k) \) from both sides, we get

\[
\eta_{\min}(k)N_2(k) \leq \eta_{\max}(k)N_2(k).
\]

(3.39)

Hence, we have

\[
0 \leq (\eta_{\max}(k) - \eta_{\min}(k))N_2(k).
\]

(3.40)

which means that \( N_2(k) \) is positive, knowing (by definition) that \( \eta_{\max}(k) \geq \eta_{\min}(k) \).

In the second case, a similar way can be used to show that \( N_2(k) \) is negative.
Let us consider each case separately.

Case 1: \((N_2(k) > 0)\)

By definition, we know that

\[ u_{\min}(k) \leq u_{\max}(k) \]  (3.41)

or equivalently

\[ N_1(k) + \eta_{\min}(k)N_2(k) \leq N_1(k) + \eta_{\max}(k)N_2(k). \]  (3.42)

We are given that \(0 < \eta_{\min}(k) \leq \eta_{\text{opt}}(k) \leq \eta_{\max}(k)\). Since \(N_2(k) > 0\), we have

\[ \eta_{\min}(k)N_2(k) \leq \eta_{\text{opt}}(k)N_2(k) \leq \eta_{\max}(k)N_2(k). \]  (3.43)

Adding \(N_1(k)\) to the previous inequality, we get

\[ N_1(k) + \eta_{\min}(k)N_2(k) \leq N_1(k) + \eta_{\text{opt}}(k)N_2(k) \leq N_1(k) + \eta_{\max}(k)N_2(k) \]  (3.44)

or

\[ u_{\min}(k) \leq u_{\text{opt}}(k) \leq u_{\max}(k). \]  (3.45)

Now, consider the second case.

Case 2: \((N_2(k) < 0)\)

By definition, we know that

\[ u_{\min}(k) \leq u_{\max}(k) \]  (3.46)

or equivalently

\[ N_1(k) + \eta_{\max}(k)N_2(k) \leq N_1(k) + \eta_{\min}(k)N_2(k). \]  (3.47)
We are given that \( 0 < \eta_{\text{min}}(k) \leq \eta_{\text{opt}}(k) \leq \eta_{\text{max}}(k) \). Since \( N_2(k) < 0 \), we have

\[
\eta_{\text{min}}(k)N_2(k) \geq \eta_{\text{opt}}(k)N_2(k) \geq \eta_{\text{max}}(k)N_2(k).
\] (3.48)

Adding \( N_1(k) \) to the previous inequality, we get

\[
N_1(k) + \eta_{\text{min}}(k)N_2(k) \geq N_1(k) + \eta_{\text{opt}}(k)N_2(k) \geq N_1(k) + \eta_{\text{max}}(k)N_2(k)
\] (3.49)

or

\[
u_{\text{max}}(k) \geq \nu_{\text{opt}}(k) \geq \nu_{\text{min}}(k)
\] (3.50)

which completes the proof. ■

**Stability Analysis**

**Theorem 4:** Suppose \(|r(k)| \leq d_1 \) for all \( k \geq 0 \). Given any constant \( g > 0 \) and any small constant \( \epsilon > 0 \), there exist positive constants \( \varphi_1(g, d_1) \), \( \varphi_2(g, d_1) \), \( \varepsilon^* = \varepsilon^*(g, \epsilon, d_1) \), and \( \delta^* = \delta^*(g, \epsilon, d_1) \) such that if Assumptions 1 and 3 are satisfied on \( S_N \supset B_{\varphi_1} \) with \( \varepsilon < \varepsilon^* \), Assumption 2 is satisfied on \( B_{\varphi_2} \) with \( |x(0)| \leq \varphi \), and \( |\phi(0)| \leq \delta < \delta^* \), then using the direct adaptive control law (3.99) with the adaptation gain selected to satisfy (3.6), we will ensure that

1. \( |\phi(k)| \) will be monotonically nonincreasing, and \( |\phi(k) - \phi(k - 1)| \) will converge to zero.

2. The tracking error between the plant output and the reference command will converge to a ball of radius \( \epsilon \) centered at the origin.

**Proof:** This proof is similar to the proof of Theorem 1, except for the part of step 2 where we need to show that the Lyapunov-like function \( V(k) = \phi^T(k)\phi(k) \) is
monotonically nonincreasing. Using Equation (3.1) and recalling that
\[ \eta(k) = \alpha(k) \frac{2(1 + \gamma|z(k-d)|^2)}{\theta_1|z(k-d)|^2} \]  
where \( \alpha_1 \leq \alpha(k) \leq \alpha_2 \) and both \( \alpha_1 \) and \( \alpha_2 \) are chosen above, we get
\[ \phi(k) = \phi(k-1) + \frac{2\alpha(k)z(k-d)}{\theta_1|z(k-d)|^2} \epsilon_k(k). \]  
In the case where \( \epsilon_k(k) \) is inside the dead zone, \( \epsilon_k(k) = 0 \) so \( \phi(k) = \phi(k-1) \). Since the parameter error does not change from time \( (k-1) \), we have \( V(k) - V(k-1) = 0 \). However, in the case where \( \epsilon(k) \) is outside the dead zone, we can express \( (V(k) - V(k-1)) \) as
\[ V(k) - V(k-1) = \left[ \phi(k-1) + \frac{2\alpha(k)z(k-d)}{\theta_1|z(k-d)|^2} \epsilon_k(k) \right]^T \left[ \phi(k-1) + \frac{2\alpha(k)z(k-d)}{\theta_1|z(k-d)|^2} \epsilon_k(k) \right] - \phi^T(k-1) \phi(k-1) \]
\[ = \frac{-4\alpha^2(k)}{\theta_1|z(k-d)|^2} \epsilon_k^2(k) + \frac{4\alpha(k)\phi^T(k-1)z(k-d)}{\theta_1|z(k-d)|^2} \epsilon_k(k) \]
\[ = \frac{\epsilon_k^2(k)}{\theta_1|z(k-d)|^2} \left[ \frac{4\alpha^2(k)}{\theta_1} - \frac{4\alpha(k)}{\pi(k)\theta(k-d)} \right] \]  
Note that division by \( \epsilon_k(k) \) here is acceptable since the error is outside the dead zone (i.e., \( \epsilon_k(k) \) is bounded away from zero). So we have
\[ V(k) - V(k-1) = \frac{\epsilon_k^2(k)}{\theta_1|z(k-d)|^2} \left[ \frac{4\alpha^2(k)}{\theta_1} - \frac{4\alpha(k)}{\pi(k)\theta(k-d)} \right] \]  
since \( \epsilon_k(k) = -\pi(k)\theta(k-d)\phi^T(k)z(k-d) \). Since \( \pi(k) < 1 \) and \( \theta(k) \leq \theta_1 \), we can say that
\[ V(k) - V(k-1) < \frac{\epsilon_k^2(k)}{\theta_1|z(k-d)|^2} \left[ \frac{4\alpha^2(k)}{\theta_1} - \frac{4\alpha(k)}{\theta_1} \right] \]  
For stability, we want the bracketed term
\[ \frac{4\alpha(k)}{\theta_1} - \frac{4\alpha^2(k)}{\theta_1} > 0 \]
or equivalently
\[ \alpha(k) - \alpha^2(k) > 0 \]  
(3.57)

Note that this condition is feasible since for \(0 < \alpha(k) < 1\), the Inequality (3.57) is always satisfied. Hence, \(V(k) - V(k - 1) \leq 0, \forall k \geq 0\)

### 3.2.2 Algorithm 2

This adaptation gain tuning algorithm proceeds according to the following steps (shown in Figure 3.2):

1. Find a range on \(r(k)\) (i.e., scalars \(r_{\text{min}}(k)\) and \(r_{\text{max}}(k)\), so that \(r(k) \in [r_{\text{min}}(k), r_{\text{max}}(k)]\)) such that the tracking error is forced to be within an \(\epsilon\)-neighborhood of zero no matter which \(r(k)\) in this range is used (i.e., so stability is guaranteed regardless of what \(r(k)\) is used).

2. Using the upper and lower bounds on \(r(k)\), find element-wise upper and lower bounds on the parameter vector (i.e., \(A_{\text{min}}(k)\) and \(A_{\text{max}}(k)\)) and use these to find a range on the control \(u(k)\) (i.e., \(u(k) \in [u_{\text{min}}(k), u_{\text{max}}(k)]\)). This dynamic bound on the control represents a time-varying constraint on the values the input can take to still ensure stability.

3. Find a feasible control \((u^*(k))\) that optimizes some cost function. It is important to note that this \(u^*(k)\) is not the one to be input to the system.

4. Find the optimal \(\eta(k) (\eta^{\text{opt}}(k))\) that will give \(u^*(k)\). This \(\eta^{\text{opt}}(k)\) is the one used to update the parameter vector \(A^{\text{opt}}(k)\), which is used to find \(u^{\text{opt}}(k)\). This \(u^{\text{opt}}(k)\) is the one to be input to the system. It is important to note that \(u^*(k)\) may not be so accurate since it is found by optimizing a criterion defined by the
designer. Hence, we are no more confident about the optimal adaptation gain than the control $u^\eta(k)$.

5. Using the optimal adaptation gain $\eta^{opt}(k)$, find the optimal parameter vector $A^{opt}(k)$ and hence the optimal control $u^{opt}(k)$.

![Diagram](image.png)

Figure 3.2: Steps used for adaptation gain selection (Algorithm 2)

Next, the previously outlined algorithm will be discussed in detail. Note that the first and fifth steps of Algorithm 2 are similar to the first and third steps of Algorithm 1. Hence, we will not repeat these two steps here. Now, we will discuss the second, third, and fourth steps.

**Finding Upper and Lower Bounds on the Parameter Vector and Control**

Since we have upper and lower bounds on the adaptation gain $\eta(k)$, upper and lower bounds on the parameter vector can be easily obtained using (2.35). It is important to mention that $\eta_{min}$ (or $\eta_{max}$) does not necessarily produce $A_{min}$ (or $A_{max}$) respectively since $\zeta(k-d)e(k)$ can be positive or negative. Therefore, it is not easy at this stage of the algorithm to know which parameter vector is the upper or lower one. Let us assume for now that

$$A_1(k) = A(k - 1) + \frac{\eta_{min}(k)\zeta(k-d)}{1 + \gamma|\zeta(k-d)|^2}e(k)$$

(3.58)
and
\[ A_2(k) = A(k-1) + \frac{\eta_{\text{max}}(k)\zeta(k-d)}{1 + \gamma|\zeta(k-d)|^2}e_\varepsilon(k). \] (3.59)

Again, \( A_1(k) \) (or \( A_2(k) \)) is not necessarily the element-wise bound \( A_{\text{min}} \) (or \( A_{\text{max}} \)), respectively. However, this missing piece of information will be fixed next.

Each of the upper and lower bounds on the parameter vector \( A(k) \) gives a bound on the control. Let us say that
\[ u_1(k) = A_1^T(k)\zeta(k) \] (3.60)
and
\[ u_2(k) = A_2^T(k)\zeta(k) \] (3.61)
The upper and lower bounds on the control can be defined as
\[ u_{\text{max}}(k) = \max(u_1(k), u_2(k)) \] (3.62)
and
\[ u_{\text{min}}(k) = \min(u_1(k), u_2(k)) \] (3.63)
At this point, the upper and lower bounds on the parameter vector \( A(k) \) can be defined in a similar fashion.

\[ A_{\text{min}}(k) = \begin{cases} A_1(k) & \text{if } u_1(k) \leq u_2(k) \\ A_2(k) & \text{if } u_1(k) > u_2(k) \end{cases} \] (3.64)
and
\[ A_{\text{max}}(k) = \begin{cases} A_1(k) & \text{if } u_1(k) \geq u_2(k) \\ A_2(k) & \text{if } u_1(k) < u_2(k) \end{cases} \] (3.65)

Next, we show how the control bounds can be used as a time-varying control constraint (i.e., we pick \( u^n(k) \in [u_{\text{min}}, u_{\text{max}}] \)).
Finding a Control \(u^n(k)\)

Here, any control scheme (which is usually obtained to optimize some cost functions or specifications defined by the designer) can be used to find a control \(u^n(k)\). This control \(u^n(k)\) is used here only to find the optimal adaptation gain \(\eta^{opt}(k)\) as shown next.

Finding the Optimal Adaptation Gain \(\eta^{opt}(k)\)

We know that the control \(u(k)\) is defined in (2.18). The relationship between \(u^n(k)\) and \(\eta^{opt}(k)\) (assuming \(u_k(k) = 0\)) can be expressed in the following equation

\[
u^n(k) = A^T(k)\zeta(k) = A^T(k - 1)\zeta(k) + \frac{\eta(k)\epsilon(k)\zeta(k-d)\zeta(k)}{1 + \gamma|\zeta(k-d)|^2}.
\]

The adaptation gain \(\eta^{opt}(k)\) can be computed as

\[
\eta^{opt}(k) = \frac{(u^n(k) - A^T(k-1)\zeta(k))(1 + \gamma|\zeta(k-d)|^2)}{\epsilon(k)\zeta(k-d)\zeta(k)}.
\]

Since \(\epsilon(k)\) in the denominator of (3.67) can be zero inside the dead zone. \(\eta^{opt}(k)\) is defined as

\[
\eta^{opt}(k) = \begin{cases} 
\frac{|u^n(k) - A^T(k-1)\zeta(k)|(1 + \gamma|\zeta(k-d)|^2)}{\epsilon(k)\zeta(k-d)\zeta(k)} & \text{if } \epsilon(k) > \epsilon(k) \\
\eta_{min}(k) = \alpha \eta(k) & \text{if } \epsilon(k) \leq \epsilon(k)
\end{cases}
\]

Since the control used to calculate \(\eta^{opt}(k)\) is not accurate, the optimal adaptation gain \(\eta^{opt}(k)\) can be outside the desired dynamic range for \(\eta(k)\). Therefore, the following method is used to ensure that \(\eta^{opt}(k)\) always lies inside our desired range as

\[
\eta^{opt}(k) = \begin{cases} 
\eta_{min}(k) & \text{if } \eta^{opt}(k) < \eta_{min}(k) \\
\eta_{max}(k) & \text{if } \eta^{opt}(k) > \eta_{max}(k) \\
\eta^{opt}(k) & \text{otherwise}
\end{cases}
\]

Now, this adaptation gain can be used in the update routine of the controller's parameter vector as shown next.
The last step is similar to the last step of Algorithm 1. It is important at this point to show that \( u^{\text{opt}}(k) \) (that is found using \( \eta^{\text{opt}}(k) \in [\eta_{\text{min}}, \eta_{\text{max}}] \)) lies inside the feasible control range \([u_{\text{min}}(k), u_{\text{max}}(k)]\). This is shown in Theorem 3.

### 3.3 Indirect Adaptive Control

Here, we present two algorithms to auto-tune the adaptation gain for indirect adaptive control systems.

#### 3.3.1 Algorithm 1

The general steps in this adaptation gain tuning algorithm are the same as the corresponding one outlined in the direct case; however, there are some key differences in the stability conditions and the determination of the control law. Let us start by discussing the steps of the algorithm which are summarized in Figure 3.1.

**Finding a Feasible Range on \( \eta(k) \)**

Based on our definition of the parameter error \( \phi(k) = A_{u}(k) - A_{u}^{*} \), \( \phi(k) \) can be expressed as

\[
\phi(k) = \phi(k - 1) + \frac{\kappa \eta(k) \zeta(k - d)}{1 + \gamma|\zeta(k - d)|^{2}} e(k)
\]  

(3.70)

Consider the Lyapunov-like function \( V'(k) = \phi^{T}(k)\phi(k) \). The case where \( e(k) \) is within the dead zone is the same as before. With error outside the dead zone, and for some \( \eta(k) \) and \( 0 < \pi(k) < 1 \), we have

\[
V'(k) - V'(k - 1) = \phi^{T}(k)\phi(k) - \phi^{T}(k - 1)\phi(k - 1)
\]

\[
= \left( \phi(k - 1) + \frac{\kappa \eta(k) \zeta(k - d)}{1 + \gamma|\zeta(k - d)|^{2}} e(k) \right)^{T}
\]

\[
\frac{\kappa \eta(k) \zeta(k - d)}{1 + \gamma|\zeta(k - d)|^{2}} e(k)
\]

\[
- \phi^{T}(k - 1)\phi(k - 1)
\]

63
so that with Equation (2.38), we have

\[ V'(k) - V'(k - 1) = \eta(k) \left[ \frac{-2}{\pi(k)} + \frac{\kappa^2\eta(k)|\zeta(k-d)|^2}{1 + \gamma|\zeta(k-d)|^2} \right] \frac{e^2(k)}{1 + \gamma|\zeta(k-d)|^2} \]  

(3.71)

Since \( 0 < \pi(k) < 1 \) and \( \kappa^2 = 1 \) (since \( \kappa = -1 \) in this case), we get

\[ V'(k) - V'(k - 1) \leq \eta(k) \left[ -2 + \frac{\eta(k)|\zeta(k-d)|^2}{1 + \gamma|\zeta(k-d)|^2} \right] \frac{e^2(k)}{1 + \gamma|\zeta(k-d)|^2} \]  

(3.72)

Note that Equation (3.72) is similar to Equation (3.3), except that \( \theta_1 = 1 \) (since in this case \( \theta(k) = 1 \)). It can be shown that \( \eta(k) \) can be defined the same way as in (3.6), where \( \bar{\eta}(k) \) is defined as

\[ \bar{\eta}(k) = \frac{2[1 + \gamma|\zeta(k-d)|^2]}{|\zeta(k-d)|^2} \]  

(3.73)

From the above analysis, stability can be guaranteed as we will show in Theorem 6 below, for any choice of \( \alpha(k) \) such that inequality (3.6) holds. Also, it can be shown that the parameter \( \rho \) can be expressed as

\[ 0 < \rho = 2 - \frac{\eta(k)|\zeta(k-d)|^2}{1 + \gamma|\zeta(k-d)|^2} \]  

(3.74)

Then, \( \eta(k) \) can be written as

\[ \eta(k) = \frac{[2 - \rho][1 + \gamma|\zeta(k-d)|^2]}{|\zeta(k-d)|^2} \]  

(3.75)

Since \( \eta(k) = \alpha(k)\bar{\eta}(k) \), then

\[ \alpha(k) = \frac{2 - \rho}{2} \]  

(3.76)

To ensure that \( 0 < \alpha(k) < 1 \), the rate of decrease has to be bounded within \( 0 < \rho < 2 \).

It is clear from this discussion that unlike the direct case, the choice of the bounds of \( \alpha \) in the indirect case is independent of the plant dynamics. However, no matter what choice we make in selecting the bounds of \( \alpha \) (as long as \( 0 < \alpha_1 < \alpha_2 < 1 \)), the a
rate of decrease of $V(k) - V(k - 1)$ is confined in the interval $(0, 2)$ (i.e., $0 < \rho < 2$).

These conclusions represent the major differences between the two cases.

Notice that since $|\zeta(k)|^2$ can be very small (as in Takagi-Sugeno fuzzy systems), it is important to assume that we know a lower bound on $|\zeta(k)|^2$ (i.e., $|\zeta(k)|^2 \geq \bar{\zeta}$). Such a lower bound on $|\zeta(k)|^2$ provides an upper bound on the adaptation gain. It is known that the maximum possible adaptation gain is given by

$$\tilde{\eta}(k) = 2 \left[ \frac{1}{|\zeta(k-d)|^2} + \gamma \right]. \quad (3.77)$$

Using Equation (3.77) and the assumption that $|\zeta(k)|^2 \geq \bar{\zeta}$, then the upper bound on $\tilde{\eta}(k)$ becomes

$$\tilde{\eta}(k) \leq 2 \left[ \frac{1}{\bar{\zeta}} + \gamma \right] \quad (3.78)$$

This can be guaranteed by letting

$$\tilde{\eta}(k) = \begin{cases} 2 \left[ \frac{1}{\bar{\zeta}} + \gamma \right] & \text{if } \tilde{\eta}(k) \geq 2 \left[ \frac{1}{\bar{\zeta}} + \gamma \right] \\ 2 \left[ \frac{1}{|\zeta(k-d)|^2} + \gamma \right] & \text{if } \tilde{\eta}(k) < 2 \left[ \frac{1}{\bar{\zeta}} + \gamma \right] \end{cases} \quad (3.79)$$

This upper bound on the maximum adaptation gain can be used to ensure boundedness of the new adaptation gain, $\eta^{opt}(k)$.

Next, we will show how to select the adaptation gain to minimize the instantaneous control energy.

**Finding the New Adaptation Gain ($\eta^{opt}(k)$)**

Since $\kappa = -1$ in the indirect case, the instantaneous control becomes

$$u(k) = A^\top(k - 1)\zeta(k) - \frac{\eta(k)\zeta(k-d)}{1 + \gamma|\zeta(k-d)|^2} e(k). \quad (3.80)$$

and hence $u^2(k)$ becomes

$$u^2(k) = T_1(k)\eta^2(k) + T_2(k)\eta(k) + T_3(k) \quad (3.81)$$

65
where
\[
T_1(k) = \frac{2e^2(k) [\zeta(k-d) \zeta(k)]^2}{[1 + \gamma|\zeta(k-d)|^2]^2},
\]
\[
T_2(k) = \frac{-2A^T(k-1)\zeta(k)e_\varepsilon(k-d) \zeta(k)}{1 + \gamma|\zeta(k-d)|^2},
\]
and
\[
T_3(k) = \left[ A^T(k-1)\zeta(k) \right]^2.
\]

Since \( T_3(k) \) in independent of \( \eta(k) \) it can be omitted: hence, the cost function can be solved as quadratic programming problem with linear inequality constraints which is known to have a unique optimal solution since \( T_1(k) \) is positive definite. Once the new adaptation gain is found, the new control is calculated as we will show next.

