© Copyright by
José J. Velásquez G.
2013
Abstract

This research is motivated by the world’s fast-growing demand for energy savings. Due to the increasing cost of fossil fuels (e.g., oil, coal, natural gas), research has been conducted to effectively reduce the electricity consumption in office buildings by means of employing smart lighting. This thesis investigates the implementation of an adaptive and nonadaptive fuzzy control for a smart light experimental testbed. The objective is to accurately regulate the light level across the experimental testbed to a desired voltage reference value, and to test the performance of the fuzzy controllers under cross-illumination effects, and bulb and sensor failures. As an initial approach, a decentralized (i.e., no communication between controllers) nonadaptive fuzzy controller is implemented and applied to the experimental testbed. This approach is convenient for this type of experimental testbed where a mathematical model of the plant is not available and heuristic information about how to control the system is sufficient. The nonadaptive fuzzy controller, when properly tuned, is able to achieve uniform lighting across the entire testbed floor in most of the tested situations but it fails whenever an on/off light bulb failure is introduced. In order to attain uniform lighting for complex failures, a fuzzy model reference learning controller (i.e., adaptive fuzzy) is developed for the experimental testbed, and this algorithm proves to be able to adapt to uncertainties such as disturbances and failures via a learning mechanism.
To my parents, José and Ana, and to my sister and brother.
Acknowledgments

First of all, I give thanks to my parents, José and Ana, for their unstoppable support throughout all my life. I could not have achieved many of my lifetime goals without them. To my sister, Keyla, and my brother, Pablo, for their love and trust in me. To my girlfriend Stephanie, who has been very supportive even though we are not physically together.

I want to thank my advisor, Prof. Kevin M. Passino, for giving me the opportunity to work in the smart light project under his supervision. His guidance and support has been a key factor for the completion of this research. I also want to thank Prof. Wei Zhang for his participation on my master’s examination committee. He is certainly one of the best Professors that I have seen through my academic life. I am very glad that I had the opportunity to take one of his courses and benefit from him.

Also, I am thankful to the Department of Electrical and Computer Engineering for giving me the opportunity to develop as a Graduate Teaching Associate in the Control System Laboratory.

I am also thankful for my american dad, Todd, for his constant support and encouragement. He helped me out on my decision to study in the United States and has made feel his home like my home.

Finally, I would like to thank God who has been constantly providing me the strength and courage to face every day of my life.
Vita

October 5, 1987 .......................... Born - Porlamar, Venezuela

July 2008 - August 2008 ............... Summer Internship,
Rockwell Automation of Venezuela,
Caracas, Venezuela

July 2009 - September 2009 .......... Hired Engineer,
Rockwell Automation of Venezuela,
Caracas, Venezuela

September 2009 - June 2010 ............ E.E. International Exchange Student,
Faculty of Computing and Engineering,
University of Ulster,
Jordanstown, Northern Ireland

February 2011 .......................... B.S. Electronic Engineering
Cum Laude
Department of Electronics & Circuits,
Universidad Simón Bolívar,
Caracas, Venezuela

August 2012 - present .................... Graduate Teaching Associate,
Department of Electrical & Computer
Engineering,
The Ohio State University,
Columbus, OH

Fields of Study

Major Field: Electrical & Computer 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>iii</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>iv</td>
</tr>
<tr>
<td>Vita</td>
<td>v</td>
</tr>
<tr>
<td>List of Tables</td>
<td>viii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>ix</td>
</tr>
<tr>
<td>1. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Overview of Smart Lighting Systems</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Contributions of this Thesis</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Thesis Outline</td>
<td>4</td>
</tr>
<tr>
<td>2. Experimental Testbed for Smart Lighting</td>
<td>5</td>
</tr>
<tr>
<td>2.1 Experimental Testbed Layout</td>
<td>5</td>
</tr>
<tr>
<td>2.2 Driving and Acquisition Circuitry</td>
<td>9</td>
</tr>
<tr>
<td>2.3 Overall Smart Lighting Control Loop</td>
<td>13</td>
</tr>
<tr>
<td>2.4 Photocell Sensor Calibration</td>
<td>15</td>
</tr>
<tr>
<td>2.4.1 Smart Light Calibration Algorithm</td>
<td>17</td>
</tr>
<tr>
<td>3. Fuzzy Control for Smart Lights</td>
<td>20</td>
</tr>
<tr>
<td>3.1 Fuzzy System Overview</td>
<td>20</td>
</tr>
<tr>
<td>3.1.1 Selecting Fuzzy Controller Inputs and Outputs</td>
<td>21</td>
</tr>
<tr>
<td>3.1.2 Converting Control Knowledge into Rule Bases</td>
<td>24</td>
</tr>
</tbody>
</table>
3.1.3 Fuzzy Numerical Interpretation of Knowledge ............... 31
3.1.4 Deciding Which Rules to Use .............................. 39
3.1.5 Inference Step: Deciding Conclusions ....................... 44
3.1.6 Transforming Decisions into Actions ........................ 49

3.2 Design Case: Direct PI Fuzzy Control for Smart Lights ........ 51
3.2.1 Implementation of a Direct PI Fuzzy Controller ............ 52
3.2.2 Direct PI Fuzzy Controller Mapping Surface ................ 54

3.3 Adaptive Fuzzy Control: Emulating Expertise Learning ........ 56
3.3.1 Fuzzy Model Reference Learning Control .................... 57

3.4 Adaptive PI Fuzzy Control for Smart Lights ................... 64
3.4.1 Implementation of the FMRLC for Smart Lights ............. 64

4. Simple Fault Tolerance Test for Smart Lights .................... 68
4.1 Single Zone Light Bulb Failure .................................. 68
4.1.1 Direct PI Fuzzy Controller Results ......................... 69
4.1.2 Direct Adaptive PI Fuzzy Controller Results ............... 71
4.2 Multiple Zones Light Bulb Failure ............................... 73
4.2.1 Direct PI Fuzzy Controller Results ......................... 73
4.2.2 Direct Adaptive PI Fuzzy Controller Results ............... 83

5. On/Off Fault Tolerance Test for Smart Lights .................... 94
5.1 Single Zone On/Off Light Bulb Failure ........................... 94
5.1.1 Direct PI Fuzzy Controller Results ......................... 95
5.1.2 Direct Adaptive PI Fuzzy Controller Results ............... 97
5.2 Multiple Zones On/Off Light Bulb Failure ....................... 98
5.2.1 Direct PI Fuzzy Controller Results ......................... 99
5.2.2 Direct Adaptive PI Fuzzy Controller Results ............... 105

6. Sensor Failure Test for Smart Lights ............................. 115
6.1 Single Zone Sensor Failure ..................................... 115
6.1.1 Direct PI Fuzzy Controller Results ......................... 116
6.1.2 Direct Adaptive PI Fuzzy Controller Results ............... 118
6.2 Two-Zone Sensor Failure ........................................ 120
6.2.1 PI Fuzzy Controller Results ................................. 120
6.2.2 Direct Adaptive PI Fuzzy Controller Results ............... 127

7. Conclusions and Future Work ..................................... 134

Bibliography ............................................................ 136
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>30</td>
</tr>
<tr>
<td>3.2</td>
<td>37</td>
</tr>
<tr>
<td>3.3</td>
<td>45</td>
</tr>
<tr>
<td>3.4</td>
<td>66</td>
</tr>
</tbody>
</table>

3.1 Rule table for smart lighting system.

3.2 Rule table for smart lighting system (the content represents the output membership function centers where each element has to be scaled by 100).

3.3 Rule table for the smart lighting system with rules that are “on” (square boxed). (The content of the table holds the output membership function centers were each element should be multiplied by 100).