**Finding the New Parameter Vector \((A^{opt}(k))\) and the New Control \((u^{opt}(k))\)**

The new adaptation gain \( \eta^{opt}(k) \) can be used to find the new parameter vector \( A^{opt}(k) \) as follows
\[
A^{opt}(k) = A^{opt}(k-1) - \frac{\eta^{opt}(k)\zeta(k-d)}{1 + \gamma|\zeta(k-d)|^2} e_\varepsilon(k).
\]
(3.82)

The new parameter vector can be used to find approximation of the plant dynamics as follows
\[
A^{opt}(k) = A^{opt}^T(k)\zeta_{\alpha}(k)
\]
(3.83)

and
\[
A^{opt}(k) = A^{opt}^T(k)\zeta_{\beta}(k).
\]
(3.84)

Now, the new control to be input to the system can be easily computed by
\[
u^{opt}(k) = \frac{-A^{opt}(k) + r(k)}{\beta^{opt}(k)}
\]
(3.85)
It is important at this point to show that \( u_{\text{opt}}(k) \) (that is found using \( \eta_{\text{opt}}(k) \in [\eta_{\text{min}}, \eta_{\text{max}}] \)) lies inside the feasible control range \([u_{\text{min}}(k), u_{\text{max}}(k)]\).

**Theorem 5:** Given that the new adaptation gain \( \eta_{\text{opt}}(k) \) is defined such that \( 0 < \eta_{\text{min}}(k) \leq \eta_{\text{opt}}(k) \leq \eta_{\text{max}}(k) \), the indirect adaptive control law \( u_{\text{opt}}(k) \) that is obtained using \( \eta_{\text{opt}}(k) \) lies inside the feasible control range \([u_{\text{min}}(k), u_{\text{max}}(k)]\).

**Proof:** The proof is the same as the one used to prove Theorem 3. ■

**Stability Analysis**

Here, stability and convergence results will be presented.

**Theorem 6:** Suppose \( |r(k)| \leq d_1 \) for all \( k \geq 0 \). Given any constant \( \varrho > 0 \) and any small constant \( \varepsilon > 0 \), there exist positive constants \( \varrho_1 = \varrho_1(\varrho, d_1) \), \( \varrho_2 = \varrho_2(\varrho, d_1) \), \( \varepsilon^* = \varepsilon^*(\varrho, \varepsilon, d_1) \), and \( \delta^* = \delta^*(\varrho, \varepsilon, d_1) \) such that if Assumptions 1 and 3 are satisfied on \( S_2 \subseteq B_{\varrho_1} \) with \( \varepsilon < \varepsilon^* \), Assumption 2 is satisfied on \( B_{\varrho_2} \), \( |x(0)| \leq \varrho \), and \( |\phi(0)| \leq \delta < \delta^* \), then using the indirect adaptive control law \( (3.85) \) with the adaptation gain selected to satisfy \( (3.6) \), we will ensure that

1. \( |\phi(k)| \) will be monotonically nonincreasing, and \( |\phi(k) - \phi(k-1)| \) will converge to zero.

2. The tracking error between the plant output and the reference command will converge to a ball of radius \( \varepsilon \) centered at the origin.

**Proof:** The proof of this theorem is similar to the proof of Theorem 2, except for the part in Step 2 where we need to show that the Lyapunov-like function \( V(k) = \)
\( \phi^T(k)\phi(k) \) is monotonically nonincreasing. From equation (3.1) and recalling that

\[
\eta(k) = \alpha(k) \frac{2(1 + \gamma\zeta(k - d)^2)}{\zeta(k - d)^2} \tag{3.86}
\]

where \( \alpha_1 \leq \alpha(k) \leq \alpha_2 \) and both \( \alpha_1 \) and \( \alpha_2 \) are chosen above, we get

\[
\phi(k) = \phi(k - 1) - \frac{2\alpha(k)\zeta(k - d)}{\zeta(k - d)^2} e_\varepsilon(k). \tag{3.87}
\]

It is clear that Equation (3.87) is a special case of Equation (3.52), where \( \theta_1 = 1 \) and \( \kappa = -1 \). It can be easily shown (as in the direct case) that \( V(k) - V(k - 1) \leq 0, \forall k \geq 0 \).

### 3.3.2 Algorithm 2

The general steps in this adaptation gain tuning algorithm (that are shown in Figure 3.2) are the same as the corresponding one outlined in the direct case; however, there are some key differences in the stability conditions and details of how the control is determined by the algorithm.

Next, the previously outlined algorithm will be discussed in detail. Note that the first and fifth steps of Algorithm 2 are similar to the first and third steps of Algorithm 1. Also, step 3 is similar to the corresponding one in the direct case. Hence, we will not repeat these steps here. Now, we will discuss the second and fourth steps.

**Finding Upper and Lower Bounds on the Parameter Vector and Control**

Since we have upper and lower bounds on the adaptation gain \( \eta(k) \), upper and lower bounds on the parameter vector can be easily obtained using (2.35) as in the direct case where

\[
A_1(k) = A(k - 1) - \frac{\eta_{\text{min}}(k)\zeta(k - d)}{1 + \gamma\zeta(k - d)^2} e_\varepsilon(k) \tag{3.88}
\]
and

\[ A_2(k) = A(k - 1) - \frac{\eta_{max}(k) \zeta(k - d)}{1 + \gamma |\zeta(k - d)|^2} e_\varepsilon(k). \]  

(3.89)

This bound on the parameter vector will be used to find a feasible control range.

Since the parameter vectors are defined as \( A_i(k) = [A_{\alpha_i}(k), A_{\beta_i}(k)]^T \) for \( i = 1, 2 \) and \( \zeta(k) = [\zeta_{\alpha}(k), \zeta_{\beta}(k) u(k)]^T \), the plant dynamics can be approximated by \( \alpha_i(k) = A_{\alpha_i}(k) \zeta_{\alpha}(k) \) and \( \beta_i(k) = A_{\beta_i}(k) \zeta_{\beta}(k) \), for \( i = 1, 2 \).

Using these approximation of plant dynamics, bounds on the control can be calculated as

\[ u_1(k) = \frac{-\alpha_1(k) + r(k)}{\beta_1(k)} \]  

(3.90)

and

\[ u_2(k) = \frac{-\alpha_2(k) + r(k)}{\beta_2(k)} \]  

(3.91)

The upper and lower bounds on the control can be defined as

\[ u_{max}(k) = \max (u_1(k), u_2(k)) \]  

(3.92)

and

\[ u_{min}(k) = \min (u_1(k), u_2(k)) \]  

(3.93)

At this point, the upper and lower bounds on the parameter vector \( A(k) \) can be defined in a similar fashion.

\[ A_{min}(k) = \begin{cases} A_1(k) & \text{if } u_1(k) \leq u_2(k) \\ A_2(k) & \text{if } u_1(k) > u_2(k) \end{cases} \]  

(3.94)

and

\[ A_{max}(k) = \begin{cases} A_1(k) & \text{if } u_1(k) \geq u_2(k) \\ A_2(k) & \text{if } u_1(k) < u_2(k) \end{cases} \]  

(3.95)

These control bounds can be used as a time-varying control constraint. These control bounds can be used in step 3 to find some control within the feasible control range.
Then, the this control can be used to find the new adaptation gain as discussed in step 4. Then, this adaptation gain can be used to find the new parameter vector and new control as shown next.

**Finding the Optimal Parameter Vector \( \alpha_{\text{opt}}(k) \) and the Optimal Control \( u_{\text{opt}}(k) \)**

The optimal adaptation gain \( \eta_{\text{opt}}(k) \) can be used to find the optimal parameter vector \( \alpha_{\text{opt}}(k) \) as follows

\[
\alpha_{\text{opt}}(k) = \alpha_{\text{opt}}(k - 1) - \frac{\eta_{\text{opt}}(k) \zeta(k - d)}{1 + \gamma \zeta(k - d)^2} e_\varepsilon(k).
\]  
(3.96)

This optimal parameter vector can be used to find approximation of the plant dynamics as follows

\[
\alpha_{\text{opt}}(k) = \alpha_{\text{opt}}^T(k) \zeta_\alpha(k)
\]
(3.97)

and

\[
\beta_{\text{opt}}(k) = \beta_{\text{opt}}^T(k) \zeta_\beta(k).
\]
(3.98)

Now, the control to be input to the system can be easily computed by

\[
u_{\text{opt}}(k) = \frac{-\alpha_{\text{opt}}(k) + r(k)}{J_{\text{opt}}(k)}
\]
(3.99)

Note here that it may be that \( u^0 \neq u_{\text{opt}} \).

As shown in the direct case, it can be easily verified that using the optimal adaptation gain, the optimal control lies inside the feasible control range.

**3.4 Simulation Results Using a Surge Tank Example**

Consider the surge tank model (taken from [91]) that can be represented by the following differential equation:

\[
\frac{dh(t)}{dt} = \frac{-c \sqrt{2gh(t)}}{A(h(t))} + \frac{1}{A(h(t))} u(t)
\]  
(3.100)
where \( u(t) \) is the input flow (control input), which can be positive or negative. Also, \( h(t) \) is the liquid level (output of the system); \( A(h(t)) \) is an unknown cross-sectional area of the tank; \( g = 9.8 \text{m/sec}^2 \) is the gravitational acceleration; and \( c = 1 \) is the known cross-sectional area of the output pipe. Let \( A(h(t)) = \sqrt{a h(t) + b} \), where \( a = 1 \) and \( b = 3 \).

Using Euler approximation to discretize the system, we have

\[
h(k + 1) = h(k) + T \left[ -\sqrt{19.6 h(k)} + \frac{u(k)}{A(h(k))} \right] \tag{3.101}
\]

where \( T = 0.1 \). We will simulate the system for \( h(k) > 0 \) so that the simulation is realistic.

**Direct Case**

The direct fuzzy controller \( u(k) = A h(k), r(k + 1) \) used here is a TSFS that has two inputs, the reference input (which is a square wave whose upper and lower values are 1.5 and 3) and error, \( e(k) = r(k) - h(k) \). Five Gaussian membership functions are used for each input universe of discourse. For the first input (reference input), the centers of the membership functions are distributed evenly between 0 and 5, and the centers for the second input (error) are distributed evenly between -5 and 5. The choices of the parameters used in the update routine are made based on the following reasoning. Since the tank equation (4.37) can be written as \( h(k + 1) = \alpha(h(k)) + \beta(h(k)) u(k) \), the error dynamics can be expressed as in Equation (2.31) where

\[
\theta(h(k)) = \beta(h(k)) = \left[ \frac{T}{A(h(k))} \right]
\]

and

\[
\nu(k) = 0
\]
Since $\theta(k)$ must satisfy

$$0 < \theta_0 \leq \theta(x(k)) \leq \theta_1$$

and assuming that $0 < h(k) \leq 7$, it can be shown that the following relation must hold

$$0.0316 \leq \theta(x(k)) \leq 0.0577$$

so let us pick $\theta_0$ and $\theta_1$ to be 0.033 and 0.05, respectively. It is easy to verify that this choice satisfies (3.18). Also, based on (3.27), $\alpha_2 - \alpha_1$ must be greater than 0.5152 so we can pick $\alpha_1$ and $\alpha_2$ to be 0.1 and 0.95, respectively. From (3.26), we know that $\rho$ has to be less than 40 and also satisfy (3.25) (which says that $\rho$ must lie in the closed interval $[22.6061, 36]$) so one reasonable choice is to pick $\rho = 30$.

**Algorithm 1**

Here, the adaptation gain is found by minimizing the instantaneous control energy. The closed-loop response of the overall system is shown in Figure 3.3. The first plot shows a comparison between the plant's output and the desired square wave reference trajectory. The second and third plots in the figure show the output error $e(k)$ and the optimal control $u^{opt}(k)$, respectively. A plot of the optimal adaptation gain $\eta^{opt}(k)$ (compared to the upper and lower bounds) is shown in Figure 3.4.

**Algorithm 2**

Since a control $u^\pi$ (that optimized some cost function) is needed to find $\eta^{opt}$, the model predictive control (MPC) algorithm (presented and analyzed in [83, 67, 68, 69, 72]...
Figure 3.3: Optimal adaptive direct fuzzy controller for the surge tank (Algorithm 1).

70, 71, 51]) is used. MPC is a model-based control scheme, so a model of the system is needed in the design. The model used here is an affine model that can be obtained by linearizing the plant (4.37) at $h = 3$. The affine model is found to be

$$h(k + 1) = 0.9065h(k) + 0.0447u(k) - 0.0624$$  \hspace{1cm} (3.102)

where a prediction horizon of length 5 is used, and the objective function weighting matrices $Q$ and $R$ are identity matrices (i.e., positive definite diagonal matrices). Again, the control obtained from MPC is not to be input to the system: it is just used to calculate the optimal adaptation gain $\eta^{opt}(k)$. Note that we are not very confident in the control ($u^*$) found by MPC since it is solved subject to linearized version of
Figure 3.4: Optimal adaptation gain compared to its upper and lower bounds (Algorithm 1).

the system. Ways to improve our confidence in this control (and hence the optimal adaptation gain) include the minimization of the cost function in MPC design subject to the actual system dynamics (if possible). Unfortunately, no optimality guarantees are provided in the case where the original system is nonlinear. The other way is to use a sufficient number of linear/affine models that capture the nonlinearities over the entire space of interest to get a more accurate locally optimal control. One such a system that can be constructed as a convex combination of multiple linear/affine models is the TSFS. More details about the construction and MPC design for TSFS are provided in [83]. The closed-loop response of the overall system is shown in
Figure 3.5. The first plot shows a comparison between the plant's output and

![Figure 3.5: Optimal adaptive direct fuzzy controller for the surge tank (Algorithm 2).](image)

the desired square wave reference trajectory. It is clear that the output tracks the reference input with a small offset error in the first cycle; this error decreases in the subsequent cycles due to the effect of adaptation. The second and third plots in the figure show the output error $e(k)$ and the optimal control $u^{opt}(k)$, respectively. A plot of the optimal adaptation gain $\eta^{opt}(k)$ (compared to the upper and lower bounds) is shown in Figure 3.6.

Hence we see that the closed-loop performance of the two algorithms is so similar that is hard to judge that one is better that the other. It is also clear from Figures 3.6
Figure 3.6: Optimal adaptation gain compared to its upper and lower bounds (Algorithm 1).

and 3.4 that the lower bound of the adaptation gain \( \eta_{\text{min}}(k) \) (in the two algorithms) is used as the optimal one when the output error is very small. This agrees with our intuition since no major changes are needed in the adaptive controller when the achieved closed-loop performance is acceptable. In this case, no major changes in the parameter vector (and hence a small adaptation gain) are needed. In the case where the output error is fairly large, considerable changes in the adaptive controller are needed. This translates to larger changes in the parameter vector (which, of course, require a large adaptation gain). As expected, the results provided by this example in the direct case verify our intuition on the choices of the adaptation gain.
Note that the two algorithms presented earlier focus on auto-tuning the adaptation gain by minimizing the control energy. For this reason, let us discuss how these algorithms impact the resulting control energy. To do that, we need to investigate how the mean squared error (MSE, that is defined as $\frac{1}{N} \sum_{k=1}^{N} e^2(k)$, where $N$ is the number of simulation time steps) and mean control energy (MCE, that is defined as $\frac{1}{N} \sum_{k=1}^{N} u^2(k)$) change for different values over the feasible range of adaptation gain (which is in this case $0 < \eta < \frac{2\pi}{\theta_1} = 40$). Figure 3.7 shows how both the MCE and MSE change for several fixed values of the adaptation gain over a simulation period of 50 seconds. The first plot in Figure 3.7 shows the changes in MCE for several

![Changes in MCE and MSE as η varies](image)

Figure 3.7: Changes in the MCE and MSE as $\eta$ varies in the direct adaptive case (Algorithm 1).
values of the adaptation gain. The dotted line in this figure shows the value of the MCE when the first auto-tuning algorithm is used. This value is found to be 50.007. It is clear from the figure that the MCE increases as the adaptation gain increases. This observation is expected since as the adaptation gain increases, the overall norm of the controller parameter vector increases, and hence the control energy increases. Similarly, the second plot in the figure shows the changes in MSE for several fixed values of the adaptation gain, and the dotted line shows the value of the MSE when the first auto-tuning algorithm is used. This value is found to be 0.0369. It is clear from the figure that the MSE decreases as the adaptation gain increases to some value (about 30), and then the MSE starts to slightly increase. The decrease in the first part of the range is due to the increase of the amount of control that improves the closed-loop performance. However, an excessive amount of control (that occurs here when the adaptation gain exceeds 30) can deteriorate the closed-loop performance, and hence increases the MSE. From the two plots, it is observed that the auto-tuning algorithm is able to achieve an MSE that is smaller than the MSE that can be achieved using any fixed adaptation gain, with a relatively small control energy. Hence, we can conclude that our simulation results support the objective of the presented algorithm in the sense that the adaptation gain is selected on-line to minimize the control energy in such a way that a good closed-loop performance is achieved.

As in the first algorithm, Figure 3.8 shows for the second algorithm how both the MCE and MSE change for several fixed values of the adaptation gain over a simulation period of 50 seconds. The values of the MCE and MSE when the second auto-tuning algorithm is used are found to be 50.2464 and 0.0366, respectively. A
similar discussion can be applied when the second algorithm is used. Hence, our conclusion from the first algorithm is also applicable here.

![Changes in MCE and MSE as \( \eta \) varies](image)

**Figure 3.8:** Changes in the MCE and MSE as \( \eta \) varies in the direct adaptive case (Algorithm 2).

**Indirect Case**

Here, two unknown functions \((\alpha(k) \text{ and } \beta(k))\) are to be approximated on-line in order for the control to be computed, assuming prior knowledge about the desired reference signal. The actual functions (that we assume to be unknown) to be approximated are

\[
\alpha(k) = h(k) + T \left[ \frac{-\sqrt{19.6}h(k)}{A(h(k))} \right]
\]
and
\[ \beta(k) = \frac{1}{\lambda(h(k))} \]

The approximators of both functions are TSFS that have one input, \( h(k) \). Three Gaussian membership functions are used for the input in each approximator, where the centers are distributed evenly between 0 and 4. It is assumed that \( W_a = W_b = 0.001 \). As in the direct case, we choose \( \alpha_1 = 0.1 \) and \( \alpha_2 = 0.95 \); this implies that \( \rho = 1.5 \). To ensure that \( \dot{\beta}(k) \) is bounded away from zero, we let \( \dot{\beta}(k) \geq \beta_0 \) where \( \beta_0 = 0.05 \). To guarantee boundedness of the adaptation gain, we assumed that \( |\zeta(k)|^2 \geq \tilde{\zeta} \), where \( \tilde{\zeta} = 1 \). Using \( \gamma = 1 \), this translates to having the upper bound on the maximum possible adaptation gain to be 4.

Algorithm 1

Here, the adaptation gain is found by minimizing the instantaneous control energy. The closed loop response of the overall system is shown in Figure 3.9, where the first plot shows a comparison between the plant's output and the desired square wave reference trajectory. The second and third plots in the figure show the output error \( e(k) \) and the optimal control \( u^{opt}(k) \), respectively. A plot of the optimal adaptation gain \( \eta^{opt}(k) \) (compared to the upper and lower bounds) is shown in Figure 3.10.

Algorithm 2

Here, the adaptation gain is the one that produces a locally optimal control obtained using MPC. The affine model used in the MPC design is the same one used in
Figure 3.9: Indirect adaptive fuzzy controller for the surge tank (Algorithm 1).

the direct case with the same weighting matrices. The closed-loop performance using Algorithm 2 is shown in Figure 3.11 where the plot is similar to the one in Algorithm 1. A plot of the optimal adaptation gain $\eta^{opt}(k)$ (compared to the upper and lower bounds) is shown in Figure 3.12. It is clear that using Algorithm 2, the tracking error decreases faster than in Algorithm 1. Also, in the second period of our simulation shown in Figures 3.12 and 3.10 (where the parameter vector are converging toward the ideal ones), the lower bound of the adaptation gain is used as an optimal one. This agrees with our intuition as in the direct case. It is important to note that the convergence of the parameter vector in the indirect case in simulation is slower than the convergence in the direct case. This simulation result agrees with our conclusion
that says that no matter what bounds we choose for our adaptation gain (as long as $0 < \alpha_1 < \alpha_2 < 1$), the parameter $\rho$ is always bounded in the interval $(0, 2)$; whereas in the direct case $\rho$ can be much larger (in this case we picked it to be 30).

As in the direct case, here we will discuss how these algorithms impact the resulting control energy. To do that, we need to investigate how the MSE and MCE change for different values over the feasible range of adaptation gain (which is in this case $0 < \eta < 2\gamma = 2$). Figure 3.13 shows how both the MCE and MSE change for several values of the adaptation gain over a simulation period of 50 seconds for the same reference signal used in the direct case. The first plot in Figure 3.13 shows the
Figure 3.11: Indirect adaptive fuzzy controller for the surge tank (Algorithm 2).

changes in MCE for several values of the adaptation gain. The dotted line in this figure shows the value of the MCE when the first auto-tuning algorithm is used. This value is found to be 46.0588. From the figure, as the adaptation gain increases, the MCE increases to around 46 and oscillates in that range. Similarly, the second plot in the figure shows the changes in MSE for several values of the adaptation gain, and the dotted line shows the value of the MSE when the first auto-tuning algorithm is used. This value is found to be 0.0846. It is clear from the figure that the MSE decreases as the adaptation gain increases to some value (about 1.2), and then the MSE starts to slightly increase. From the two plots, it is observed that the auto-tuning algorithm is able to achieve an acceptable MSE with a relatively small control energy. Hence,
we can conclude that our simulation results support the objective of the presented algorithm in the sense that the adaptation gain is selected online to minimize the control energy in such a way that a good closed-loop performance is achieved.

As in the first algorithm, Figure 3.14 shows how both the MCE and MSE change for the second algorithm at several values of the adaptation gain over a simulation period of 50 seconds. The values of the MCE and MSE when the second auto-tuning algorithm is used are found to be 43.1138 and 0.1636, respectively. Note that, with our set of current tuning parameters, a good MCE is achieved at the expense of a relatively large MSE. Hence, there is a trade-off between the desired closed-loop
Figure 3.13: Changes in the MCE and MSE as \( \eta \) varies in the indirect adaptive case (Algorithm 1).

performance relative to the affordable control energy. A similar discussion to the one presented in the first algorithm can be applied when the second algorithm is used. Hence, our conclusion from the first algorithm is also applicable here.

### 3.5 Concluding Remarks

The main contribution of this chapter is the two algorithms to “optimally” update the parameter vector (in the sense that the adaptation gain of the update law is updated on-line to minimize certain criterion) for both direct and indirect adaptive controllers. The optimal adaptation gain is expected to improve the approximation
accuracy of the adaptive controller: however, it is important to mention that our approaches do not necessarily provide the true optimal gains that could help achieve this. The first algorithm we presented obtains the optimal adaptation gain by minimizing the instantaneous control energy which is very desirable in many applications. The second approach we presented uses a control ($u^\eta$) (that is found by optimizing some criterion) to obtain the optimal adaptation gain. We are no more confident about the optimal adaptation gain than the control ($u^\eta$). Based on the results of the example presented earlier, it is difficult to favor one of the two algorithms over the other. However, comparisons can be made between direct and indirect cases. Unlike

Figure 3.14: Changes in the MCE and MSE as $\eta$ varies in the indirect adaptive case (Algorithm 2).
the direct case, it is shown that the selection of the bounds of the adaptation gain in the indirect case is independent of the plant dynamics. This represents a major advantage of the direct case since plant information is incorporated to find the optimal adaptation gain whereas no such information is used in the indirect case.
CHAPTER 4

AUTO-TUNING THE DIRECTION OF DESCENT FOR DISCRETE-TIME NONLINEAR ADAPTIVE CONTROL SYSTEMS

4.1 Introduction

In direct adaptive control, the adaptation mechanism attempts to adjust a parameterized nonlinear controller to approximate an ideal controller. In the indirect case, however, we approximate parts of the plant dynamics that are used by a feedback controller to cancel the system nonlinearities. In both cases, "approximators" such as linear mappings, polynomials, fuzzy systems, or neural networks can be used as either the parameterized nonlinear controller or identifier model. In this chapter, we present an algorithm to tune the direction of descent for a gradient-based approximator parameter update law used for a class of nonlinear discrete-time systems (considered in [43]) in both direct and indirect cases. In this algorithm, the direction of descent is obtained by minimizing the instantaneous control energy. We will show that updating the adaptation gain (presented in [84, 85]) can be viewed as a special case of updating the direction of descent. Finally, we will illustrate the performance of the algorithm via a simple surge tank example.
As discussed earlier, the gradient update routine is based on the idea that starting with an initial value for the parameter vector, the gradient algorithm changes (updates) this vector by adding to it another vector having a magnitude and a direction of descent. We can think of this as searching for the ideal parameter vector. To improve (or loosely speaking, attempt to "optimize") the performance of the searching mechanism in the gradient-based update law, we will attempt to modify the direction of descent so that a certain cost criterion of interest is optimized. Note that this approach is applicable to both direct and indirect adaptive control schemes.

This chapter is organized as follows. In Section 4.2, the assumptions used in the algorithm development and stability analysis are stated. Then, in Section 4.3, the algorithm for auto-tuning the direction of descent is presented. In Section 4.4, a geometric interpretation of the algorithm is provided, and a relationship between auto-tuning the adaptation gain and auto-tuning the direction of descent is mathematically derived and discussed. Then, in Section 4.5 the stability results are presented. The simulation results are presented in Section 4.6. Finally, in section 4.7 we provide some concluding remarks.

4.2 Assumptions

In addition to Assumptions 1, 2, and 3, we present the following assumptions.

**Assumption 4:** The new direction of descent vector $\zeta^*(k)$ can be expressed as

$$\zeta^*(k) = \zeta(k) + \zeta_e(k) \quad (4.1)$$
where $\zeta(k)$ is defined as an incremental vector that when it is added to the nominal directional vector $\xi(k)$, $\xi^*(k)$ is obtained. We also assume that

$$-\bar{\sigma} \text{abs} [\zeta(k)] \leq \zeta(k) \leq \bar{\sigma} \text{abs} [\zeta(k)]$$  \hspace{1cm} (4.2)$$

where $\zeta(k)$ and $\text{abs} [\zeta(k)] \in \mathbb{R}^p$ are defined as

$$\zeta(k) = \begin{bmatrix}
\zeta_1(k) \\
\zeta_2(k) \\
\vdots \\
\zeta_p(k)
\end{bmatrix}$$  \hspace{1cm} (4.3)$$

and

$$\text{abs} [\zeta(k)] = \begin{bmatrix}
\text{abs} [\zeta_1(k)] \\
\text{abs} [\zeta_2(k)] \\
\vdots \\
\text{abs} [\zeta_p(k)]
\end{bmatrix}.$$  \hspace{1cm} (4.4)$$

Assumption 5: We assume that $|\zeta(k)|$ has a known upper bound $\zeta_{\text{max}}$ (i.e., $|\zeta(k)| \leq \zeta_{\text{max}}, \forall k = 0, 1, 2, \cdots$).

Now, we will present our proposed algorithm for auto-tuning the direction of descent.

4.3 Algorithm Description

Using the above assumptions, our goal here is to auto-tune the direction of descent vector that is characterized by $\zeta(k)$ so that an improved directional vector $\xi^*(k)$ is obtained. Recall the gradient update law defined in Equation (2.35). Substituting Equation (4.1) into the update law defined in Equation (2.35), we get

$$A(k) = A(k - 1) + \frac{\kappa \eta \epsilon(k) [\zeta(k - d) + \zeta_e(k - d)]}{1 + \gamma |\zeta(k - d) + \zeta_e(k - d)|^2}. \hspace{1cm} (4.5)$$
To simplify notation, let \( \varphi(k) = \frac{c_{pe}(k)}{1 + \gamma(c(k-d) + \zeta(k-d))} \), then Equation (4.5) becomes

\[
A(k) = A(k-1) + \varphi(k) [\zeta(k-d) + \zeta_e(k-d)] .
\] (4.6)

Based on the definition of the parameter error \( \phi(k) = \dot{A}(k) - A^* \), \( \phi(k) \) can be expressed as

\[
\phi(k) = \dot{\phi}(k-1) + \varphi(k) [\zeta(k-d) + \zeta_e(k-d)] .
\] (4.7)

One choice of a cost criterion that we wish to optimize is the control energy defined as \( J_u = u^2(k) \). Thus, we need a new formulation of \( u(k) \) in which the new definition of the directional vector is incorporated. Based on Equation (4.1), \( u(k) \) can be defined as

\[
u(k) = A^T(k) [\zeta(k) + \zeta_e(k)] .
\] (4.8)

Substituting Equation (4.6) into Equation (4.8), \( u(k) \) can be written as

\[
u(k) = \Psi(k)\zeta_e(k) + \Phi(k).
\] (4.9)

where

\[

\Psi(k) = A^T(k-1) + \varphi(k)\zeta^T(k-d) + \varphi(k)\zeta_e^T(k-d) .
\]

and

\[

\Phi(k) = A^T(k-1)\zeta(k) + \varphi(k)\zeta^T(k-d)\zeta(k) + \varphi(k)\zeta_e^T(k-d)\zeta(k) .
\]

Based on this formulation, it can be shown that the instantaneous control energy function \( J_u = u^2(k) \) can be written as

\[
J_u = u^2(k) = \zeta_e^T(k)H_1(k)\zeta_e(k) + 2H_2(k)\zeta_e(k) + H_3(k) ,
\] (4.10)

where \( H_1(k) = \Psi^T(k)\Psi(k), \) \( H_2(k) = \Phi(k)\Psi(k), \) and \( H_3(k) = \Phi^T(k)\Phi(k) . \) Since \( H_3(k) \) is not defined in terms of \( \zeta_e(k) \), it can be omitted from the function we need
to optimize. Thus, to find $\zeta_e(k)$ we need to solve the following cost function

$$\min J_u(\zeta_e) = \min \{ \zeta_e^T(k)H_1(k)\zeta_e(k) + 2H_2(k)\zeta_e(k) \}$$  \hspace{1cm} (4.11)

It is known that the cost function (4.11) has a unique solution when it is solved subject to a linear constraint. In this case the cost function $J_u(\zeta_e)$ can be solved subject to the linear constraint (4.2) defined in Assumption 4. Hence, to solve for $\zeta_e(k)$, the following cost function needs to be minimized.

$$J_u = \min \{ \zeta_e^T(k)H_1(k)\zeta_e(k) + 2H_2(k)\zeta_e(k) \} \hspace{1cm} (4.12)$$

s.t. $-\bar{\sigma} \, \text{abs}[\zeta(k)] \leq \zeta_e(k) \leq \bar{\sigma} \, \text{abs}[\zeta(k)]$.

At this point, we have derived a direction of descent that minimizes not only the squared output error, but also the control energy. Substituting the direction of descent $\zeta^*(k)$ in (2.35), a new parameter vector (that uses $\zeta^*(k)$ as its new direction of descent) can be obtained. Finally, this new direction of descent, $\zeta^*(k)$, can be used to determine the new parameter vector (using $\zeta(k) = \zeta^*(k)$ in (2.35)) and new control (using (2.18) and (2.26) in the direct and indirect cases, respectively).

### 4.4 Geometrical Interpretation and Relation to Auto-Tuning the Adaptation Gain

In Assumption 4, we assumed that the new direction of descent can be expressed by (4.1), where $\zeta_e(k)$ is defined as an incremental vector that when it is added the nominal directional vector $\zeta(k)$, an improved vector $\zeta^*(k)$ is obtained. To describe this assumption geometrically, consider (for simplicity) the two dimensional case (i.e., $\zeta(k) \in \mathbb{R}^2$). This situation can be shown in Figure 4.1. We know (from Assumption 4) that

$$\zeta^*(k) = \zeta(k) + \zeta_e(k).$$  \hspace{1cm} (4.13)
It can be verified that $\zeta^*(k)$ can be expressed as

$$\zeta^*(k) = \zeta(k) + \sigma(k)\zeta(k)$$  \hspace{1cm} (4.14)$$

where

$$\sigma(k) = \begin{bmatrix} \sigma_1(k) & 0 & 0 \\ 0 & \sigma_2(k) & \cdot \\ \cdot & 0 & \cdot \\ \cdot & \cdot & \cdot \\ 0 & 0 & \sigma_p(k) \end{bmatrix}$$  \hspace{1cm} (4.15)$$

and $-\bar{\sigma} \leq \sigma_i(k) \leq \bar{\sigma}$, $\forall i = 1, 2, \cdots, p$. Equation (4.14) can be written as

$$\zeta^*(k) = \Lambda(k)\zeta(k),$$  \hspace{1cm} (4.16)$$

where $\Lambda(k) = I + \sigma(k)$ and $I$ is the identity matrix. Note that the diagonal matrices $\Lambda(k)$, $\sigma(k)$, and $I \in \mathbb{R}^{p \times p}$. It is clear that $\Lambda(k)$, which can be expressed as

$$\Lambda(k) = \begin{bmatrix} 1 + \sigma_1(k) & 0 & 0 \\ 0 & 1 + \sigma_2(k) & \cdot \\ \cdot & 0 & \cdot \\ \cdot & \cdot & \cdot \\ 0 & 0 & 1 + \sigma_p(k) \end{bmatrix}$$
is a diagonal gain matrix that when it is multiplied by the nominal direction of
descent, a new vector $\zeta^*(k)$ is obtained. From the definition of $\sigma(k)$, we know that

$$-\bar{\sigma} \leq \sigma_i(k) \leq \bar{\sigma}$$

$$1 - \bar{\sigma} \leq \sigma_i(k) + 1 \leq \bar{\sigma} + 1$$

$$1 - \bar{\sigma} \leq \Lambda_i(k) \leq 1 + \bar{\sigma}. \; \forall i = 1, 2, \ldots, p$$

(4.18)

Note that for the special case when

$$\Lambda_1(k) = \Lambda_2(k) = \cdots = \Lambda_p(k).$$

(4.19)

$\zeta^*(k)$ will be a scaled version of $\zeta(k)$, that is we only change the magnitude of the
direction of descent vector. It is clear that this special case is equivalent to auto-tuning
the adaptation gain (presented in [84, 85]). However, when the condition (4.19) is not
satisfied, some type of change will occur to the direction of the direction of descent
vector.

As a result of this algorithm, the overall control energy is expected to decrease
as $\bar{\sigma}$ increases since $\zeta^*(k)$ will be affected by $\zeta(\epsilon)$, which is found by minimizing
the instantaneous control energy. It is interesting to note that the line of argument
used here to select the magnitude of $\bar{\sigma}$ is similar to one used in selecting the gains
of the quadratic cost function $(x^T(k)Qx(k) + u^T(k)Ru(k))$ used in linear quadratic
regulation problems, where the gains $Q$ and $R$ are selected based on the desired
closed-loop performance relative to the affordable control energy.
4.5 Stability Analysis

For both direct and indirect cases, we present the following result.

**Theorem 7:** Suppose \(|r(k)| \leq d_1\) for all \(k \geq 0\). Given any constant \(q > 0\) and any small constant \(\epsilon > 0\), there exist positive constants \(q_1 = q_1(q, d_1)\), \(q_2 = q_2(q, d_1)\), \(\epsilon^* = \epsilon^*(q, \epsilon, d_1)\), and \(\delta^* = \delta^*(q, \epsilon, d_1)\) such that if Assumptions 1 and 3 are satisfied on \(S_x \supset B_{q_1}\) with \(\epsilon < \epsilon^*\), Assumption 2 is satisfied on \(B_{q_2}\). Assumptions 4 and 5 are satisfied for all \(k \geq 0\). \(|x(0)| \leq q\), \(|\phi(0)| \leq \delta < \delta^*\), and the parameters \(\eta\) and \(\gamma\) are selected to satisfy

\[
0 < (\eta \theta_1 - 2\gamma) < \frac{2}{(\sigma + 2)^2 \zeta_{\text{max}}^2},
\]

then using either the direct (2.18) or indirect (2.26) adaptive control law with the direction of descent selected by solving (4.12), we will ensure that

1. \(|\phi(k)|\) will be monotonically nonincreasing, and \(|\phi(k) - \phi(k - 1)|\) will converge to zero.

2. The tracking error between the plant output and the reference command will converge to a ball of radius \(\epsilon\) centered at the origin.

**Proof:** The proof of this theorem is similar to the proof of Theorem 1 in the direct case (or Theorem 2 in the indirect case), except for the part in Step 2 where we need to show that the Lyapunov-like function \(V(k) = \phi^T(k)\phi(k)\) is monotonically nonincreasing. We will consider the case where \(e(k)\) is within the dead zone separate from the case where \(e(k)\) is outside the dead zone. First, consider the case where \(e(k)\) is inside the dead zone. In this case, \(e_\epsilon(k) = 0\) so \(\phi(k) = \phi(k-1)\) and \(V(k) - V(k-1) = \)
0. With the error outside the dead zone, we have

\[
\begin{align*}
V'(k) - V'(k-1) &= \phi^T(k)\phi(k) - \phi^T(k-1)\phi(k-1) \\
&= \left(\phi(k-1) + \frac{\kappa \eta [\zeta(k-d) + \zeta_e(k-d)]}{1 + \gamma \|\zeta(k-d) + \zeta_e(k-d)\|^2} e_e(k)\right)^T \\
&\quad \left(\phi(k-1) + \frac{\kappa \eta [\zeta(k-d) + \zeta_e(k-d)]}{1 + \gamma \|\zeta(k-d) + \zeta_e(k-d)\|^2} e_e(k)\right) \\
&\quad - \phi^T(k-1)\phi(k-1) \\
&= e_e(k) - K T T(k) \theta(k-d)\phi(k-d) \left[\zeta(k-d) + \zeta_e(k-d)\right]
\end{align*}
\] (4.21)

Based on Equation (2.38), it can be shown that

\[
e_e(k) = -\kappa \pi(k)\theta(k-d)\phi(k-d)^T [\zeta(k-d) + \zeta_e(k-d)]
\] (4.22)

Using Equation (4.22) and the fact that \(\kappa^2 = 1\), Equation (4.21) can be expressed as

\[
\begin{align*}
V'(k) - V'(k-1) &= \eta \left[ -\frac{2}{\pi(k)\theta(k-d)} \\
&\quad + \frac{\eta [\zeta(k-d) + \zeta_e(k-d)]^2}{1 + \gamma \|\zeta(k-d) + \zeta_e(k-d)\|^2} \right] \\
&\quad \frac{e_e^2(k)}{1 + \gamma \|\zeta(k-d) + \zeta_e(k-d)\|^2}
\end{align*}
\] (4.23)

Since \(0 < \pi(k) < 1\) and \(0 < \theta_0 \leq \theta(k-d) \leq \theta_1\), we get

\[
V'(k) - V'(k-1) \leq \eta \left[ -\frac{2}{\theta_1} \\
&\quad + \frac{\eta [\zeta(k-d) + \zeta_e(k-d)]^2}{1 + \gamma \|\zeta(k-d) + \zeta_e(k-d)\|^2} \right] \\
&\quad \frac{e_e^2(k)}{1 + \gamma \|\zeta(k-d) + \zeta_e(k-d)\|^2}
\]

For \((V'(k) - V'(k-1)) \leq 0\), we need

\[
-\frac{2}{\theta_1} + \frac{\eta [\zeta(k-d) + \zeta_e(k-d)]^2}{1 + \gamma \|\zeta(k-d) + \zeta_e(k-d)\|^2} < 0
\] (4.24)

Define \(\Gamma(k)\) such that \(\Gamma(k) = |\zeta(k) + \zeta_e(k)|^2\), where \(\zeta(k)\) and \(\zeta_e(k)\) \(\in \mathcal{R}^p\), and \(\Gamma(k) \in \mathcal{R}^+\) (\(\mathcal{R}^+\) is defined as the set of positive real numbers). It can be shown that

\[
\Gamma(k) = |\zeta(k)|^2 + |\zeta_e(k)|^2 + 2\zeta(k)^T \zeta_e(k)
\] (4.25)
Since $\zeta(k) = \Lambda(k)\zeta(k)$, Equation (4.25) becomes

$$
\Gamma(k) = |\zeta(k)|^2 + |\Lambda(k)\zeta(k)|^2 + 2\zeta(k)^T\Lambda(k)\zeta(k)
= \sum_{i=1}^{p} \zeta_i^2(k) + \sum_{i=1}^{p} \Lambda_i^2(k)\zeta_i^2(k) + 2\sum_{i=1}^{p} \Lambda_i(k)\zeta_i^2(k)
= \sum_{i=1}^{p} \{\zeta_i^2(k) \left[1 + \Lambda_i^2(k) + 2\Lambda_i(k)\right]\}
= \sum_{i=1}^{p} \{\zeta_i^2(k) \left[1 + \Lambda_i(k)\right]^2\} \quad (4.26)
$$

Since $\Lambda_i(k) = 1 + \sigma_i(k)$. Equation (4.26) becomes

$$
\Gamma(k) = \sum_{i=1}^{p} \{\zeta_i^2(k) \left[\sigma_i(k) + 2\right]^2\} \quad (4.27)
$$

Based on Equation (4.24), to guarantee that $V(k) - V(k - d) \leq 0$, we need

$$
\left[\frac{-2}{\theta_i} + \frac{\eta\Gamma(k - d)}{1 + \gamma\Gamma(k - d)}\right] < 0 \quad (4.28)
$$
or equivalently

$$
\frac{\eta\theta_i\Gamma(k - d) - 2(1 + \gamma\Gamma(k - d))}{\theta_i(1 + \gamma\Gamma(k - d))} < 0 \quad (4.29)
$$

This can be achieved when

$$
\Gamma(k - d) \left[\eta\theta_i - 2\gamma\right] < 2 \quad (4.30)
$$

Hence, we need to ensure that

$$
\Gamma(k - d) < \frac{2}{\eta\theta_i - 2\gamma} \quad (4.31)
$$

Combining Equations (4.27) and (4.31), we need to guarantee that

$$
\max_k \left\{\sum_{i=1}^{p} \{\zeta_i^2(k - d) \left[\sigma_i(k - d) + 2\right]^2\}\right\} < \frac{2}{\eta\theta_i - 2\gamma} \quad (4.32)
$$

Since $\sigma_i(k - d) \leq \bar{\sigma}$, we need to ensure that

$$
\max_k \left\{[\bar{\sigma} + 2]^2 \sum_{i=1}^{p} \{\zeta_i^2(k - d)\}\right\} < \frac{2}{\eta\theta_i - 2\gamma} \quad (4.33)
$$
It is clear from (4.34) that we need also to guarantee that $0 < \eta \theta_1 - 2\gamma$ (since both sides of the equation have to be positive). Therefore, choosing the parameters $\eta$ and $\gamma$ to satisfy

$$0 < (\eta \theta_1 - 2\gamma) < \frac{2}{(\sigma + 2)^2 \zeta_{\text{max}}}.$$  

(4.35)

will ensure that $V(k) - V(k - d) \leq 0$.