3.4 Rule table for smart lights fuzzy inverse model.
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Floorplan of the experimental testbed for smart lights.</td>
<td>6</td>
</tr>
<tr>
<td>2.2</td>
<td>Cross section view “AA” of zone 2 illustrating light bulb and sensor.</td>
<td>7</td>
</tr>
<tr>
<td>2.3</td>
<td>(a) Relative spectral response of the CdS photocell. (b) Typical resistance vs. illumination characteristics of the CdS photocell (taken from [1]).</td>
<td>8</td>
</tr>
<tr>
<td>2.4</td>
<td>Driving circuitry for each zone of the experimental testbed.</td>
<td>10</td>
</tr>
<tr>
<td>2.5</td>
<td>(a) TIP120 package layout (b) TIP120 equivalent circuit (taken from [2])</td>
<td>11</td>
</tr>
<tr>
<td>2.6</td>
<td>Acquisition circuitry for each zone of the experimental testbed.</td>
<td>12</td>
</tr>
<tr>
<td>2.7</td>
<td>Overall control loop diagram.</td>
<td>13</td>
</tr>
<tr>
<td>2.8</td>
<td>Simulink model for the implementation of eight independent controllers.</td>
<td>15</td>
</tr>
<tr>
<td>2.9</td>
<td>Raw sensor data after applying a voltage sweep from 0 V to 10 V.</td>
<td>16</td>
</tr>
<tr>
<td>2.10</td>
<td>Calibrated sensor data after applying a voltage sweep from 0 V to 10 V.</td>
<td>18</td>
</tr>
<tr>
<td>2.11</td>
<td>Comparison of calibrated sensor data.</td>
<td>19</td>
</tr>
<tr>
<td>3.1</td>
<td>Fuzzy controller for smart lights (adapted from [3]).</td>
<td>21</td>
</tr>
<tr>
<td>3.2</td>
<td>Human controlling a smart light (adapted from [4]).</td>
<td>22</td>
</tr>
<tr>
<td>3.3</td>
<td>Fuzzy controller for smart light illumination problem (adapted from [4]).</td>
<td>23</td>
</tr>
</tbody>
</table>
3.4 Membership function for linguistic value “possmall” (adapted from [4]).

3.5 Membership functions for smart lighting system (adapted from [4]).

3.6 Membership functions of premise terms (adapted from [4]).

3.7 Input membership functions with input values (adapted from [4]).

3.8 (a) Consequent membership function and (b) implied fuzzy set with membership function \( \mu(1)(V_{app}) \) for rule (1) (adapted from [4]).

3.9 (a) Consequent membership function and (b) implied fuzzy set with membership function \( \mu(2)(V_{app}(t)) \) for rule (2) (adapted from [4]).

3.10 Implied fuzzy sets from rule (1) and (2) (adapted from [4]).

3.11 PI fuzzy controller for smart lights with scaling gains.

3.12 Normalized universes of discourse for fuzzy controller for smart light system (scaling gains are given in boxed values).

3.13 Nonlinear control surface by direct PI fuzzy controller for smart lights.

3.14 Direct adaptive control block diagram (adapted from [3]).

3.15 Fuzzy model reference learning controller (adapted from [5]).

3.16 Membership functions for input universe of discourse (adapted from [5]).

4.1 Light levels at each zone for the direct PI fuzzy controller with no partition between zones under zone 1 light bulb failure.

4.2 Applied voltage level to the light bulbs at each zone for the direct PI fuzzy controller with no partition between zones under zone 1 light bulb failure.

4.3 Light levels at each zone for the direct adaptive PI fuzzy controller with no partition between zones under zone 1 light bulb failure.
4.4 Applied voltage level to the light bulbs at each zone for the direct adaptive PI fuzzy controller with no partition between zones under zone 1 light bulb failure. ........................................ 72

4.5 Light levels at each zone for the direct PI fuzzy controller with no partition between zones under zones 1, 3, and 4 light bulb failure. ........................................ 74

4.6 Applied voltage level to the light bulbs at each zone for the direct PI fuzzy controller with no partition between zones under zones 1, 3, and 4 light bulb failure. ........................................ 75

4.7 Light level in controlled zones 2, 5, 6, 7, and 8 for the direct PI fuzzy controller with no partition between zones under zones 1, 3, and 4 light bulb failure. ........................................ 76

4.8 Light levels at each zone for the direct PI fuzzy controller with no partition between zones under zones 1, 4, 5, and 8 light bulb failure. ........................................ 77

4.9 Applied voltage level to the light bulbs at each zone for the direct PI fuzzy controller with no partition between zones under zones 1, 4, 5, and 8 light bulb failure. ........................................ 78