Note that the condition (4.20) may be infeasible for certain class of systems. However, if there is a class of systems for which this condition is satisfied, then stability can be guaranteed.

### 4.6 Simulation Results

Consider the surge tank model (taken from [91]) that can be represented by the following differential equation:

$$\frac{dh(t)}{dt} = -c\sqrt{2gh(t)} + \frac{1}{Ar(h(t))}u(t)$$  

(4.36)

where $u(t)$ is the input flow (control input), which can be positive or negative. Also, $h(t)$ is the liquid level (output of the system); $Ar(h(t))$ is the cross-sectional area of the tank; $g = 9.8m/sec^2$ is the gravitational acceleration; and $c = 1$ is the known cross-sectional area of the output pipe. Let $Ar(h(t)) = \sqrt{ah(t) + b}$, where $a = 1$ and $b = 3$. Using Euler approximation to discretize the system, we have

$$h(k + 1) = h(k) + T \left[ -\sqrt{19.6h(k)} \cdot \frac{1}{Ar(h(k))} + \frac{u(k)}{Ar(h(k))} \right]$$  

(4.37)
where $T = 0.1$. Note that the system (4.37) belongs to the same class of systems (2.1), where $d = 1$, 

$$f_0(x(k)) = h(k) - \frac{T \sqrt{19.6h(k)}}{A_r(h(k))},$$  \hspace{1cm} (4.38) 

and 

$$g_0(x(k)) = \frac{T}{A_r(h(k))}.$$ \hspace{1cm} (4.39) 

We will simulate the system for $h(k) > 0$ so that the simulation is realistic. This system will be used here to demonstrate how to use the previously discussed algorithm for auto-tuning the direction of descent for the direct adaptive case: similar analysis can be performed also for the indirect adaptive control case.

The ideal controller is approximated here using a TSFS that has two inputs, the reference input (which is a square wave whose upper and lower values are 1.5 and 3) and error, $e(k) = r(k) - h(k)$. Five Gaussian membership functions are used for each input universe of discourse. For the first input (reference input), the centers of the membership functions are distributed evenly between 0 and 3, and the centers for the second input (error) are distributed evenly between $-5$ and 5. To satisfy Assumption 4, the function (4.12) is minimized subject to $-\bar{\sigma} \leq \sigma_{\text{abs}}[\zeta(k)] \leq \bar{\sigma}$, where we choose $\bar{\sigma}$ is to be 0.7. It is assumed that $0 < \theta_0 \leq \theta(x(k)) \leq \theta_1$, where $\theta_0$ and $\theta_1$ are 0.033 and 0.05, respectively. Also, based on Assumption 5 (i.e., $\|\zeta(k)\| \leq \zeta_{\text{max}}$), we assume $\zeta_{\text{max}}$ to be 2. To guarantee that $0 < (\eta \theta_1 - 2\gamma)$, we choose the parameters $\eta$ and $\gamma$ to be 1 and 0.01, respectively. To satisfy (4.20), we choose $\bar{\sigma}$ to be 0.7. Note that the condition (4.20) can be satisfied for other values of $\bar{\sigma}$. To study the effect of the direction of descent adaptation, we compared the case where the direction of descent is adapted to the case where it is not. The performance of the closed-loop system for the first case is shown in Figure 4.2, where the first plot
shows a comparison between the plant's output and the desired square wave reference trajectory. The second and third plots in the figure show the output error $e(k)$ and the new control $u(k)$, respectively. The performance of the closed-loop system for the second case (when no direction of descent adaptation is used) is shown in Figure 4.3, where the first plot shows a comparison between the plant's output and the desired square wave reference trajectory. Also, the second and third plots in the figure show the output error $e(k)$ and the control $u(k)$, respectively. It is clear from the first plot in Figure 4.2 that the closed-loop performance seems unacceptable at the beginning; however, the performance starts to improve over time. Note that the

Figure 4.2: Performance of direct adaptive controller when direction is adapted.
Figure 4.3: Performance of direct adaptive controller when direction is not adapted.

The magnitude of overshoot in Figure 4.2 (when the direction of descent is adapted) decreases faster than the corresponding one in Figure 4.3 (when the direction of descent is not adapted). However, the output error in Figure 4.3 (when the direction of descent is not adapted) decreases faster than the corresponding one in Figure 4.2 (when the direction of descent is adapted). This observation agrees with our conclusions since the parameter vector in the first case is adapted to minimize only the squared output error; in the second case, however, there is a trade-off between minimizing the squared instantaneous output error and the instantaneous control energy. Variations in the MCE and the MSE are shown in Figure 4.4. The values shown in Figure 4.4 are average values over a simulation period of 100 seconds. It is clear that the MCE
decreases as \( \sigma \) increases. This observation makes sense since as \( \sigma \) increases, the incremental vector \( \zeta_e(k) \) (which is found by minimizing the instantaneous control energy) may become larger, and hence \( \zeta(k) \) may become greatly affected by \( \zeta_e(k) \). Therefore, as \( \sigma \) increases the resulting \( \zeta(k) \) will be obtained such that more consideration is given to minimizing the control energy. Also, it is clear from the figure that the MSE increases as \( \sigma \) increases. Hence, adapting the direction of descent can be used to trade-off between the desired closed-loop performance relative to the affordable control energy.
4.7 Concluding Remarks

Considering both direct and indirect adaptive control schemes, the main contribution of this chapter is to auto-tune the direction of descent for a gradient-based approximator parameter update law used for a class of nonlinear discrete-time systems. The adaptation mechanism of the gradient update law is usually based on minimizing the squared output error. Here, however, we update the direction of descent to minimize the control energy. We have found via an example that auto-tuning the direction of descent helps to decrease the magnitude of overshoot and control energy, but standard trade-offs between tracking quality and use of control energy apply. We were also able to conclude that auto-tuning the adaptation gain can be viewed as a special case of auto-tuning the direction of descent.
5.1 Introduction

The authors in [32] presented indirect and direct adaptive control algorithms using linearly parameterized fuzzy systems or neural networks to control SISO affine systems with guaranteed convergence of the tracking error to zero in both cases. For the indirect case, the control law consists of three terms: a bounding control term to ensure boundedness of the output and states, a sliding mode term to compensate for approximation error, and a certainty equivalence term for which it is assumed that the current estimate of the plant parameters are the actual ones. In the direct case, however, an adaptive control term is used instead of the certainty equivalence term. Unique features of the algorithms presented in this dissertation include the utilization of prior knowledge (about the plant dynamics in the indirect case and the controller dynamics in the direct case) to specify the control and improve the performance, the incorporation of the inverse model dynamics in the direct case to enhance the overall performance, and the applicability of these algorithms to systems containing zero dynamics and/or state dependent input gain.
Building on this work, here we will auto-tune the adaptation gain for both direct and indirect adaptive control of a class of continuous time nonlinear systems. The adaptation gain is adapted to minimize the the instantaneous control energy.

This chapter is organized as follows. In Section 5.2, we discuss the plant description, plant assumptions, and both the direct and indirect adaptive control schemes. Then, in Section 5.3 we present the gradient-based hybrid parameter update law along with its stability results. In Section 5.4, we present an algorithm to auto-tune the adaptation gain, and we present its stability results. Then, in Section 5.5, we present the simulation results of this algorithm via a wing rock regulation example. Finally, in Section 5.6 we provide few concluding remarks.

5.2 Direct and Indirect Adaptive Control

In this section, we start by describing the system we consider for control, along with its assumptions. Then, both direct and indirect adaptive control schemes are briefly discussed, and the error equation is derived for both cases.

5.2.1 Plant Description

Here, we will consider the single-input single-output continuous-time system described by

\[
\begin{align*}
\dot{X} &= f(X) + g(X)u_p \\
y_p &= h(X)
\end{align*}
\]

(5.1)

where \( X \in \mathbb{R}^n \) is the state vector, \( u_p \in \mathbb{R} \) is the input, \( y_p \in \mathbb{R} \) is the output of the plant and functions \( f(X), g(X) \in \mathbb{R}^n \), and \( h(X) \in \mathbb{R} \) are smooth. If the system has "strong relative degree" \( r \), then it can be shown (as in [32]) that
\begin{align*}
\dot{\xi}_1 &= \xi_2 = L_1 h(X) \\
& \vdots \\
\dot{\xi}_{r-1} &= \xi_r = L_1^{r-1} h(X) \\
\dot{\xi}_r &= L_1^r h(X) + L_2 L_1^{r-1} h(X) u_r \tag{5.2}
\end{align*}

with \( \xi_1 = y_p \), which may be rewritten as

\[ y_p^{(r)} = (\alpha_k(t) + \alpha(X)) + (\beta_k(t) + \beta(X)) u_p \tag{5.3} \]

where \( L_g^r h(X) \) is the \( r \)th Lie derivative of \( h(X) \) with respect to \( g \) (\( L_g h(X) = \frac{\partial h}{\partial X} g(X) \)), and e.g. \( L_g^2 h(X) = L_g(L_g h(X)) \), and it is assumed that for some \( \beta_0 > 0 \), we have \( |\beta_k(t) + \beta(X)| \geq \beta_0 \) so that it is bounded away from zero (for convenience we assume that \( \beta_k(t) + \beta(X) > 0 \) however, the following analysis may easily be modified for systems which are defined with \( \beta_k(t) + \beta(X) < 0 \)). We will assume that \( \alpha_k(t) \) and \( \beta_k(t) \) are known components of the dynamics of the plant (that may depend on the state) or known exogenous time dependent signals and that \( \alpha(X) \) and \( \beta(X) \) represent nonlinear dynamics of the plant that are unknown. It is 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 \( t \geq 0 \).

**Definition 2:** The dynamics for a relative degree \( r \) plant described by 5.1 (as shown in [32]) may be written in normal form as

\begin{align*}
\dot{\xi}_1 &= \xi_2 \tag{5.4} \\
& \vdots \\
\dot{\xi}_r &= \xi_{r-1} \tag{5.5}
\end{align*}
\[ \dot{x}_{r-1} = \xi_r \]  
\[ \dot{\xi}_r = \alpha(\xi, \pi) + \beta(\xi, \pi)up \]  
\[ \dot{\pi} = \Psi(\xi, \pi) \]

with \( \pi \in \mathbb{R}^{n-r} \), and \( y_p = \xi_1 \). The "zero-dynamics" of the system are given as

\[ \dot{\pi} = \Psi(0, \pi). \]

Here, we will consider plants that either have no zero dynamics (i.e., \( n = r \)), or plants with zero dynamics (i.e., \( 1 \leq r < n \)) that are exponentially attractive. These types of plants are defined in the following plant assumptions [32].

### 5.2.2 Plant Assumptions

**Assumption 6:** The plant is of relative degree \( r = n \) (i.e. no zero dynamics), such that

\[
\frac{d}{dt} x_i = x_{i+1}, \quad i = 1, \ldots, n-1 \\
\frac{d}{dt} x_n = \alpha(X) + \alpha_k(t) + (\beta(X) + \beta_k(t)) u_p
\]

where \( y_p = x_1 \), \( \alpha_k(t) \) and \( \beta_k(t) \) known functions. Here it is assumed that there exists \( \beta_0 > 0 \) such that \( \beta(X) + \beta_k(t) \geq \beta_0 \), and that \( x_1, \ldots, x_n \) are measurable.

**Assumption 7:** The plant is of relative degree \( r \), \( 1 \leq r < n \), with the zero dynamics exponentially attractive and there exists \( \beta_0 > 0 \) such that \( \beta(X) + \beta_k(t) \geq \beta_0 \). The outputs \( y_p, \ldots, y_p^{(r-1)} \) are measurable.

It is clear that plants satisfying Assumption 6 have bounded states if the reference input and its derivatives are bounded, and the output error and its derivatives are
also bounded. It can also be shown (as in [32]) that plants satisfying Assumption 7 have bounded states if the output is bounded.

Next, a brief description of direct and indirect adaptive control schemes will be presented.

5.2.3 Direct Approach

A direct adaptive controller, that seeks to drive the output of a relative degree \( r \) plant \( y_p \) to track a known desired output trajectory \( y_m \), uses an approximator that attempts to approximate the ideal controller dynamics (\( u^* \), that we assume to exist) by adjusting the controller parameters. Hence, our objective is design a controller which makes the output of the plant \( y_p \) track the output trajectory \( y_m \).

In additions to the plant Assumptions 6 and 7, we use the following plant assumption [32].

**Assumption 8:** Given \( y_p^{(r)} = (\alpha(X) + \alpha_k(t)) + (\beta(X) + \beta_k(t))y_p \), we require that \( \beta_k(t) = 0, t \geq 0 \), and that there exists positive constants \( J_0 \) and \( J_1 \) such that \( 0 < J_0 \leq \beta(X) \leq J_1 < \infty \) and some function \( B(X) \geq 0 \) such that \( |\beta'(X)| = \left| \frac{\partial \beta}{\partial X} \dot{X} \right| \leq B(X) \) for all \( X \in S_X \). Here, \( \alpha_k(t) \) is a known time dependent signal. Here, we also require the following output trajectory assumption [32].

**Assumption 9:** The desired output trajectory and its derivatives \( y_m, \ldots, y_m^{(r)} \) are measurable and bounded. Using feedback linearization [92], we know that there exists some ideal controller

\[
\begin{align*}
  u^* &= \frac{1}{\beta(X)}(-\alpha(X) + \nu(t)) \\
\end{align*}
\] (5.10)
where \( \nu(t) := y_m^{(r)} + \delta \epsilon_s + \bar{e}_s - \alpha_k(t) \), with \( \bar{e}_s := \epsilon_s - o_0^{(r)} \), and \( \delta > 0 \). For now we assume that \( J_k(t) + \hat{\beta}(X) \) is bounded away from zero so that (5.10) is well-defined, however, we shall later show how to ensure that this is the case. The tracking error is defined as \( \epsilon_s := k^T \epsilon \) where \( \epsilon := \left[ e_o, \hat{e}_o, \ldots, e_o^{(r-1)} \right]^T \), \( k := [k_0, \ldots, k_{r-2}, 1]^T \), and \( e_o := y_m - y_p \), thus \( \bar{e}_s = \left[ \hat{e}_o, \ldots, e_o^{(r-1)} \right][k_0, \ldots, k_{r-2}]^T \). We pick the elements of \( k \) such that \( \hat{L}(s) := s^{r-1} + k_{r-2}s^{r-2} + \cdots + k_1 s + k_0 \) has its roots in the open left half plane. The goal of the adaptive controller is to "learn" how to control the plant to drive \( \epsilon_s \) (which is a measure of the tracking error) to some neighborhood of zero. We may express \( u^* \) as
\[
    u^* = A_u^T \zeta_u(X, \nu) + u_k + d_u(X) \tag{5.11}
\]
The ideal parameter vector, \( A_u^* \), is defined as
\[
    A_u^* := \arg \min_{A_u \in \Omega_u} \left[ \sup_{X \in S_x, \nu \in S_m} |A_u^T \zeta_u(X, \nu) - (u^* - u_k)| \right] \tag{5.12}
\]
where \( A_u \) is assumed to be defined within the compact parameter set \( \Omega_u \), and \( S_x \) and \( S_m \subseteq \mathbb{R}^n \) are defined as the spaces through which the state trajectory and the free parameter \( \nu(t) \) may travel under closed-loop control. Also, \( \zeta_u \) is defined as the partial of the approximator with respect to the parameter vector. \( u_k \) is a known part of the controller, and \( d_u(X) \) is the approximation error which arises when \( u^* \) is represented by an approximator (e.g., fuzzy system, neural network, or other universal approximator) of finite size. It is assumed that \( |d_u(X)| \leq D_u(X) \), where \( D_u(X) \) is a known upper bound on the error. Since universal approximators are used for approximation, \( |d_u(x)| \) may be made arbitrarily small by a proper choice of the approximator structure. To do this, we will require \( X \) and \( \nu \) to be available. The
ideal control (5.10) can be approximated by

\[ u_d = A_u^T \zeta_u + u_k, \]  

(5.13)

where \( A_u \) is updated on line. Using the control (5.13), the \( r \)th derivative of the output error becomes

\[ e_o^{(r)} = y_m^{(r)} - y_p^{(r)} = y_m^{(r)} - \alpha(X) - \alpha_k(t) - \beta(X)u_d \]  

(5.14)

Using the definition of \( u^* \) (5.10) we may rearrange (5.14) so that

\[ e_o^{(r)} = y_m^{(r)} - \alpha(X) - \alpha_k(t) - \beta(X)u^* - \beta(X)(u_d - u^*) \]  

(5.15)

\[ = -\delta e_s - \delta e_s - \beta(X)(u_d - u^*). \]  

(5.16)

We may alternatively express (5.16) as

\[ \dot{e}_s + \delta e_s = -\beta(X)(u_d - u^*). \]  

(5.17)

Assume for now that parameter vector \( A_u(k) \), is updated on line using a hybrid adaptive law (in later section, we will discuss this adaptive law in detail). Define the approximator parameter error as \( \phi(k) = A_u(k) - A_u^* \). Using the definitions of the ideal control (5.11) and the actual one (5.13), it can be shown that

\[ u_d - u^* = \phi(k)^T \zeta_u - d_u(X) \]  

(5.18)

Substituting (5.18) into (5.17), we can define \( \dot{e} := u^* - u_d \) in the direct case such that

\[ \dot{e} = \dot{\epsilon}_s + \delta \epsilon_s \frac{1}{\beta(X)} = -\phi(k)^T \zeta_u + d_u(X) \]  

(5.19)

Note that \( \dot{e} \) (which is a function of the plant dynamics, \( \beta(X) \)) is a measure of the tracking performance, and will be used in the parameter hybrid update law (as we will show in later sections).
5.2.4 Indirect Approach

Unlike the direct approach, in the indirect approach we approximate the plant dynamics \( \alpha(x) \) and \( \beta(x) \), then the feedback controller uses these estimates of the plant dynamics to tune the parameters of the controller so that the plant output \( y_p \) tracks the output trajectory \( y_m \). The plant dynamics \( \alpha(x) \) and \( \beta(x) \) can be expressed as

\[
\alpha(x) = A_\alpha^\top \zeta_\alpha(x) + d_\alpha(x) \quad (5.20)
\]

\[
\beta(x) = A_\beta^\top \zeta_\beta(x) + d_\beta(x) \quad (5.21)
\]

where

\[
A_\alpha^* := \arg\min_{A_\alpha \in \Omega_\alpha} \left[ \sup_{X \in S_\alpha} |A_\alpha^\top \zeta_\alpha(x) - \alpha(x)| \right] \quad (5.22)
\]

\[
A_\beta^* := \arg\min_{A_\beta \in \Omega_\beta} \left[ \sup_{X \in S_\beta} |A_\beta^\top \zeta_\beta(x) - \beta(x)| \right] \quad (5.23)
\]