4.10 Light level in controlled zones 2, 3, 6, and 7 for the direct PI fuzzy controller with no partition between zones under zones 1, 4, 5, and 8 light bulb failure. ........................................ 79

4.11 Light levels at each zone for the direct PI fuzzy controller with no partition between zones under zones 3, 4, 5, and 6 light bulb failure. ........................................ 80

4.12 Applied voltage level to the light bulbs at each zone for the direct PI fuzzy controller with no partition between zones under zones 3, 4, 5, and 6 light bulb failure. ........................................ 81

4.13 Light level in controlled zones 1, 2, 7, and 8 for the direct PI fuzzy controller with no partition between zones under zones 3, 4, 5, and 6 light bulb failure. ........................................ 82

4.14 Light levels at each zone for the direct adaptive PI fuzzy controller with no partition between zones under zones 1, 3, and 4 light bulb failure. ........................................ 83
4.15 Applied voltage level to the light bulbs at each zone for the direct adaptive PI fuzzy controller with no partition between zones under zones 1, 3, and 4 light bulb failure. ........................................... 84

4.16 Light level in controlled zones 2, 5, 6, 7, and 8 for the direct adaptive PI fuzzy controller with no partition between zones under zones 1, 3, and 4 light bulb failure. ........................................... 85

4.17 Light levels in each zone for the direct adaptive PI fuzzy controller with no partition between zones under zones 1, 4, 5, and 8 light bulb failure. ........................................... 86

4.18 Applied voltage level to the light bulbs at each zone for the direct adaptive PI fuzzy controller with no partition between zones under zones 1, 4, 5, and 8 light bulb failure. ........................................... 87

4.19 Light level in controlled zones 2, 3, 6, and 7 for the direct adaptive PI fuzzy controller with no partition between zones under zones 1, 4, 5, and 8 light bulb failure. ........................................... 88

4.20 Light levels at each zone for the direct adaptive PI fuzzy controller with no partition between zones under zones 3, 4, 5, and 6 light bulb failure. ........................................... 89

4.21 Applied voltage level to the light bulbs at each zone for the direct adaptive PI fuzzy controller with no partition between zones under zones 3, 4, 5, and 6 light bulb failure. ........................................... 90

4.22 Light level in controlled zones 1, 2, 7, and 8 for the direct adaptive PI fuzzy controller with no partition between zones under zones 3, 4, 5, and 6 light bulb failure. ........................................... 91

4.23 Nonlinear control surface implemented by the direct PI fuzzy controller in zone 1 at $t = 60$ seconds under zones 3, 4, 5, and 6 light bulb failure. 92

5.1 Light levels at each zone for the direct PI fuzzy controller with no partition between zones under zone 1 on/off light bulb failure. .......  95

5.2 Applied voltage level to the light bulbs at each zone for the direct PI fuzzy controller with no partition between zones under zone 1 on/off light bulb failure. ........................................... 96
5.3 Light levels at each zone for the direct adaptive PI fuzzy controller with no partition between zones under zone 1 on/off light bulb failure. 97

5.4 Applied voltage level to the light bulbs at each zone for the direct adaptive PI fuzzy controller with no partition between zones under zone 1 on/off light bulb failure. 98

5.5 Light levels at each zone for the direct PI fuzzy controller with no partition between zones under zones 1, 3, and 4 on/off light bulb failure. 100

5.6 Applied voltage level to the light bulbs at each zone for the direct PI fuzzy controller with no partition between zones under zones 1, 3, and 4 on/off light bulb failure. 101

5.7 Light levels at each zone for the direct PI fuzzy controller with no partition between zones under zones 1, 4, 5 and 8 on/off light bulb failure. 102

5.8 Applied voltage level to the light bulbs at each zone for the direct PI fuzzy controller with no partition between zones under zones 1, 4, 5 and 8 on/off light bulb failure. 103

5.9 Light levels at each zone for the direct PI fuzzy controller with no partition between zones under zones 3, 4, 5 and 6 on/off light bulb failure. 104

5.10 Applied voltage level to the light bulbs at each zone for the direct PI fuzzy controller with no partition between zones under zones 3, 4, 5 and 6 on/off light bulb failure. 105

5.11 Light levels at each zone for the direct adaptive PI fuzzy controller with no partition between zones under zones 1, 3, and 4 on/off light bulb failure. 106

5.12 Applied voltage level to the light bulbs at each zone for the direct adaptive PI fuzzy controller with no partition between zones under zones 1, 3, and 4 on/off light bulb failure. 107
5.13 Light levels at each zone for the direct adaptive PI fuzzy controller with no partition between zones under zones 1, 4, 5 and 8 on/off light bulb failure. ...................................................... 108

5.14 Applied voltage level to the light bulbs at each zone for the direct adaptive PI fuzzy controller with no partition between zones under zones 1, 4, 5 and 8 on/off light bulb failure. ...................................................... 109

5.15 Light levels at each zone for the direct adaptive PI fuzzy controller with no partition between zones under zones 3, 4, 5 and 6 on/off light bulb failure. ...................................................... 110

5.16 Applied voltage level to the light bulbs at each zone for the direct adaptive PI fuzzy controller with no partition between zones under zones 3, 4, 5 and 6 on/off light bulb failure. ...................................................... 111

5.17 Nonlinear control surface implemented by the direct adaptive PI fuzzy controller in zone 1 at \( t = 35 \) seconds under an on/off light bulb failure in zones 3, 4, 5 and 6. ...................................................... 112

5.18 Nonlinear control surface implemented by the direct adaptive PI fuzzy controller in zone 1 at \( t = 60 \) seconds under an on/off light bulb failure in zones 3, 4, 5 and 6. ...................................................... 113

6.1 Light levels at each zone for the PI fuzzy controller with half partitions between zones under zone 1 sensor failure. ...................................................... 116

6.2 Applied voltage level to the light bulbs at each zone for the PI fuzzy controller with half partition between zones under zone 1 sensor failure.117

6.3 Light levels at each zone for the direct adaptive PI fuzzy controller with half partition between zones under zone 1 sensor failure. ...................... 118

6.4 Applied voltage level to the light bulbs at each zone for the direct adaptive PI fuzzy controller with half partition between zones under zone 1 sensor failure. ...................... 119

6.5 Light levels at each zone for the PI fuzzy controller with half partition between zones under zones 1 and 8 sensor failure. ...................... 121
6.6 Applied voltage level to the light bulbs at each zone for the PI fuzzy controller with half partition between zones under zones 1 and 8 sensor failure. ................................. 122

6.7 Light level in controlled zones 2, 3, 4, 5, 6, and 7 for the PI fuzzy controller with half partition between zones under zones 1 and 8 sensor failure. ................................. 123

6.8 Light levels at each zone for the PI fuzzy controller with half partition between zones under zones 3 and 6 sensor failure. ................................. 124

6.9 Applied voltage level to the light bulbs at each zone for the PI fuzzy controller with half partition between zones under zones 3 and 6 sensor failure. ................................. 125

6.10 Light level in controlled zones 1, 2, 4, 5, 7, and 8 for the direct PI fuzzy controller with half partition between zones under zones 3 and 6 sensor failure. ................................. 126

6.11 Light levels at each zone for the direct adaptive PI fuzzy controller with half partition between zones under zones 1 and 8 sensor failure. ................................. 127

6.12 Applied voltage level to the light bulbs at each zone for the direct adaptive PI fuzzy controller with half partition between zones under zones 1 and 8 sensor failure. ................................. 128

6.13 Light level in controlled zones 2, 3, 4, 5, 6, and 7 for the direct adaptive PI fuzzy controller with half partition between zones under zones 1 and 8 sensor failure. ................................. 129

6.14 Light levels at each zone for the direct adaptive PI fuzzy controller with half partition between zones under zones 3 and 6 sensor failure. ................................. 130

6.15 Applied voltage level to the light bulbs at each zone for the direct adaptive PI fuzzy controller with half partition between zones under zones 3 and 6 sensor failure. ................................. 131

6.16 Light level in controlled zones 1, 2, 4, 5, 7, and 8 for the direct adaptive PI fuzzy controller with half partition between zones under zones 3 and 6 sensor failure. ................................. 132
Chapter 1: Introduction