The parameter vectors, \( A_\alpha \) and \( A_\beta \), are assumed to be defined within the compact parameter sets, \( \Omega_\alpha \) and \( \Omega_\beta \), respectively. In addition, we define the subspace \( S_\alpha \subseteq \mathbb{R}^n \) as the space through which the state trajectory may travel under closed-loop control. Also, \( d_\alpha(x) \) and \( d_\beta(x) \) are approximation errors which arise when \( \alpha(x) \) and \( \beta(x) \) are represented by approximators of finite size. We assume that \( D_\alpha(x) \geq |d_\alpha(x)| \), and \( D_\beta(x) \geq |d_\beta(x)| \), where \( D_\alpha(x) \) and \( D_\beta(x) \) are known bounds on the approximation errors. Since universal approximators (e.g., fuzzy systems, neural networks, and others [93]), both \( |d_\alpha(x)| \) and \( |d_\beta(x)| \) may be made arbitrarily small by a proper choice of the approximator if \( \alpha(x) \) and \( \beta(x) \) are smooth. It is important to keep in mind that \( D_\alpha(x) \) and \( D_\beta(x) \) represent the magnitude of error between the actual nonlinear functions describing the system dynamics and the the approximators when the “best” parameters are used.
We assume that the actual plant dynamics, \( \dot{\alpha}(X) \) and \( \dot{\beta}(X) \), can be expressed as

\[
\dot{\alpha}(X) = A_\alpha^T \zeta_\alpha \tag{5.24}
\]
\[
\dot{\beta}(X) = A_\beta^T \zeta_\beta \tag{5.25}
\]

where the vectors \( A_\alpha(k) \) and \( A_\beta(k) \) are updated on line (as we will show later) using a hybrid adaptive law. The parameter error vectors

\[
\phi_\alpha(k) = A_\alpha(k) - A_\alpha^* \tag{5.26}
\]
\[
\phi_\beta(k) = A_\beta(k) - A_\beta^* \tag{5.27}
\]

are used to define the difference between the current estimate of the parameters (at time \( k \)) and the best values of the parameters defined by (5.22) and (5.23). The certainty equivalence control term [94] is defined as

\[
u(t) := \gamma^T (\alpha_k(t) + \dot{\alpha}(X)) + \nu(t)
\]

where \( \nu(t) := y^{(r)}_m + \delta e_s + \bar{e}_s \), with \( e_s \) and \( \bar{e}_s \) defined as in the direct case. For now we assume that \( J_k(t) + J(X) \) is bounded away from zero so that (5.28) is well-defined, however, we shall later show how to ensure that this is the case. Using the control (5.28), the \( r^{th} \) derivative of the output error becomes \( e^{(r)}_o = y^{(r)}_m - y^{(r)}_p \) so

\[
e^{(r)}_o = y^{(r)}_m - (\alpha_k(t) + \alpha(X)) - \frac{J_k(t) + J(X)}{J_k(t) + J(X)} [-(\alpha_k(t) + \dot{\alpha}(X)) + \nu(t)]. \tag{5.29}
\]

We may rearrange terms so that

\[
e^{(r)}_o = \left[1 - \frac{J_k(t) + J(X)}{J_k(t) + J(X)}\right] \left(-\alpha_k(t) + \dot{\alpha}(X) + \nu(t)\right) - \alpha(X) + \dot{\alpha}(X) \tag{5.30}
\]

\[
= (\dot{\alpha}(X) - \alpha(X)) + (\dot{\beta}(X) - \beta(X)) u_i - \delta e_s - \bar{e}_s. \tag{5.31}
\]

112
We may express (5.31) as

\[ \dot{e}_s + \delta e_s = (\hat{\alpha}(X) - \alpha(X)) + \left(\hat{\beta}(X) - \beta(X)\right) u_i. \]  

(5.32)

Analogous to the direct case, it can be shown that

\[ \hat{\alpha}(X) - \alpha(X) = \phi^\top \zeta - d_\alpha(X) \]  

(5.33)

\[ \hat{\beta}(X) - \beta(X) = \phi d \zeta - d_\beta(X) \]  

(5.34)

Substituting (5.33) and (5.34) in (5.32), we get

\[ \dot{e}_s + \delta e_s = \left[\phi^\top \zeta - d_\alpha(X)\right] + \left[\phi d \zeta - d_\beta(X)\right] \]

\[ = \phi^\top \zeta - [d_\alpha(X) + d_\beta(X)] \]  

(5.35)

where \( \phi = [\phi_\alpha^\top, \phi_\beta^\top]^\top \), and \( \zeta = [\zeta_\alpha^\top, \zeta_\beta^\top u_i]^\top \). Let us define \( \dot{e} \) as

\[ \dot{e} := -\dot{e}_s - \delta e_s = -\phi^\top \zeta + [d_\alpha(X) + d_\beta(X)] \]  

(5.36)

Note that \( \dot{e} \) is a measure of the tracking performance, and will be used in the parameter hybrid update law (as we will show in later sections).

In summary, the measure of tracking performance \( \dot{e} \) for both direct and indirect cases can be written as

\[ \dot{e} := \frac{\kappa (\dot{e}_s + \delta e_s)}{\theta(X)} = -\phi^\top \zeta + d \]  

(5.37)

where the parameters for both direct and indirect cases are summarized in the table.

Next, we present a hybrid adaptive law that can be used for parameter adaptation.
Table 5.1: Summary of parameters

<table>
<thead>
<tr>
<th></th>
<th>( \kappa )</th>
<th>( \theta(X) )</th>
<th>( d )</th>
<th>( \phi(t) )</th>
<th>( \zeta(t) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct</td>
<td>1</td>
<td>( \beta(X) )</td>
<td>( d_\alpha(X) )</td>
<td>( \phi_u(t) )</td>
<td>( \zeta_u(t) )</td>
</tr>
<tr>
<td>Indirect</td>
<td>-1</td>
<td>1</td>
<td>( d_\alpha(X) + d_\beta(X) )</td>
<td>( \phi_\alpha(t)^\top, \phi_\beta(t)^\top )</td>
<td>( \left[ \zeta_\alpha(t)^\top, \zeta_\beta(t)^\top \right]^\top u_i(t) )</td>
</tr>
</tbody>
</table>

5.3 Hybrid Adaptive Law: Update and Stability

Consider the adaptive law

\[
\dot{\theta} = \eta \epsilon \zeta
\]  

(5.38)

where the \( \dot{\theta} \) is the derivative of the parameter vector with respect to time, \( \eta > 0 \) is a scalar adaptation gain, and \( \epsilon := \frac{\dot{\epsilon}}{m^2} \). As defined in [4], \( m^2 = 1 + n_\theta^2 \), and \( m \) is designed so that \( \frac{\epsilon}{m} \) and \( \frac{d(X)}{m} \in L_\infty \), and \( n_\theta \) is chosen such that \( \frac{|\epsilon|}{m} \leq 1 \). A typical choice for \( n_\theta^2 \) is \( n_\theta^2 = \gamma \zeta^\top \zeta \), where \( \gamma \geq 1 \). The adaptive law (5.38) is usually used with systems that have no modeling error [4]. In the presence of modeling error, however, a leakage modification is often used. The idea behind leakage is to modify the adaptive law (5.38) so that the time derivative of the Lyapunov function used to analyze the adaptive scheme becomes negative in the space of the parameter estimates when these parameters exceed certain bounds [4]. One way to solve this problem is to modify the adaptive law (5.38) as follows

\[
\dot{A} = \eta \epsilon \zeta - w \eta A
\]  

(5.39)

where \( w \) (which will be defined later) is a positive scalar signal (i.e., \( w(t) \geq 0 \)) that is designed so that stability is maintained if the parameter error exceeds a certain bound.

114
In the update law (5.39), the parameter vector $A$ is updated continuously with time, such that at every instant of time, $t$, we have a new estimate of the parameter vector. In many cases, it is desirable to update the estimate at specific instants of time $t_k$, where $\{t_k\}$ is an unbounded monotonically increasing sequence in $\mathbb{R}^+$. Let $t_k = kT_s$ where $T_s = t_{k+1} - t_k$ is the sampling period, and $k \in \mathcal{N}^+$ (i.e., $k = 0, 1, \ldots$).

To derive the hybrid adaptive law, integrate the continuous adaptive law (5.39) from some time instant $t_k = kT_s$ to the subsequent time instant $t_{k+1} = (k + 1)T_s$ to have

$$A(k + 1) = A(k) + \eta \int_{t_k}^{t_{k+1}} [\epsilon(\tau)\zeta(\tau) - w(\tau)A(k)] d\tau \quad (5.40)$$

where $A(k) := A(t_k)$. Note that the adaptive law (5.40) generates a sequence of estimates $A(k) = A(kT_s)$, for $k = 0, 1, 2, \ldots$. Although $\epsilon(t)$ and $\zeta(t)$ may change over time, the estimate $A(k)$ is constant for $t \in [t_k, t_{k+1})$.

As mentioned earlier, stability and boundedness results that can be obtained using the hybrid adaptive law (5.40) is dependent on the choice of the parameter $w(t)$. In [4], some stability results have been established for the "switching $\sigma$-modification." The choice of $w(t)$ in the switching $\sigma$-modification is defined as $w(t) = \sigma_s$, where

$$\sigma_s = \begin{cases} 0 & \text{if } |A(k)| < M_0 \\ \sigma_0 & \text{if } |A(k)| \geq M_0, \end{cases} \quad (5.41)$$

$\sigma_0 > 0$, and $M_0 \geq 2|A^*|$. Note that the adaptive law (5.39) has actually been analyzed in [4] for three different choices of the leakage term $w(t)$. These choices are the $\sigma$-modification, the switching-$\sigma$ modification, and the $\epsilon$-modification. The authors in [4] have shown that, unlike the $\sigma$-modification and the $\epsilon$-modification, the switching-$\sigma$ modification is able to achieve robustness without having to destroy some important properties (i.e., $\epsilon, \epsilon m, \dot{A} \in L_2$) of the adaptive law. Also, the selection of a discontinuous $\sigma_s$ (5.41) fits the discrete-time nature of the adaptive law (5.40). For
a more detailed analysis on the choices of $w(t)$ refer to [4]. Next, we will show the
stability properties that are established by the hybrid adaptive law (5.40). However,
before we start the theorem, let us state the following definition.

**Definition 3:** Let $x : [0, \infty) \rightarrow \mathbb{R}^n$, where $x \in \mathcal{L}_{2e}$ (the $\mathcal{L}_{2e}$ norm is defined as $\|x(t)\|_2 := \left( \int_0^t |x(\tau)|^2 d\tau \right)^{\frac{1}{2}}$, and we say that $x(t) \in \mathcal{L}_{2e}$ when $\|x(t)\|_2$ exists for any
finite $t$), and consider the set

$$
\mathcal{S}(\mu) = \{ x : [0, \infty) \rightarrow \mathbb{R}^n \mid \int_0^{t+T_s} x^\top(\tau)x(\tau)d(\tau) \leq c_0\mu T_s + c_1. \forall t. T_s \geq 0 \} \quad (5.42)
$$

for a given constant $\mu \geq 0$, where $c_0, c_1 \geq 0$ are some finite constants, and $c_0$ is
independent of $\mu$. If $x \in \mathcal{S}$, we say that $x$ is $\mu$-small in the mean square sense.

**Theorem 8:** Let $m, \sigma_0, T_s, \eta$ be chosen so that

- $\frac{d}{m} \in \mathcal{L}_{\infty}, \frac{c}{m^2} \leq 1$
- $2T_s\eta < 1, 2\sigma_0 T_s \eta < 1$

Then the hybrid adaptive law (5.40) guarantees that

1. $\epsilon, \epsilon n_s \in \mathcal{L}_{\infty}, A(k) \in l_{\infty}$ (for a sequence $x = (x_1, x_2, \ldots)$ and $x_i \in \mathcal{R}$ where
   $i \geq 1$. the $l_{\infty}$ norm is defined as $\|x\|_{\infty} := \sup_{i \geq 1} |x_i|$. We say that $x \in l_{\infty}$ if
   $\|x\|_{\infty}$ exists)
2. $\epsilon, \epsilon m \in \mathcal{S}(\frac{d^2}{m^2})$

**Proof:** Consider the following Lyapunov-like function

$$
V(k) = \phi(k)^\top \eta^{-1} \phi(k) \quad (5.43)
$$
Since $\mathcal{A}(k+1) - \mathcal{A}^* = \mathcal{A}(k) - \mathcal{A}^* + \mathcal{A}(k+1) - \mathcal{A}(k)$, we know that

$$\phi(k+1) = \phi(k) + \Delta \mathcal{A}(k)$$

(5.44)

where

$$\Delta \mathcal{A}(k) = \eta \int_{t_k}^{t_{k+1}} [e(\tau)\zeta(\tau) - w(\tau)\mathcal{A}(k)] \, d\tau$$

(5.45)

Using (5.44) in (5.43), we can write

$$V(k+1) = [\phi(k) + \Delta \mathcal{A}(k)]^\top \eta^{-1} [\phi(k) + \Delta \mathcal{A}(k)]$$

$$= \phi(k)^\top \eta^{-1} \phi(k) + 2\phi(k)^\top \eta^{-1} \Delta \mathcal{A}(k) + \Delta \mathcal{A}(k)^\top \eta^{-1} \Delta \mathcal{A}(k)$$

$$= V(k) + 2\phi(k)^\top \eta^{-1} \Delta \mathcal{A}(k) + \Delta \mathcal{A}(k)^\top \eta^{-1} \Delta \mathcal{A}(k)$$

(5.46)

Substituting (5.45) in (5.46), we get

$$V(k+1) = V(k) + 2\phi(k)^\top \int_{t_k}^{t_{k+1}} [e(\tau)\zeta(\tau) - w(\tau)\mathcal{A}(k)] \, d\tau$$

$$+ \int_{t_k}^{t_{k+1}} [e(\tau)\zeta(\tau) - w(\tau)\mathcal{A}(k)]^\top \eta \int_{t_k}^{t_{k+1}} [e(\tau)\zeta(\tau) - w(\tau)\mathcal{A}(k)] \, d\tau$$

(5.47)

Using (5.37) and the definition that $\epsilon = \frac{\epsilon}{\eta}$, it can be easily shown that

$$\phi(k)^\top \zeta = -\epsilon m^2 + d$$

(5.48)

or

$$\phi(k)^\top \epsilon \zeta = -\epsilon^2 m^2 + \epsilon d$$

(5.49)

Integrating both sides of (5.49), we get

$$\phi(k)^\top \int_{t_k}^{t_{k+1}} \epsilon(\tau)\zeta(\tau) \, d\tau = \int_{t_k}^{t_{k+1}} [-\epsilon^2(\tau)m^2(\tau) + \epsilon(\tau)d(\tau)] \, d\tau$$

(5.50)

We know that $-a^2 + ab \leq -\frac{a^2}{2} + \frac{b^2}{2}$. Let $a = \epsilon m$ and $b = \frac{d}{m}$. Hence, (5.50) can be written as

$$\phi(k)^\top \int_{t_k}^{t_{k+1}} \epsilon(\tau)\zeta(\tau) \, d\tau \leq - \int_{t_k}^{t_{k+1}} \frac{\epsilon^2(\tau)m^2(\tau)}{2} \, d\tau + \int_{t_k}^{t_{k+1}} \frac{d^2(\tau)}{2m^2(\tau)} \, d\tau$$

(5.51)
Consider the last term in (5.47) and note that

\[
\int_{t_k}^{t_{k+1}} [\epsilon(\tau)\zeta(\tau) - w(\tau)A(k)]^T d\tau \eta \int_{t_k}^{t_{k+1}} [\epsilon(\tau)\zeta(\tau) - w(\tau)A(k)] d\tau
= \eta \int_{t_k}^{t_{k+1}} [\epsilon(\tau)\zeta(\tau) - w(\tau)A(k)] d\tau^2
\]  

(5.52)

Since \((a + b)^2 \leq 2a^2 + 2b^2\), where \(a = \int_{t_k}^{t_{k+1}} \epsilon(\tau)\zeta(\tau) d\tau\) and \(b = \int_{t_k}^{t_{k+1}} w(\tau)A(k) d\tau\), then

\[
\left| \int_{t_k}^{t_{k+1}} [\epsilon(\tau)\zeta(\tau) - w(\tau)A(k)] d\tau \right|^2 \leq 2 \left| \int_{t_k}^{t_{k+1}} \epsilon(\tau)m(\tau) \frac{\zeta(\tau)}{m(\tau)} d\tau \right|^2 + 2 \left| \int_{t_k}^{t_{k+1}} w(\tau)A(k) d\tau \right|^2
\]  

(5.53)

Also, since \(\frac{\zeta(\tau)}{m(\tau)} \leq 1\), \(w(t) \leq \sigma^2\), and \(\int_{t_k}^{t_{k+1}} |A(k)|^2 d\tau = |A(k)|^2 T_s^2\), then

\[
2\eta \left| \int_{t_k}^{t_{k+1}} \epsilon(\tau)m(\tau) \frac{\zeta(\tau)}{m(\tau)} d\tau \right|^2 + 2\eta \left| \int_{t_k}^{t_{k+1}} w(\tau)A(k) d\tau \right|^2 \leq 2\eta T_s \int_{t_k}^{t_{k+1}} \epsilon^2(\tau)m^2(\tau) d\tau + 2\eta \sigma_s^2 T_s^2 |A(k)|^2
\]  

(5.54)

Hence, from (5.52) and (5.54), we can say that

\[
\int_{t_k}^{t_{k+1}} [\epsilon(\tau)\zeta(\tau) - w(\tau)A(k)]^T d\tau \eta \int_{t_k}^{t_{k+1}} [\epsilon(\tau)\zeta(\tau) - w(\tau)A(k)] d\tau \leq 2\eta T_s \int_{t_k}^{t_{k+1}} \epsilon^2(\tau)m^2(\tau) d\tau + 2\eta \sigma_s^2 T_s^2 |A(k)|^2
\]  

(5.55)

Using (5.51) and (5.55) in (5.47), we have

\[
V(k + 1) \leq V(k) - \int_{t_k}^{t_{k+1}} \epsilon^2(\tau)m^2(\tau) d\tau + \int_{t_k}^{t_{k+1}} \frac{d^2(\tau)}{m^2(\tau)} d\tau - 2\sigma_s T_s \phi(k)^T A(k)
+ 2\eta T_s \int_{t_k}^{t_{k+1}} \epsilon^2(\tau)m^2(\tau) d\tau + 2\eta \sigma_s^2 T_s^2 |A(k)|^2
\]  

(5.56)

Rearranging terms, (5.56) becomes

\[
V(k + 1) \leq V(k) - [1 - 2\eta T_s] \int_{t_k}^{t_{k+1}} \epsilon^2(\tau)m^2(\tau) d\tau + \int_{t_k}^{t_{k+1}} \frac{d^2(\tau)}{m^2(\tau)} d\tau
- 2\sigma_s T_s \left[ \phi(k)^T A(k) - \eta \sigma_s T_s |A(k)|^2 \right]
\]  

(5.57)
Using the fact that $\|A^*\| \geq \|A(k)\|$, we know that

\[ \phi(k)^T A(k) = [A(k) - A^*]^T A(k) \]
\[ \geq \|A(k)\|^2 - \|A^*\| \|A(k)\| \]
\[ \geq \|A(k)\|^2 - \|A^*\|^2 \]
\[ \geq \frac{\|A(k)\|^2}{2} - \frac{\|A^*\|^2}{2} \] (5.58)

Using (5.58) in (5.57), we have

\[ V(k+1) \leq V(k) - [1 - 2\eta T_s] \int_{t_k}^{t_{k+1}} \epsilon^2(\tau) m^2(\tau) d\tau + \int_{t_k}^{t_{k+1}} \frac{d^2(\tau)}{m^2(\tau)} d\tau \]
\[ - 2\sigma_s T_s \left[ \frac{\|A(k)\|^2}{2} - \frac{\|A^*\|^2}{2} - \eta \sigma_s T_s \|A(k)\|^2 \right] \] (5.59)

or

\[ V(k+1) \leq V(k) - [1 - 2\eta T_s] \int_{t_k}^{t_{k+1}} \epsilon^2(\tau) m^2(\tau) d\tau + \int_{t_k}^{t_{k+1}} \frac{d^2(\tau)}{m^2(\tau)} d\tau \]
\[ - 2\sigma_s T_s \left[ \left( \frac{1}{2} - \eta \sigma_0 T_s \right) \|A(k)\|^2 - \frac{\|A^*\|^2}{2} \right] \] (5.60)

Let $D_m = \sup_{t \in [t_k]} d \in L_\infty$. Since $D_m$ is constant over the time $[t_k, t_{k+1}]$, then (5.59) becomes

\[ V(k+1) \leq V(k) - [1 - 2\eta T_s] \int_{t_k}^{t_{k+1}} \epsilon^2(\tau) m^2(\tau) d\tau + T_s |D_m|^2 \]
\[ - 2\sigma_s T_s \left[ \left( \frac{1}{2} - \eta \sigma_0 T_s \right) \|A(k)\|^2 - \frac{\|A^*\|^2}{2} \right] \] (5.61)

Choose the parameters $\eta$, $T_s$, and $\sigma_0$ such that

\[ 1 - 2\eta T_s > 0 \] (5.62)

and

\[ \frac{1}{2} - \eta \sigma_0 T_s > 0 \] (5.63)

119
are both satisfied.

To show that \( A(k) \in l_\infty \), we need to consider two cases: the case when \(|A(k)| < M_0 \) and the case when \(|A(k)| \geq M_0 \). In the first case (when \(|A(k)| < M_0 \) for some bounded parameter \( M_0 \)), it clear that \( A(k) \in l_\infty \). In the other case (when \(|A(k)| \geq M_0 \) and hence \( \sigma_s = \sigma_0 \)), however, we consider (5.61). In (5.61) (when (5.62) and (5.63) are both satisfied), we can guarantee that \( V'_{k+1} \leq V'_{k} \) whenever

\[
T_s |D_m|^2 - 2\sigma_s T_s \left[ \frac{1}{2} - \eta \sigma_0 T_s \right] |A(k)|^2 + \sigma_s T_s |A^*|^2 \leq 0
\]

(5.64)

or since \( \sigma_s = \sigma_0 \)

\[
|A(k)|^2 \geq \frac{|D_m|^2 + \sigma_0 |A^*|^2}{\sigma_0 [1 - 2\eta \sigma_0 T_s]}
\]

(5.65)

Since in this case we know that \(|A(k)| \geq M_0 \), we can guarantee that \( V'_{k+1} \leq V'_{k} \) whenever

\[
|A(k)|^2 \geq \max \left\{ M_0^2, \frac{|D_m|^2 + \sigma_0 |A^*|^2}{\sigma_0 [1 - 2\eta \sigma_0 T_s]} \right\}
\]

(5.66)

Note. here, that \( M_0 \) is assumed to be large enough (e.g., \( M_0 \geq 1 \)) such that \( M_0 \geq |A^*| \). We can conclude from (5.66) (since all of its parameters are bounded) that \( V'_{k+1} \leq V'_{k} \). Since \( V(k) \) is defined as \( V(k) = \phi(k)^{T} \eta^{-1} \phi(k) \), then it can be shown (for some bounded parameter initial condition, \( A(0) \in l_\infty \)) that \( A(k) \in l_\infty \).

Since all parameters in the hybrid adaptive law (5.40) are bounded, then we can conclude that \( \epsilon \) and hence \( \epsilon m \) are bounded. This establishes the first part of the theorem.

To establish the second part, consider

\[
\sigma_s \phi^{T}(k) A(k) = \sigma_s A(k)^{T} A(k) - \sigma_s A^*^{T} A(k)
\]

\[
= \frac{\sigma_s}{2} A(k)^{T} A(k) + \left[ \frac{\sigma_s}{2} A(k)^{T} A(k) - \sigma_s A^*^{T} A(k) \right]
\]

120
Since \( M_0 \geq 2|A^*| \), (5.67) becomes

\[
\sigma_s A^T(k)A(k) \geq \frac{\sigma_s}{2} |A(k)|^2 + \frac{\sigma_s}{2} |A(k)| [||A(k)|| - M_0] \tag{5.68}
\]

Also, using (5.41) in (5.68), we get

\[
\sigma_s A^T(k)A(k) \geq \frac{\sigma_s}{2} |A(k)|^2 \tag{5.69}
\]

Adding and subtracting \( \sigma_0 \eta T_s \) from the right hand side of (5.69), and using the fact that \( 2\sigma_0 \eta T_s < 1 \), (5.69) becomes

\[
\sigma_s A^T(k)A(k) \geq \sigma_s \left[ \frac{1}{2} + \eta \sigma_0 T_s - \eta \sigma_0 T_s \right] |A(k)|^2 \tag{5.70}
\]

Defining \( c_\sigma = \frac{1}{2} - \eta \sigma_0 T_s \), (5.70) can be written as

\[
\sigma_s \left[ \phi^T(k)A(k) - \eta \sigma_0 T_s |A(k)|^2 \right] \geq \sigma_s c_\sigma |A(k)|^2 \tag{5.71}
\]

Rearranging terms, (5.57) can be written as

\[
[1 - 2\eta T_s] \int_{t_k}^{t_{k+1}} \epsilon^2(\tau) m^2(\tau) d\tau + 2\sigma_s T_s \left[ \phi(k)^T A(k) - \eta \sigma_s T_s |A(k)|^2 \right] \leq V(k) - V(k + 1) + \int_{t_k}^{t_{k+1}} \frac{d^2(\tau)}{m^2(\tau)} d\tau \tag{5.72}
\]

and using (5.71), (5.72) becomes

\[
[1 - 2\eta T_s] \int_{t_k}^{t_{k+1}} \epsilon^2(\tau) m^2(\tau) d\tau + 2\sigma_s T_s c_\sigma |A(k)|^2 \leq V(k) - V(k + 1) + \int_{t_k}^{t_{k+1}} \frac{d^2(\tau)}{m^2(\tau)} d\tau \tag{5.73}
\]

which implies that \( \epsilon m \in \mathcal{S}(\frac{d^2}{m^2}) \). Since \( |\epsilon| \leq |\epsilon m| \) (because \( m \geq 1 \)), we conclude that \( \epsilon \in \mathcal{S}(\frac{d^2}{m^2}) \). \( \blacksquare \)
Now, we will state the stability results for continuous-time direct and indirect adaptive control schemes when the hybrid adaptive law is used.

**Theorem 9:** Given the error dynamics (5.37) with the reference trajectory assumption (Assumption 9) satisfied, and either Assumption 6 or 7 holds (and for the direct case, Assumption 8 holds), then the hybrid adaptive law in both direct and indirect cases will ensure (in addition to the results stated in Theorem 8) that

1. \( e_s \) is bounded.

2. The plant output and its derivatives \( y_p, \ldots, y_p^{(r-1)} \) are bounded.

3. The control signal (\( u_d \) in the direct case or \( u_i \) in the indirect case) is bounded.

**Proof:**

**Part 1**

From Theorem 8 we know that

\[
\epsilon = \frac{\kappa (\dot{e}_s + \delta e_s)}{\theta m^2} \in \mathcal{S} \left( \frac{d^2}{m^2} \right)
\]  

(5.74)

Rearranging terms, (5.74) can be written as

\[
\dot{e}_s + \delta e_s = \frac{\epsilon \theta m^2}{\kappa}
\]  

(5.75)

The differential equation shown in (5.75) is a first order non-homogeneous differential equation, and its solution can be written as the sum of the homogeneous solution and the particular one (i.e., \( e_s = (e_s)_h + (e_s)_p \), where \( (e_s)_h \) and \( (e_s)_p \) are the homogeneous and particular solutions, respectively). It can be shown that the homogeneous solution has the form

\[
(e_s)_h = ce^{-\delta t}
\]  

(5.76)
where $c_\varepsilon$ is a constant that depends on the initial condition, and the particular solution is

$$ (e_\varepsilon)_p = \frac{\epsilon \theta m^2}{\kappa \delta} \tag{5.77} $$

Hence, $e_\varepsilon$ can be expressed as

$$ e_\varepsilon = c_\varepsilon e^{-\delta t} + \frac{\epsilon \theta m^2}{\kappa \delta} \tag{5.78} $$

Since $\delta > 0$, $c_\varepsilon e^{-\delta t}$ decreases exponentially as time increases. Hence

$$ \lim_{t \to \infty} e_\varepsilon = \frac{\epsilon \theta m^2}{\kappa \delta} \tag{5.79} $$

We know from Theorem 8 that both of $\epsilon$ and $\epsilon m \in \mathcal{S}\left(\frac{\delta^2}{m^2}\right)$. This implies that $m$ is bounded. Since all parameters in the right hand side of (5.79) are bounded, $e_\varepsilon$ is bounded.

**Part 2**

Hence, $|e_\varepsilon|$ is bounded by some upper bound $M_c (d, m)$ (i.e., $|e_\varepsilon| \leq M_c$). The following analysis, to show that the output error and its derivatives are bounded provided that $|e_\varepsilon| \leq M_c$, has been discussed in [32]. Define transfer functions

$$ \hat{G}_i(s) := \frac{y_i(s)}{L(s)}, \quad i = 0, \ldots, r - 1 \tag{5.80} $$

which are stable since $\hat{L}(s)$ has its poles in the open left half plane. Since $e_\varepsilon^{(i)} = \hat{G}_i(s)e_\varepsilon$, with $e_\varepsilon$ bounded, then $e_\varepsilon^{(i)}$ is bounded (i.e., $e_\varepsilon^{(i)} \in \mathcal{L}_\infty$). Since $e_\varepsilon^{(i)}$ is bounded and $e_\varepsilon^{(i)} = y_m^{(i)} - y_p^{(i)}$, and $y_m^{(i)}$ is bounded (by Assumption 9), then the output and its derivatives are bounded ($y_p^{(i)} \in \mathcal{L}_\infty, \forall i = 0, 1, \ldots, r - 1$). This establishes the second part of the theorem.
Part 3

Since the output and its derivatives are bounded, using assumption 6 or 7 we know that the states of the plant are bounded. Hence, in the indirect case the functions \( \alpha(X), \alpha_k(t), \beta(X), \beta_k(t) \in \mathcal{L}_\infty \). The projection algorithm ensures that \( \beta_k(t) + \dot{\beta}(X) \) is bounded away from zero and that \( \dot{\alpha}(X) \) is bounded, thus \( u_i \in \mathcal{L}_\infty \). In the direct case, since the states are bounded then \( \zeta \) is bounded. Also, we can use a projection algorithm to ensure that \( \mathcal{A} \in \mathcal{L}_\infty \). Hence, from the definition of the control \( u_d \) (5.13), we know that \( u_d \) is bounded (i.e., \( u_d \in \mathcal{L}_\infty \)). This establishes the third part of the theorem.

5.4 Auto-Tuning the Adaptation Gain

In this section, we will present a methodology to auto-tune the adaptation gain for a continuous-time nonlinear adaptive control systems when a gradient-based hybrid adaptive law is used for parameter adaptation. The gradient update law relies on the following idea. Starting with an initial value for the parameter vector, the gradient algorithm changes (updates) this vector by adding to it another vector having a magnitude and a direction of descent. We can think of this as searching for the ideal parameter vector. In most adaptive schemes, the adaptation gain is held constant. Here, however, we argue that the adaptation gain can be selected (adapted) on-line to minimize the instantaneous control energy. It is important to mention that our objective here is to search for an "optimal" \( \eta \) (that we will call \( \eta^{opt} \)). Note that \( \eta^{opt} \) is not necessarily the optimal adaptation gain. The step of finding \( \eta^{opt} \) is crucial to find the new parameter vector \( (A^{opt}(k)) \), and hence the new control, \( u^{opt}(t) \). The
term *optimal* is used here only because the adaptation gain (as shown below) will be selected to minimize the instantaneous control energy $J_u(\eta) = u^2(t)$.

We would like to note that the adaptation gain $\eta(k)$ is fixed over the interval $[t_k, t_{k+1})$.

### 5.4.1 Auto-Tuning Algorithm

The adaptation gain tuning algorithm (for both direct and indirect adaptive cases) proceeds according to the following steps (shown in Figure 5.1):

1. Find a range on $\eta(k)$ (i.e., $\eta \in [\eta_{\text{min}}, \eta_{\text{max}}]$), such that the stability is maintained no matter which $\eta(k)$ in this range is used.

2. Find the new adaptation gain ($\eta^{\text{opt}}(k)$) that minimizes the instantaneous control energy $J_u(\eta) = u^2(t)$.

3. Using $\eta^{\text{opt}}(k)$, find the new parameter vector $A^{\text{opt}}(k)$ and hence the new control $u^{\text{opt}}(t)$.

![Figure 5.1: Steps used for adaptation gain selection.](image-url)
Finding a Feasible Range on $\eta(k)$

Recall from Theorem 8 that the parameters $m, \sigma_0, T_s, \eta(k)$ need to be chosen so that

$$2T_s\eta(k) < 1$$  \hspace{1cm} (5.81)

or

$$\eta(k) < \frac{1}{2T_s}$$  \hspace{1cm} (5.82)

and

$$2\sigma_0 T_s \eta(k) < 1$$  \hspace{1cm} (5.83)

or

$$\eta(k) < \frac{1}{2\sigma_0 T_s}$$  \hspace{1cm} (5.84)

To satisfy both conditions (5.82) and (5.84), we select $\eta(k)$ such that

$$\eta(k) < \min \left\{ \frac{1}{2T_s}, \frac{1}{2\sigma_0 T_s} \right\} := \bar{\eta}$$  \hspace{1cm} (5.85)

where $\bar{\eta}$ is an upper bound on the adaptation gain. Define $\eta(k) = \rho(k)\bar{\eta}$ and

$$[\eta_{\text{min}} = \rho_1\bar{\eta}] \leq [\eta(k) = \rho(k)\bar{\eta}] \leq [\rho_2\bar{\eta} = \eta_{\text{max}}],$$  \hspace{1cm} (5.86)

where $0 < \rho_1 \leq \rho(k) \leq \rho_2 < 1$ for fixed constants $\rho_1$ and $\rho_2$. Note that both $\eta(k)$ and $\rho(k)$ are constants over the interval $[t_k, t_{k+1})$.

Finding the New Adaptation Gain $\eta^{\text{opt}}(k)$ via Minimizing the Instantaneous Control Energy

Here, the new adaptation gain is obtained by minimizing the following cost function

$$\min \ J_u(\eta) = u^2(t)$$  \hspace{1cm} (5.87)
such that $\eta_{\text{min}} \leq \eta(k) \leq \eta_{\text{max}}$. Assume that the ideal control can be approximated by

$$u(t) = A^T(k)\zeta(t)$$

(5.88)

where $A(k)$ is constant over the interval $[t_k, t_{k+1})$. Substituting the hybrid adaptive law (5.40) into (5.88), we get

$$u(t) = A(k - 1)\zeta(t) + \eta(k) \left[ \int_{t_{k-1}}^{t_k} [e(\tau)\zeta(\tau) - w(\tau)A(k - 1)] d\tau \right] \zeta(t)$$  (5.89)

Let $\varphi_1(t) = \left[ \int_{t_{k-1}}^{t_k} [e(\tau)\zeta(\tau) - w(\tau)A(k - 1)] d\tau \right] \zeta(t)$ and $\varphi_2(t) = A(k - 1)^T\zeta(t)$, then $u(t)$ can be written as $u(t) = \varphi_1(t)\eta(k) + \varphi_2(t)$. Hence, $u'(t)$ can be written as

$$u'(t) = \varphi_1^T(t)\eta^2(k) + 2\varphi_1(t)\varphi_2(t)\eta(k) + \varphi_2^2(t).$$

Since $u'(t)$ is in quadratic form, the cost function (5.87) can be minimized as a quadratic programming problem with linear inequality constraint ($\eta_{\text{min}} \leq \eta(k) \leq \eta_{\text{max}}$). Since $\varphi_2^2(t)$ is independent of $\eta(k)$, it can be omitted from the cost function we want to minimize. Hence, the instantaneous control energy $u^2(t)$ (that we need to minimize to obtain the new adaptation gain) can be expressed as

$$\min_{\eta} u^2(t) = \min_{\eta} \varphi_1^2(t)\eta^2(k) + 2\varphi_1(t)\varphi_2(t)\eta(k)$$  (5.90)

$$s.t. \quad \eta_{\text{min}} \leq \eta(k) \leq \eta_{\text{max}}.$$

Since $\varphi_1^2(t)$ is positive definite, this problem (5.90) is known to have a unique global minimum, $\eta^{\text{opt}}(k)$, which can be used to find the new parameter vector and hence the new control. Now, this adaptation gain can be used in the update routine of the controller's parameter vector as shown next.
Finding the New Parameter Vector \(A^{opt}(k)\) and the New Control \(u^{opt}(t)\)

The new adaptation gain \(\eta^{opt}(k)\) can be used to find the new parameter vector \(A^{opt}(k)\) as follows

\[
A^{opt}(k) = A^{opt}(k - 1) + \eta^{opt}(k) \int_{t_{k-1}}^{t_k} [e(\tau)\zeta(\tau) - w(\tau)A^{opt}(k - 1)] \, d\tau \tag{5.91}
\]

This new parameter vector of the controller is used to find the new control as

\[
u^{opt}(t) = A^{opt^T}(k)\zeta(t) \tag{5.92}
\]

which is the final control to be input to the system.

5.4.2 Stability Analysis

Here, we will present the stability results when the adaptation gain is auto-tuned according to the algorithm presented above.

**Theorem 10:** Let \(m, \sigma_0, T_s, \eta(k)\) be chosen so that \(\frac{\sigma_0}{m} \in \mathcal{L}_\infty, \frac{\sigma_0}{m} \leq 1\), then the hybrid adaptive law (5.91) (when the adaptation gain is auto-tuned to minimize the instantaneous control energy) guarantees that

1. \(e, e_n \in \mathcal{L}_\infty, A(k) \in l_\infty\)
2. \(e, em \in \mathcal{S}\left(\frac{\sigma_0^2}{m^2}\right)\)

**Proof:** Consider the following Lyapunov-like function

\[
V(k) = \phi(k)^T \phi(k) \tag{5.93}
\]

Since \(A(k + 1) - A^* = A(k) - A^* + A(k + 1) - A(k)\), we know that

\[
\phi(k + 1) = \phi(k) + \Delta A(k) \tag{5.94}
\]

128
where
\[ \Delta A(k) = \rho(k + 1)\bar{\eta} \int_{t_k}^{t_{k+1}} [\epsilon(\tau)\zeta(\tau) - w(\tau)A(k)] \, d\tau \] (5.95)

Here, \( \rho(k)\bar{\eta} = \eta(k) \) represents any adaptation gain within the feasible range
\[ [\eta_{\text{min}} = \rho_1\bar{\eta}] \leq [\eta(k) = \rho(k)\bar{\eta}] \leq [\eta_{\text{max}} = \rho_2\bar{\eta}], \]

where \( 0 < \rho_1 \leq \rho \leq \rho_2 < 1 \). Using (5.91) in (5.93), we can write
\[
V(k + 1) = [\phi(k) + \Delta A(k)]^T [\phi(k) + \Delta A(k)]
\]
\[
= \phi(k)^T \phi(k) + 2\phi(k)^T \Delta A(k) + \Delta A^T(k) \Delta A(k)
\]
\[
= V(k) + 2\phi(k)^T \Delta A(k) + \Delta A^T(k) \Delta A(k)
\] (5.96)

Substituting (5.95) in (5.96), we get
\[
V'(k + 1) = V(k) + 2\phi(k)^T \rho(k + 1)\bar{\eta} \int_{t_k}^{t_{k+1}} [\epsilon(\tau)\zeta(\tau) - w(\tau)A(k)] \, d\tau
\]
\[
+ \int_{t_k}^{t_{k+1}} [\epsilon(\tau)\zeta(\tau) - w(\tau)A(k)]^T \, d\tau \rho(k + 1)\bar{\eta}^2 \int_{t_k}^{t_{k+1}} [\epsilon(\tau)\zeta(\tau) - w(\tau)A(k)] \, d\tau
\]
Using (5.37) and the definition that \( \epsilon = \frac{\hat{e}}{m^2} \), it can be easily shown that
\[
\phi(k)^T \zeta = -\epsilon m^2 + d
\] (5.98)

or
\[
\phi(k)^T \epsilon \zeta = -\epsilon^2 m^2 + \epsilon d
\] (5.99)

Integrating both sides of (5.99), we get
\[
\phi(k)^T \int_{t_k}^{t_{k+1}} \epsilon(\tau)\zeta(\tau) \, d\tau = \int_{t_k}^{t_{k+1}} \left[ -\epsilon^2(\tau)m^2(\tau) + \epsilon(\tau)d(\tau) \right] \, d\tau
\] (5.100)

We know that \(-a^2 + ab \leq -\frac{a^2}{2} + \frac{b^2}{2}\). Let \( a = \epsilon m \) and \( b = \frac{d}{m} \). Hence, (5.100) can be written as
\[
\phi(k)^T \int_{t_k}^{t_{k+1}} \epsilon(\tau)\zeta(\tau) \, d\tau \leq -\int_{t_k}^{t_{k+1}} \frac{\epsilon^2(\tau)m^2(\tau)}{2} \, d\tau + \int_{t_k}^{t_{k+1}} \frac{d^2(\tau)}{2m^2(\tau)} \, d\tau
\] (5.101)
Consider the last term in (5.97) and note that

\[
\int_{t_k}^{t_{k+1}} [\epsilon(\tau)\zeta(\tau) - w(\tau)A(k)]^\top d\tau \left[ \rho(k+1)\bar{\eta} \right]^2 \int_{t_k}^{t_{k+1}} [\epsilon(\tau)\zeta(\tau) - w(\tau)A(k)] d\tau
= \left[ \rho(k+1)\bar{\eta} \right]^2 \left| \int_{t_k}^{t_{k+1}} [\epsilon(\tau)\zeta(\tau) - w(\tau)A(k)] d\tau \right|^2
\]

(5.102)

Since \((a + b)^2 \leq 2a^2 + 2b^2\), where \(a = \int_{t_k}^{t_{k+1}} \epsilon(\tau)\zeta(\tau) d\tau\) and \(b = \int_{t_k}^{t_{k+1}} -w(\tau)A(k) d\tau\), then

\[
\left| \int_{t_k}^{t_{k+1}} [\epsilon(\tau)\zeta(\tau) - w(\tau)A(k)] d\tau \right|^2 \leq 2 \left| \int_{t_k}^{t_{k+1}} \epsilon(\tau)m(\tau)\frac{\zeta(\tau)}{m(\tau)} d\tau \right|^2 + 2 \left| \int_{t_k}^{t_{k+1}} w(\tau)A(k) d\tau \right|^2
\]

(5.103)