1.1 Overview of Smart Lighting Systems

Lighting represents one of the key components of today’s residential and commercial buildings. In fact, lighting consumes around 35% of the electricity used in commercial building in the United States. Also, lighting uses 18% of the electricity produced in the U.S., and an additional 4% to 5% has to be used just to remove the waste heat caused by the lights [6]. A good smart lighting system design can be a promising solution to this energy consumption problem. The lighting energy usage in most buildings can be reduced significantly while maintaining the quality of service. This reduction of electricity usage can be achieved by reducing the over-illumination produced by neighboring lights and considering external light sources (e.g., daylight harvesting). A significant saving (up to 40%) can be obtained by applying modern control strategies such as daylight harvesting, occupancy sensing, scheduling and load shedding [7, 8].

Over recent years, in order to improve the lighting usage in commercial buildings and reduce the utility costs, smart lighting systems have been developed and implemented for controlling the illumination level in an office setting [8, 9, 10, 11]. Martirano discusses the importance of using efficiency and effectiveness to approach
a smart lighting control problem [8]. Efficiency is achieved via using low consumption equipment (i.e., LED luminaries) and improved lighting design practices (i.e., localized task lighting systems). Effectiveness is achieved by improving the current lighting control systems to avoid additional unnecessary energy usage and adopting a technical building management system (i.e., maintenance and metering). This research claims that a smart lighting control system can allow a saving of up to 25% in industrial and commercial buildings, and up to 45% in tertiary and educational buildings [8].

Wen et al. developed an energy-efficient lighting system based on a wireless sensor network technology [9, 10]. In this system, they implemented a centralized intelligent lighting optimization algorithm where the overall lighting in an office is determined as a linear combination of the light contributions from each of the luminaries based on a discretization of the office into square grids. This light setting control is coded as a linear programming optimization problem supported by the fact that the power consumption is directly proportional to the light output from the luminaries, and hence, minimizing the illuminances is similar to minimizing the electricity usage. Once the centralized optimization algorithm computes the optimal linear combination of the individual luminaries that minimizes the entire lighting output, then the control signal is sent via the wireless network [9, 10].

An adaptive lighting system has been developed in [11]. Hughes et al. proposed the usage of an intelligent control system that allows the lighting system to dynamically respond to the current conditions of the area it illuminates. For this adaptive system, they have included within their control loop occupancy sensors, photosensors, and personal control modules. The proposed adaptive lighting system is programmed
to keep the desired ambient illumination levels and continuously adjust task illumination levels in individual areas as occupancy varies. This adaptive control system has been able to provide a significant reduction in energy consumption of about 69% compared to a conventional static lighting system [11].

1.2 Contributions of this Thesis

This research focuses on the first application of fuzzy control for intelligent illumination control. The implemented adaptive fuzzy controller has been able to effectively achieve uniform lighting across the entire floor of the experimental testbed under different testbed settings. The no-partition case has been the selected to present most of our results since it represents the most challenging illumination control problem (i.e., maximized cross-illumination between zones). We have proposed a decentralized control approach where each zone of the testbed has an independent fuzzy controller. This fact eliminates the necessity of communication between the zones, and furthermore, not requiring the implementation of a sensor network to acquire the data, compute the control action, and then apply the control signal to the plant.

In this thesis, we adapted the “fuzzy model reference learning controller” (FMRLC) approach [5] to our smart lighting testbed, something that has not been done before. We have been able to justify the advantages of the FMRLC which are mentioned in [5]. These advantages are: improved convergence rates (i.e., faster transient response), use of less control energy (i.e., reduced light consumption), enhanced disturbance rejection properties (i.e., rejection to light bulb and sensor failure), and lack of dependence on a mathematical model (i.e., heuristic knowledge about the plant is sufficient).
1.3 Thesis Outline