Also, since \(\frac{|\epsilon(t)|}{m(t)} \leq 1\), \(w(t) \leq \sigma_s^2\), and \(\int_{t_k}^{t_{k+1}} |A(k)|^2 d\tau = |A(k)|^2 T_s^2\), then

\[
2 \left| \int_{t_k}^{t_{k+1}} \epsilon(\tau)m(\tau)\frac{\zeta(\tau)}{m(\tau)} d\tau \right|^2 + 2 \left| \int_{t_k}^{t_{k+1}} w(\tau)A(k) d\tau \right|^2 \leq 2T_s \int_{t_k}^{t_{k+1}} \epsilon^2(\tau)m^2(\tau) d\tau + 2\left[ \rho(k+1)\bar{\eta} \right]^2 \sigma_s^2 T_s^2 |A(k)|^2
\]

(5.104)

Hence, from (5.102), (5.103), and (5.104), we can say that

\[
\int_{t_k}^{t_{k+1}} [\epsilon(\tau)\zeta(\tau) - w(\tau)A(k)]^\top d\tau \left[ \rho(k+1)\bar{\eta} \right]^2 \int_{t_k}^{t_{k+1}} [\epsilon(\tau)\zeta(\tau) - w(\tau)A(k)] d\tau \leq
\]

\[
2 \left[ \rho(k+1)\bar{\eta} \right]^2 T_s \int_{t_k}^{t_{k+1}} \epsilon^2(\tau)m^2(\tau) d\tau + 2\left[ \rho(k+1)\bar{\eta} \right]^2 \sigma_s^2 T_s^2 |A(k)|^2
\]

(5.105)

It can be shown that (5.97) can be written as

\[
V(k+1) \leq V(k) + 2\rho(k+1)\bar{\eta}\phi^\top(k) \int_{t_k}^{t_{k+1}} \epsilon(\tau)\zeta(\tau) d\tau
\]

\[
+ \int_{t_k}^{t_{k+1}} [\epsilon(\tau)\zeta(\tau) - w(\tau)A(k)]^\top d\tau \left[ \rho(k+1)\bar{\eta} \right]^2 \int_{t_k}^{t_{k+1}} [\epsilon(\tau)\zeta(\tau) - w(\tau)A(k)] d\tau
\]

\[
- 2\rho(k+1)\bar{\eta}\phi^\top(k) \int_{t_k}^{t_{k+1}} w(\tau)A(k) d\tau
\]

(5.106)

Using (5.101) and (5.105) in (5.106), we have

\[
V(k+1) \leq V(k) - \rho(k+1)\bar{\eta} \int_{t_k}^{t_{k+1}} \epsilon^2(\tau)m^2(\tau) d\tau + \rho(k+1)\bar{\eta} \int_{t_k}^{t_{k+1}} \frac{d^2(\tau)}{m^2(\tau)} d\tau
\]

\[
- 2\rho(k+1)\bar{\eta}\sigma_s T_s \phi(k) A(k) + 2\left[ \rho(k+1)\bar{\eta} \right]^2 T_s \int_{t_k}^{t_{k+1}} \epsilon^2(\tau)m^2(\tau) d\tau
\]

\[
+ 2\left[ \rho(k+1)\bar{\eta} \right]^2 \sigma_s^2 T_s^2 |A(k)|^2
\]

(5.107)
Rearranging terms, (5.107) becomes

\[
V(k + 1) \leq V(k) - \rho(k + 1) \tilde{\eta} [1 - 2\rho(k + 1) \tilde{\eta} T_s] \int_{t_k}^{t_{k+1}} \epsilon^2(\tau) m^2(\tau) d\tau
+ \rho(k + 1) \tilde{\eta} \int_{t_k}^{t_{k+1}} \frac{d^2(\tau)}{m^2(\tau)} d\tau
- \rho(k + 1) \tilde{\eta} \sigma_s T_s \left[ |\phi(k)^T A(k) - \rho(k + 1) \tilde{\eta} \sigma_s T_s |A(k)|^2 \right] (5.108)
\]

Using the fact that |A*| \geq |A(k)|, we know that

\[
\phi(k)^T A(k) = [A(k) - A^*]^T A(k)
= |A(k)|^2 - A^*^T A(k)
\geq |A(k)|^2 - |A*|^2 |A(k)|
\geq |A(k)|^2 - |A*|^2
\geq \frac{|A(k)|^2}{2} - \frac{|A*|^2}{2} (5.109)
\]

Using (5.109) in (5.108), we have

\[
V(k + 1) \leq V(k) - \rho(k + 1) \tilde{\eta} [1 - 2\rho(k + 1) \tilde{\eta} T_s] \int_{t_k}^{t_{k+1}} \epsilon^2(\tau) m^2(\tau) d\tau
+ \rho(k + 1) \tilde{\eta} \int_{t_k}^{t_{k+1}} \frac{d^2(\tau)}{m^2(\tau)} d\tau
- 2\rho(k + 1) \tilde{\eta} \sigma_s T_s \left[ \frac{|A(k)|^2}{2} - \frac{|A*|^2}{2} - \rho(k + 1) \tilde{\eta} \sigma_s T_s |A(k)|^2 \right] (5.110)
\]

or

\[
V(k + 1) \leq V(k) - \rho(k + 1) \tilde{\eta} [1 - 2\rho(k + 1) \tilde{\eta} T_s] \int_{t_k}^{t_{k+1}} \epsilon^2(\tau) m^2(\tau) d\tau
+ \rho(k + 1) \tilde{\eta} \int_{t_k}^{t_{k+1}} \frac{d^2(\tau)}{m^2(\tau)} d\tau
- 2\rho(k + 1) \tilde{\eta} \sigma_s T_s \left[ \left( \frac{1}{2} - \rho(k + 1) \tilde{\eta} \sigma_0 T_s \right) |A(k)|^2 - \frac{|A*|^2}{2} \right] (5.111)
\]
Let $D_m = \sup_{t_m} d \in L_\infty$. Since $D_m$ is constant over the time $[t_k, t_{k+1})$, then (5.111) becomes

$$V(k + 1) \leq V(k) - \rho(k + 1)\bar{\eta}[1 - 2\rho(k + 1)\bar{\eta}T_s] \int_{t_k}^{t_{k+1}} \epsilon^2(\tau)m^2(\tau)d\tau + T_s\rho(k + 1)\bar{\eta}|D_m|^2 - 2\rho(k + 1)\bar{\eta}\sigma_s T_s \left[\left(\frac{1}{2} - \rho(k + 1)\bar{\eta}\sigma_0 T_s\right)|\mathcal{A}(k)|^2 - \frac{|\mathcal{A}^*|^2}{2}\right]$$

(5.112)

From the definition (5.85), we know that

$$\rho(k + 1)\bar{\eta} < \bar{\eta} \leq \frac{1}{2T_s}$$

(5.113)

and

$$\rho(k + 1)\bar{\eta} < \bar{\eta} \leq \frac{1}{2\sigma_0 T_s}$$

(5.114)

From (5.113) and (5.114), we know that

$$2\rho(k + 1)\bar{\eta}T_s < 1$$

(5.115)

and

$$\rho(k + 1)\bar{\eta}\sigma_0 T_s < \frac{1}{2}$$

(5.116)

To show that $\mathcal{A}(k) \in l_\infty$, we need to consider two cases: the case when $|\mathcal{A}(k)| < M_0$ and the case when $|\mathcal{A}(k)| \geq M_0$. In the first case (when $|\mathcal{A}(k)| < M_0$ for some bounded parameter $M_0$), it clear that $\mathcal{A}(k) \in l_\infty$. In the other case (when $|\mathcal{A}(k)| \geq M_0$ and hence $\sigma_s = \sigma_0$), however, we consider (5.112). In (5.112) (when (5.115) and (5.116) are both satisfied), we can guarantee that $V(k + 1) \leq V(k)$ whenever

$$T_s\rho(k+1)\bar{\eta}|D_m|^2 - 2\sigma_s T_s\rho(k + 1)\bar{\eta}\left[\frac{1}{2} - \rho(k + 1)\bar{\eta}\sigma_0 T_s\right]|\mathcal{A}(k)|^2 + \rho(k + 1)\bar{\eta}\sigma_s T_s|\mathcal{A}^*|^2 \leq 0$$

(5.117)

or since $\sigma_s = \sigma_0$

$$|\mathcal{A}(k)|^2 \geq \frac{|D_m|^2 + \sigma_0|\mathcal{A}^*|^2}{\sigma_0[1 - 2\rho(k + 1)\bar{\eta}\sigma_0 T_s]}$$

(5.118)
Since in this case we know that $|A(k)| \geq M_0$, we can guarantee that $V(k+1) \leq V(k)$ whenever
\[ |A(k)|^2 \geq \max \left\{ M_0^2, \frac{|D_m|^2 + \sigma_0 A^*|^2}{\sigma_0 [1 - 2\sigma_0 T_s]} \right\} \quad (5.119) \]

Note, here, that $M_0$ is assumed to be large enough (e.g., $M_0 \geq 1$) such that $M_0 \geq |A^*|$. We can conclude from (5.119) (since all of its parameters are bounded) that $V(k+1) \leq V(k)$. Since $V(k)$ is defined as $V(k) = \sigma(k)^T \sigma(k)$, then it can be shown (for some bounded parameter initial condition. $\sigma(0) \in l_\infty$) that $\sigma(k) \in l_\infty$. Since all parameters in the hybrid adaptive law (5.91) are bounded, then we can conclude that $\epsilon$ and hence $em$ are bounded. This establishes the first part of the theorem.

To establish the second part, consider
\[
\sigma_s \phi^T(k) A(k) = \sigma_s A(k)^T A(k) - \sigma_s A^*^T A(k) \\
= \frac{\sigma_s}{2} A(k)^T A(k) + \left[ \frac{\sigma_s}{2} A(k)^T A(k) - \sigma_s A^*^T A(k) \right] \\
\geq \frac{\sigma_s}{2} |A(k)|^2 + \frac{\sigma_s}{2} [ |A(k)|^2 - 2|A^*||A(k)| ] \\
\geq \frac{\sigma_s}{2} |A(k)|^2 + \frac{\sigma_s}{2} |A(k)| (|A(k)| - 2|A^*|) \quad (5.120)
\]

Since $M_0 \geq 2|A^*|$, (5.120) becomes
\[
\sigma_s \phi^T(k) A(k) \geq \frac{\sigma_s}{2} |A(k)|^2 + \frac{\sigma_s}{2} |A(k)| (|A(k)| - M_0) \quad (5.121)
\]

Also, using (5.41) in (5.121), we get
\[
\sigma_s \phi^T(k) A(k) \geq \frac{\sigma_s}{2} |A(k)|^2 \quad (5.122)
\]

Adding and subtracting $\sigma_0 \rho(k+1) \eta T_s$ from the right hand side of (5.122), and using the fact that $2\sigma_0 \rho(k+1) \eta T_s < 1$, (5.122) becomes
\[
\sigma_s \phi^T(k) A(k) \geq \sigma_s \left[ \frac{1}{2} + \rho(k+1) \eta \sigma_0 T_s - \rho(k+1) \eta \sigma_0 T_s \right] |A(k)|^2 \quad (5.123)
\]

133
Defining \( c_\sigma = \frac{1}{2} - \rho(k + 1)\hat{\sigma}_0 T_s \), (5.123) can be written as

\[
\sigma_s \left[ \phi^T(k) \dot{A}(k) - \rho(k + 1)\hat{\sigma}_0 T_s |A(k)|^2 \right] \geq \sigma_s c_\sigma |A(k)|^2 \tag{5.124}
\]

Rearranging terms, (5.108) can be written as

\[
\rho(k + 1)\hat{\eta} [1 - 2\rho(k + 1)\hat{\eta} T_s] \int_{t_k}^{t_{k+1}} \varepsilon^2(\tau) m^2(\tau) d\tau + 2\rho(k + 1)\hat{\sigma}_s T_s |A(k)|^2 \\
\leq V'(k) - V'(k + 1) + \rho(k + 1)\hat{\eta} \int_{t_k}^{t_{k+1}} \frac{d^2(\tau)}{m^2(\tau)} d\tau 
\]

and using (5.124), (5.125) becomes

\[
\rho(k + 1)\hat{\eta} [1 - 2\rho(k + 1)\hat{\eta} T_s] \int_{t_k}^{t_{k+1}} \varepsilon^2(\tau) m^2(\tau) d\tau + 2\rho(k + 1)\hat{\sigma}_s T_s |A(k)|^2 \\
\leq V'(k) - V'(k + 1) + \rho(k + 1)\hat{\eta} \int_{t_k}^{t_{k+1}} \frac{d^2(\tau)}{m^2(\tau)} d\tau 
\]

which implies that \( \varepsilon m \in \mathcal{S}(\frac{d^2}{m^2}) \). Since \(|\varepsilon| \leq |\varepsilon m| \) (because \( m \geq 1 \)), we conclude that \( \varepsilon \in \mathcal{S}(\frac{d^2}{m^2}) \).

Now, we present the following theorem to show boundedness of all signals.

**Theorem 11:** Given the error dynamics (5.37) with the reference trajectory assumption (Assumption 9) satisfied, and either Assumption 6 or 7 holds (and for the direct case, Assumption 8 holds), then the hybrid adaptive law (5.91) in both direct and indirect cases will ensure (in addition to the results stated in Theorem 10) that

1. \( e_s \) is bounded.
2. The plant output and its derivatives \( y_p, \ldots, y_p^{(s-1)} \) are bounded.
3. The control signal (\( u_d \) in the direct case or \( u_i \) in the indirect case) is bounded.

**Proof:** The proof of this theorem is similar to the proof of Theorem 9.
5.5 Aircraft Wing Rock Example

Aircraft wing rock is a limit cycling oscillation in the aircraft roll angle $\phi$ and roll rate $\dot{\phi}$. Limit cycle roll and roll rate are experienced by aircraft with pointed forebodies at high angle of attack. Such phenomenon may present serious danger due to the potential of aircraft instability. If $\delta_A$ is the actuator output, a model of this phenomenon is given by

$$\ddot{\phi} = a_1 \phi + a_2 \dot{\phi} + a_3 \dot{\phi}^3 + a_4 \phi^2 \dot{\phi} + a_5 \phi \dot{\phi}^3 + b \delta_A$$  \hspace{1cm} (5.127)

Choose the state vector $x = [x_1, x_2, x_3]^T$ with $x_1 = \phi$, $x_2 = p$, and $x_3 = \delta_A$. Suppose that we use a first order model to represent the actuator dynamics of the aileron (the control surface at the outer part of the wing). Then we have

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = a_1 x_1 + a_2 x_2 + a_3 x_2^3 + a_4 x_1^2 x_2 + a_5 x_1 x_2^2 + bx_3$$

$$\dot{x}_3 = -\frac{1}{\tau} x_3 + \frac{1}{\tau} u$$

$$y = x_1$$  \hspace{1cm} (5.128)

where $u$ is the control input to the actuator and $\tau$ is the aileron time constant. For an angle of attack of 21.5 degrees, $a_1 = -0.0148927$, $a_2 = 0.0415424$, $a_3 = 0.01668756$, $a_4 = -0.06578382$, and $a_5 = 0.08578836$. Also, $b = 1.5$ and $\tau = \frac{1}{15}$. Here, we use an initial condition of $x(0) = [0.4, 0, 0]^T$. Here, the reference signal $y_m(t) = 0$. The above model was taken from [95, 96] and is based on wind tunnel data in [97]. The objective of this example is to demonstrate the auto-tuning algorithm presented earlier.

Based on the definition of the plant considered (5.1) we can show that

$$f(X) = \left[ a_1 x_1 + a_2 x_2 + a_3 x_2^3 + a_4 x_1^2 x_2 + a_5 x_1 x_2^2 + bx_3 \right] \cdot \left[ x_2 \atop -\frac{1}{\tau} x_3 \right]$$  \hspace{1cm} (5.129)

135
\[ g(X) = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\tau} \end{bmatrix}, \quad (5.130) \]

and

\[ h(X) = x_1. \quad (5.131) \]

Also, it can be verified that the relative degree of the system is \( r = n = 3 \) (no zero dynamics). It can be shown (according to (5.3)) that

\[ y^{(3)} = (\alpha_k(t) + \alpha(X)) + (\beta_k(t) + \gamma(X))u \quad (5.132) \]

where (assuming that \( \alpha_k(t) = \beta_k(t) = 0 \))

\[ \alpha(X) = a_1 x_2 + 2a_4 x_1 x_2^2 + a_5 x_3^2 - \frac{b}{\tau} x_3 + \left[ a_2 + 3a_3 x_2^2 + a_4 x_1^2 + 2a_5 x_1 x_2 \right] \times \left[ a_1 x_1 + a_2 x_2 + a_3 x_3^2 + a_4 x_1^2 x_2 + a_5 x_1 x_2^2 + b x_3 \right] \quad (5.133) \]

and

\[ \gamma(X) = \frac{b}{\tau}. \quad (5.134) \]

Here, it is assumed that there exists positive constants \( \beta_0 \) and \( \beta_1 \) such that \( 0 < \beta_0 \leq \gamma(X) \leq \beta_1 < \infty \), where in this case \( \beta_0 = 10 \) and \( \beta_1 = 40 \).

As defined earlier, our measure of the tracking performance is

\[ \epsilon = \frac{\kappa (\dot{e}_s + \delta e_s)}{\theta(X)m^2} \quad (5.135) \]

where \( e_s = k^\top e \) and \( k = [k_0, k_1, 1] = [100, 20, 1] \).

### 5.5.1 Direct Case

Here, we attempt to approximate the ideal controller by an approximator in the form of a Takagi-Sugeno fuzzy system (TSFS). The TSFS used here has nine rules, and it has two inputs to the premise of each rule, \( x_1 \) and \( x_2 \). Also, it has three inputs to
the consequent of each rule, \( x_1, x_2, \) and \( x_3 \). The certainties of the rules are determined by Gaussian membership functions whose centers are evenly distributed between \(-2\) and \(2\). The parameters of the TSFS are updated using the hybrid adaptive law

\[
A(k) = A(k - 1) + \eta(k) \int_{t_{k-1}}^{t_k} \left[ \epsilon(\tau) \zeta(\tau) - w(\tau)A^{opt}(k - 1) \right] d\tau
\]

(5.136)

After some tuning, we have found that we can obtain a small value of \( \epsilon \) (as shown in Figure 5.2) using \( T_s = 0.005, \sigma_0 = 1, M_0 = 1, \gamma = 1.8, \) and \( \delta = 30 \). The value of \( T_s \) is chosen to be 0.005 since small sampling time is needed to simulate such a continuous time system. To simplify the tuning procedure, \( \sigma_0 \) is chosen to be 1 so that

\[
\frac{1}{2T_s} = \frac{1}{2\sigma_0 T_s}
\]

and hence \( \tilde{\eta} \) defined in (5.85) becomes

\[
\tilde{\eta} = \min \left\{ \frac{1}{2T_s}, \frac{1}{2\sigma_0 T_s} \right\} = \frac{1}{2T_s} = \frac{1}{2\sigma_0 T_s}
\]

Based on theory, \( \gamma \) should be selected such that \( \gamma \geq 1 \). Here, we started with \( \gamma = 1 \) and we found by some tuning that an acceptable response can be achieved using \( \gamma = 1.8 \). Also, \( \delta \) which needs to be positive serves as a weighting between \( \epsilon_s \) and \( \dot{\epsilon}_s \) in the definition of \( \epsilon \). We started with \( \delta = 1 \). After some tuning, we have found that \( \delta = 30 \) is an acceptable choice. Based on the theory, \( M_0 \) has to be selected such that \( M_0 \geq 2|A^*| \). Since \( |A^*| \) is unknown, we initially selected \( M_0 \) to be some large positive scalar, and by some tuning we were able to decrease the magnitude of this scalar to \( M_0 = 1 \) such that we achieve some acceptable performance. Using (5.85), it can be easily shown that \( \tilde{\eta} = 100 \). Here, we selected \( \rho_1 \) and \( \rho_2 \) to be 0.05 and 0.95, respectively. This implies that the lower and upper bounds on the adaptation gain are 5 and 95, respectively. In the first plot of Figure 5.2, we show how \( \epsilon \) decreases.
to a small value. The second plot in the figure shows how the adaptation gain varies based on the variations of $\epsilon$. It is clear from the figure that the adaptation gain, in almost the first two seconds, increases to its upper bound since large adaptation gain is needed to derive $\epsilon$ to some small value. After the two seconds, the adaptation gain usually takes its lower bound since $\epsilon$ has small value over that time. However, at certain instants the adaptation gain starts to increase to its upper bound for relatively short time intervals. It is unclear from Figure 5.2 why the adaptation gain behaves in such manner after the first two seconds (when $\epsilon$ is relatively small). To investigate this observation, we consider Figure 5.3, where the first plot which is a scaled version of the first plot in Figure 5.2 shows the behavior of $\epsilon$ at a smaller range. It is clear
from this figure that $\epsilon$ exhibits small variations in magnitude. This shows (for this particular example) how sensitive the presented auto-tuning algorithm is for small variations in $\epsilon$. It is important to note that since $\epsilon \in \mathcal{S}(\frac{\sigma^2}{mt})$, $\epsilon$ can be made smaller by either decreasing $d^2$ (by improving the approximation accuracy) or by increasing $m^2$ (by increasing $\gamma$). The changes of the parameter vector of the TSFS are shown in Figure 5.4. It is clear the parameters are bounded as stated in the first part of Theorem 8. The response of the aircraft roll angle, $\phi$, is shown in the first plot of Figure 5.5. The second plot of this figure shows the behavior of the aileron input, $\delta_A$.

It is clear that the response of the aircraft roll angle is unacceptable with this set of

Figure 5.3: The responses of $\epsilon$ and $\eta$ in the direct case.
controller parameters. However, such results are expected since the objective of the adaptive control law is to drive $\varepsilon$ (not the tracking error, $e_0$) to a small value that is function of $d$ and $m$ (since $\varepsilon \in \mathcal{S}(\frac{d^2}{m^2})$). We know that $\varepsilon$ is defined as

$$\varepsilon = \frac{\kappa (\dot{e}_s + \delta e_s)}{\theta(X)m^2} \in \mathcal{S}(\frac{d^2}{m^2})$$

One way to decrease the tracking error is to try to make $e_s$ dominate the effects on the dynamics of $\varepsilon$ (by increasing the value of $\delta$), and hence $e_0$ will become smaller since $e_s$ is smaller. This is clear from the response of the aircraft roll angle shown in Figure 5.6 when $\delta$ is increased to 5000.
Figure 5.5: The responses of the aircraft roll angle and aileron input in the direct case for $\delta = 30$

Note that the algorithm presented earlier focuses on auto-tuning the adaptation gain by minimizing the control energy. For this reason, let us discuss how this algorithm impacts the resulting control energy. To do that, we need to investigate how the MSE and MCE change for different values over the feasible range of adaptation gain (which is in this case $0 < \eta < 100$). Figure 5.7 shows how both the MCE and MSE change for several fixed values of the adaptation gain over a simulation period of 10 seconds. The first plot in Figure 5.7 shows the changes in MCE for several values of the adaptation gain. The dotted line in this figure shows the value of the MCE when the auto-tuning algorithm is used. This value is found to be 0.2205. It
Figure 5.6: The responses of the aircraft roll angle and aileron input in the direct case for $\delta = 5000$

is clear that MCE (except at very small values of $\eta$) slightly oscillates around 0.9. The large MCE values at small fixed values of $\eta$ can be due to the large error that may result when small fixed values of $\eta$ are used. To decrease such large error, a relatively large control energy is often needed. It is clear that the MCE achieved when the auto-tuning algorithm is used is smaller than the MCE obtained using any fixed adaptation gain. Similarly, the second plot in the figure shows the changes in MSE for several fixed values of the adaptation gain, and the dotted line shows the value of the MSE when the auto-tuning algorithm is used. This value is found to be 0.0138. The third plot is only a scaled version of the second plot to clarify the
Figure 5.7: Changes in the MCE and MSE as $\eta$ varies in the direct adaptive case.

variation of MSE at large values of the adaptation gain. It is clear from the figure that the MSE decreases as the adaptation gain increases. This decrease is due to the relatively large control energy (compared to the MCE obtained using the auto-tuning algorithm) that improves the closed-loop performance. The MSE obtained using the auto-tuning algorithm is found to be larger than almost any MSE value obtained at a fixed adaptation gain. This is due to the fact that in the auto-tuning algorithm the adaptation gain is obtained to minimize the control energy at the expense of error
energy. Hence, we can conclude that our simulation results support the objective of
the presented algorithm in the sense that the adaptation gain is selected on-line to
minimize the control energy in such a way that a good closed-loop performance is
achieved.

5.5.2 Indirect Case

Here, we attempt to approximate parts of the plant dynamics (i.e., \( \alpha(X) \) and
\( J(X) \)) and use these estimates to find the control. The function \( \alpha \) is approximated
here by an approximator in the form of a Takagi-Sugeno fuzzy system (TSFS) that
has nine rules. This TSFS has two inputs to the premise of each rule, \( x_1 \) and \( x_2 \). Also,
it has three inputs to the consequent of each rule, \( x_1, x_2, \) and \( x_3 \). The certainties of
the rules are determined by Gaussian membership functions whose centers are evenly
distributed between \(-2\) and \(2\). The function \( J(X) \), however, is approximated by a
scalar. The parameters of both approximators are updated using the hybrid adaptive
law \((5.136)\). After some tuning, we have found that we can obtain a small value of \( \epsilon \)
as shown in Figure 5.8) using \( T_s = 0.005, \sigma_0 = 1, M_0 = 200, \gamma = 3, \) and \( \delta = 3 \).

\( T_s \) and \( \sigma_0 \) are chosen to be 0.005 and 1, respectively, for the same reason stated
in the direct case. Also, since \( \gamma \) should be selected such that \( \gamma \geq 1 \), we started
with \( \gamma = 1 \), and we found by some tuning that good response can be achieved using
\( \gamma = 3 \). As in the direct case, \( \delta \) needs to be positive; we started with \( \delta = 1 \), and after
some tuning we have found that \( \delta = 3 \) is an acceptable choice. Also, \( M_0 \) has to be
selected such that \( M_0 \geq 2|A^*| \). Since \(|A^*| \) is unknown, we initially \( M_0 \) to be some
large positive scalar, and by some tuning we have found that \( M_0 = 200 \) provides an
acceptable performance. Using \((5.85)\), it can be easily shown that \( \bar{\eta} = 100 \). Here, we
selected $\rho_1$ and $\rho_2$ to be 0.05 and 0.95, respectively. This implies that the lower and upper bounds on the adaptation gain are 5 and 95, respectively. The responses of $\varepsilon$ and $\eta$ are shown in Figure 5.8. In the first plot of Figure 5.8, we show how $\varepsilon$ decreases to a small value. The second plot in the figure shows how the adaptation gain varies based on the variations of $\varepsilon$. It is clear from the figure that the adaptation gain, in almost the first five seconds, increases to its upper bound whenever necessary to keep $\varepsilon$ small, and after that the adaptation gain decreases to its lower bound since $\varepsilon$ is small enough that no major changes in the approximators are needed and hence small adaptation gain is sufficient. As in the direct case, $\varepsilon \in \mathcal{S}(\frac{d^2}{m_0^2})$, $\varepsilon$ can be made smaller.

![Figure 5.8: The responses of $\varepsilon$ and $\eta$ in the indirect case.](image)
by either decreasing $d^2$ or by increasing $m^2$. The changes in the parameter vector of the TSFS are shown in Figure 5.9. It is clear the parameters converge to some value after some time. The response of the aircraft roll angle, $\phi$, is shown in the first plot of Figure 5.10. The second plot of this figure shows the behavior of the aileron input, $\delta_A$. It is clear that the response of the aircraft roll angle is unacceptable with this

![Figure 5.9: The parameter vector, $A$, in the indirect case.](image)

set of controller parameters. As in the direct case, however, such results are expected since the objective of the adaptive control law is to drive $\varepsilon$ (not the tracking error, $e_0$) to a small value. The main difference is that $\delta$ cannot be made much larger than what we have here (we can only increase it to about 20), and hence $e_s$ (and $e_0$) cannot
Figure 5.10: The responses of the aircraft roll angle and aileron input in the indirect case for $\delta = 3$

be driven to smaller values. The response of the aircraft roll angle and aileron input in the for $\delta = 20$ is shown in Figure 5.11.

As in the direct case, we need to investigate how the MSE and MCE change for different values over the feasible range of adaptation gain (which is in this case $0 < \eta < 100$). Figure 5.12 shows how both the MCE and MSE change for several fixed values of the adaptation gain over a simulation period of 10 seconds. The first plot in Figure 5.12 shows the changes in MCE for several values of the adaptation gain. The dotted line in this figure shows the value of of the MCE when the auto-tuning algorithm is used. This value is found to be 0.2074. It is clear that MCE (except at
very small values of $\eta$) slightly oscillate around 0.9. The reason behind large MCE values at small fixed values of $\eta$ is stated earlier in the direct case. It is clear that the MCE achieved when the auto-tuning algorithm is used is larger than most MCE values obtained using fixed adaptation gains. Similarly, the second plot in the figure shows the changes in MSE for several fixed values of the adaptation gain, and the dotted line shows the value of the MSE when the auto-tuning algorithm is used. This value is found to be 1.6528. The third plot is only a scaled version of the second plot to clarify the variation of MSE at large values of the adaptation gain. It is clear from the figure that the MSE decreases as the adaptation gain increases. This decrease
is due to the relatively large control energy (compared to the MCE obtained using the auto-tuning algorithm) that improves the closed-loop performance. The MSE obtained using the auto-tuning algorithm is found to be larger than almost any MSE value obtained at a fixed adaptation gain. As in the direct case, this is due to the fact that in the auto-tuning algorithm the adaptation gain is obtained to minimize the control energy at the expense of error energy. Hence, we can also conclude that our simulation results support the objective of the presented algorithm in the sense
that the adaptation gain is selected on-line to minimize the control energy in such a way that a good closed-loop performance is achieved.

5.6 Concluding Remarks

Considering both direct and indirect adaptive control schemes, the main contribution of this chapter is to auto-tune the adaptation gain for a gradient-based approximator parameter update law used for a class of nonlinear continuous-time systems. The adaptation mechanism of the gradient update law is usually based on minimizing the squared output error. Here, however, we update some parameters in the update law to minimize some other cost function (e.g., control energy). Based on the results of the wing rock example presented earlier, a comparison (to some extent) can be made between direct and indirect adaptive control schemes. Unlike the direct case, it is shown by the example that in the indirect case it is not feasible to decrease $e_s$ (and hence the tracking error, $e_0$) to small values. By example, this represents a major difference between the two approaches.
CHAPTER 6

CONCLUSIONS

In this dissertation we have focused on the problem of auto-tuning some of the controller parameters for nonlinear adaptive control systems. We have considered both discrete and continuous-time systems. In the discrete case, we have auto-tuned the adaptation gain and the direction of descent. In the continuous case, we auto-tuned the adaptation gain. The common feature of all these problems considered is the on-line optimization of some cost function so that some of the control parameters are updated. Next, we summarize the results obtained, and point out some of their weaknesses and strengths.

6.1 Summary

For a class of discrete-time systems, we first presented the system we consider for control along with its assumptions. Then, we discussed the gradient update law that is used for parameter adaptation. We also briefly discussed direct and indirect adaptive control schemes, and derived their output error equations. Then, we have established (based on the results obtained in [43]) the stability results for both direct and indirect adaptive control schemes. These results are considered local with respect to the parameters, but not with respect to the state. These results are the basic
foundation of the auto-tuning results that have been established for the discrete-time throughout the rest of the dissertation.

We have presented two algorithms to auto-tune the adaptation gain for discrete-time nonlinear systems. In the first algorithm, the adaptation gain was selected to minimize the the instantaneous control energy which is very desirable in many applications. In the second algorithm, however, the adaptation gain selected is the one that produces a control that is found by optimizing some criterion. In both cases, the adaptive law was updated not only to minimize the squared output error (as we usually do when we use the gradient update law), but also minimize some other cost function of interest. We have also applied the presented algorithms (for both direct and indirect cases) to a surge tank example. Based on the results of this example, it was difficult to favor one of the two algorithms over the other. However, a comparison (to some extent) was made between direct and indirect cases. Unlike the direct case, it was shown that the selection of the bounds of the adaptation gain in the indirect case is independent of the plant dynamics. This represents a major advantage of the direct case since plant information is incorporated to find the optimal adaptation gain; whereas no such information is used in the indirect case.

We also presented an algorithm to auto-tune the direction of descent for a gradient-based approximator parameter update law used for the same class of nonlinear discrete-time systems considered for auto-tuning the adaptation gain. In this algorithm, the direction of descent is obtained by minimizing the instantaneous control energy. We have shown that updating the adaptation gain can be viewed as a special case of updating the direction of descent. We also illustrated the performance of the presented algorithm via a simple surge tank example. Based on simulation, we have
shown (since adapting the direction of descent is based on minimizing the instantaneous control energy and the gradient update law is based on minimizing the squared output error) that adapting the direction of descent can be used to trade-off between the desired closed-loop performance relative to the affordable control energy.

We have also discussed direct and indirect adaptive control methodologies for a class of continuous-time nonlinear systems. For these control schemes, we have used a gradient-based hybrid update law for parameter adaptation. Then, we presented an algorithm to auto-tune the adaptation gain to minimize the instantaneous control energy. For this case, we have established local results and guaranteed boundedness of all input and output variables. Then, we implemented the auto-tuning algorithm on a wing rock example. Based on the results of the example, some comparisons were made between direct and indirect adaptive control schemes. Unlike the direct case, it was shown by the example that in the indirect case it is not feasible to decrease $e_\nu$ (and hence the tracking error, $e_0$) to small values. In simulation, this represents a major difference between the two approaches.

6.2 Contributions

The main contribution of this dissertation lies within the methodologies presented to auto-tune adaptive controllers for nonlinear systems. The work presented in this document can be considered as an initial step to improve the performance of adaptive controllers via auto-tuning some of the controller parameters such that some cost functions of interest are optimized. Successful completion of this step and development of other methodologies to auto-tune other controller parameters can lead to the development of an "optimal adaptive control" algorithm. Note that optimal is used
here only since some cost functions are optimized in the auto-tuning process of the controller parameters.

Some work has already been done for the purpose of developing some adaptive optimal control methodologies. In the development of such methodologies, some type of quadratic cost function is typically used in the design of LQ-based control. Two major LQ-based regulation/tracking control schemes are linear quadratic Gaussian control and model predictive control. These approaches have been studied extensively for deterministic and stochastic linear processes [51, 42]. However, unlike here no satisfying advancements have been made in the nonlinear case since no optimality guarantees are provided when the quadratic cost function is solved subject to a nonlinear system.

One of the most important LQ-based optimal control problems is the linear quadratic regulator. This problem is very well studied and an analytical solution is derived as shown in [51, 42] and some references within. For discrete-time linear systems, the adaptive LQG control problem has been studied extensively, especially for ARMAX models [54, 55, 56, 57, 58, 59, 60, 61, 62, 18]. The adaptive LQG control problem has not been studied extensively for continuous time systems [63, 64, 65]. Another quadratic-based control methodology is MPC for which some stability results for linear systems were derived under certain assumptions in and for nonlinear systems in [72, 73, 74, 75, 76, 77].

The underlying principle of all these techniques when they are used as (the so called) “optimal adaptive control” design methodologies is adapting the parameters of some linear model that is used to solve some LQ-based control problem. Note that in all techniques listed above, the authors rely on the feasibility of obtaining some optimal solution when the LQ-based problem is solved subject to some type
of linear system. Unfortunately, most systems encountered in practice are nonlinear. Hence, the techniques listed above fail to provide optimality guarantees when the LQ-based problem is solved subject to nonlinear systems. Hence, all of the above techniques should be classified as "adaptive optimal control" (not "optimal adaptive control") techniques in the sense that the parameters of the linear model used in the design of the LQ-based control problem are updated using some adaptive law. For the linear case, stability and optimality can be guaranteed. However, working with nonlinear systems the problem becomes much more challenging. Considering a class of discrete and continuous-time nonlinear systems, in this dissertation we take the first step toward our objective of attempting to optimize the controller performance via auto-tuning the adaptation gain and direction of descent in the gradient-based adaptive law.

Among the little work done for nonlinear adaptive control of discrete-time systems, the authors in [43] designed an indirect adaptive controller for a class of discrete-time nonlinear systems using nonlinear in the parameter neural networks to approximate the plant dynamics. They established asymptotic convergence of the tracking error to a neighborhood of zero by assuming that the parameters of the neural network are close to the actual ones. In the design methodology presented in [43] it is assumed that the parameters of the adaptive law are chosen a priori such that stability is guaranteed. However, closed-loop performance may deteriorate for certain fixed values of the parameters used in the adaptive law. Here, we approach this problem by auto-tuning some of the parameters used in the adaptive law such that some cost function of interest is optimized and stability is guaranteed. Examples of such parameters include the adaptation gain and the direction of descent.
To auto-tune the adaptation gain for discrete-time systems, we started by establishing (based on the results obtained for the indirect case in [43]) stability results for both direct and indirect adaptive control schemes. These results are considered local with respect to the parameters, but not with respect to the state. Note that our objective in establishing these results was not to obtain some global results; we only focused on setting up the basic foundation that can be used to establish the autotuning results for the discrete-time case. Then, for both direct and indirect adaptive schemes we presented two algorithms to auto-tune the adaptation gain. In the first one, the adaptation gain is selected to minimize the instantaneous control energy. In the second algorithm, the adaptation gain selected is the one that produces a control that is found by optimizing some criterion.

Comparisons can be made between direct and indirect cases. Unlike the direct case, it is shown that the selection of the bounds of the adaptation gain in the indirect case is independent of the plant dynamics. This represents a major advantage of the direct case since plant information is incorporated to find the optimal adaptation gain; whereas no such information is used in the indirect case.

In simulation, it is shown that the closed-loop performance of the two algorithms is so similar that is hard to judge that one is better that the other. We have also found that the lower bound of the adaptation gain \( \eta_{\text{min}}(k) \) in the two algorithms is used as the optimal one when the output error is very small. This result agrees with our intuition since no major changes are needed in the adaptive controller when the achieved closed-loop performance is acceptable. In this case, no major changes in the parameter vector (and hence a small adaptation gain) are needed. In the case where the output error is fairly large, considerable changes in the adaptive controller are
needed. This translates to relatively large changes in the parameter vector (which, of course, require a large adaptation gain). Also, we have found from simulation that the auto-tuning algorithm is able to achieve an acceptable MSE with a relatively small control energy. Hence, we can conclude that our simulation results support the objective of the presented algorithm in the sense that the adaptation gain is selected online to minimize the control energy in such a way that a good closed-loop performance is achieved.

We have also presented an algorithm to tune the direction of descent for a gradient-based approximator parameter update law used for a class of nonlinear discrete-time systems (considered in [43]) in both direct and indirect cases. In this algorithm, the direction of descent is obtained by minimizing the instantaneous control energy. We will show that updating the adaptation gain can be viewed as a special case of updating the direction of descent. From simulation, we have shown that the mean control energy decreases as \( \sigma \) increases. Also, the mean squared output error increases as \( \sigma \) increases. Hence, adapting the direction of descent can be used to trade-off between the desired closed-loop performance relative to the affordable control energy.

Among the work done for continuous-time systems, the authors in [32] presented indirect and direct adaptive control algorithms using linearly parameterized fuzzy systems or neural networks to control SISO affine systems with guaranteed convergence of the tracking error to zero in both cases relying on a sliding mode term to compensate for approximation error. As in the discrete-time case, in the design methodology presented in [4], it is assumed that the parameters of the adaptive law are chosen \textit{a priori} such that stability is guaranteed. However, closed-loop performance may deteriorate for certain fixed values of the parameters used in the adaptive law. Here,
we approach this problem by auto-tuning the adaptation gain used in the adaptive law such that some cost function of interest is optimized and stability is guaranteed.

Considering the class of continuous-time systems used in [32], we have established some local results using a gradient-based hybrid adaptive law used in [4]. Note that our objective in establishing this stability result was not to obtain some global result; we only focused on setting up the basic foundation that can be used to establish the auto-tuning results for the continuous-time case. Then, for both direct and indirect adaptive schemes we presented algorithms to auto-tune the adaptation gain to minimize the instantaneous control energy. Based on simulation, we can conclude that our simulation results support the objective of the presented algorithm in the sense that the adaptation gain is selected on-line to minimize the control energy in such a way that a good closed-loop performance is achieved. Also, we have found from simulation that the main difference between direct and indirect adaptive control schemes is that in the indirect case \( \delta \) cannot be made very large, and hence \( e_s \) (and \( e_0 \)) cannot be driven to small values.

6.3 Future Directions

There are several problems that remain open in the research proposed in this dissertation. In the problem of adaptive control for nonlinear systems, the following problem needs to be investigated: Is it possible to obtain global results for both discrete and continuous-time systems (without relying on high gain terms)? The answer to this problem is not clear, and it needs some mathematical investigation so that the approach of on-line approximation-based adaptive control may become more applicable.
Another problem of interest is extending the work presented in this dissertation to handle MIMO systems for discrete and continuous-time systems. We hope that (in the case of successful completion of this work for both direct and indirect cases) it will be used for a more general class of systems, but expect that typical "diagonal dominance" conditions will be needed. We also hope that it will provide some insight about the relationship between direct and indirect adaptive control schemes. Also, for the problem of auto-tuning the adaptation gain, we will attempt to obtain some results (similar to the ones obtained in this dissertation for a gradient-based update law) when the Recursive Least Squares (RLS) algorithm is used. This problem will also provide a basis for comparison not only between the direct and indirect schemes as online approximation-based control methodologies, but also between gradient and RLS as parameters update routines.

One other problem of interest is investigating the applicability of extending the work presented in this dissertation for continuous-time systems so that the output error (not \( \epsilon \) which is a function of the output error) is driven to some small neighborhood of zero. This problem (as we have seen in the indirect case) has deteriorated the closed-loop tracking performance.

Also, another problem of great importance is extending the work presented in this dissertation to handle time-varying and output feedback systems. This would clearly broaden the applicability of the work presented in this dissertation. Another problem is to explore the use of other cost functions. Also, another challenging problem is to investigate whether parameters that enter in a nonlinear fashion could be tuned to minimize instantaneous control energy.
BIBLIOGRAPHY