This thesis is organized as follows. In Chapter 2, we present a detailed description of the smart light experimental testbed. This chapter is provided to describe the testbed layout, the circuit schematic for the driving and acquisition circuits, the implementation of the overall control loop, and the photocell sensor calibration which are the backbone of the experimental smart light testbed. In Chapter 3, we introduce the application of fuzzy control for smart lighting. This chapter unifies all the basic ideas connected to the development and implementation of a nonadaptive and adaptive fuzzy controller specifically for our smart light testbed. In Chapter 4, we provide the results obtained for the performance of the nonadaptive and adaptive fuzzy controllers for single zone and multiple zones light bulb failure. We illustrate the ability of both the PI fuzzy control and direct adaptive PI fuzzy control to reject plant disturbances. In Chapter 5, we cover the results achieved by the nonadaptive and adaptive fuzzy controllers for a single zone and multiple zones on/off light bulb failure. This type of light bulb failure introduces additional control challenges to the experimental testbed and we illustrate the ability of the adaptive fuzzy controller to adapt to plant variations and disturbances via its built-in learning mechanism. In Chapter 6, we discuss the results given by a sensor failure in single and multiple zone scenarios. This type of sensor failure provides a special case where the cross-illumination is maximized around the failing zones; hence, the tracking of certain voltage reference inputs might not be feasible by either the nonadaptive or the adaptive fuzzy controllers. In Chapter 7, we provide a brief conclusion and mention some suggestions for possible future research directions.
Chapter 2: Experimental Testbed for Smart Lighting

In this chapter we provide a description of the experimental testbed. This experimental testbed is designed to replicate the layout of a small office setting composed of eight different zones. Each zone is equipped with a light bulb and a light dependent resistor (LDR) in order to sense the illumination across the floor of each corresponding zone via a voltage signal. When there are not partitions between the zones the cross-illumination effect is maximized, and thus, a more challenging control problem is created. Additionally, the overall feedback control loop implemented in Simulink/dSPACE is also presented and described.

2.1 Experimental Testbed Layout

The experimental tested is made up from a shoe box of 22.5 × 33 cm. The floorplan of the testbed is presented in Figure 2.1. Notice in Figure 2.1 that the eight zones are not equally distributed across the entire floorplan. Zones 1 and 2 are 11.25 × 10 cm, zones 3 and 4 are 11.25 × 6.5 cm, zones 5 and 6 are 11.25 × 7.5 cm, and zones 7 and 8 are 11.25 × 9 cm. Additionally, we can observe that the location of the light dependent resistor for sensing the illumination across the floor in each zone is not located in the center of the corresponding zone.
This particular testbed layout introduces additional challenges for the design of a controller intended to accurately regulate the light level across the floor of the experimental testbed. The bold lines separating each zone represent cardboard partitions that can be set up between the zones in order to simulate different room settings. There are three main room partition settings: full height, half height, and no partition. Full height partitions provide us the case with eight independent rooms (i.e., eight independent sensors), half height partitions give us the case with some cross-illumination effect between neighboring rooms, and the case of no partitions introduces maximized cross-illumination effects throughout the entire testbed.
In this work, we will focus in the no partition case because it provides the most challenging control problem as described in [12, 13, 14]. However, all the control algorithms discussed in the following chapter were tested under all the room settings to guarantee a stable system performance.

Figure 2.2 presents the cross section view “AA” of zone 2 illustrating the location of both the light bulb and the light sensor. Clearly, the light bulb is located right above the light sensor for better light sensing. The light bulb is a miniature incandescent bulb (base #1847) of 0.25 Watts operating at 6.3 Volts with a length of 3 cm. The light sensor is a Cadmium-Sulfide (CdS) photocell (RadioSchack Part #276-1657) featuring visible light response, sintered construction, and low cost [1].

![Figure 2.2: Cross section view “AA” of zone 2 illustrating light bulb and sensor.](image)
The eight Cadmium-Sulfide photocells installed in the experimental testbed play a key role in our smart light system. For this reason, it is very important to understand the photocell characteristics and its limitations due to its low cost. Figure 2.3 contains the two most important plots of a light dependent resistor. Figure 2.3 (a) has the relative spectral response of the Cadmium-Sulfide photocell and shows that around 550 nm (i.e., visible light wavelength) the sensor has the best relative spectral response (i.e., a relative response of 1.0) which is desired for our application. Figure 2.3 (b) has the typical resistance versus illumination characteristics of the Cadmium-Sulfide photocell and illustrates how at lower illumination (i.e, 0.01 lux) the equivalent resistance is about 2 MΩ, and at higher illumination (i.e., 100 lux) the equivalent resistance is approximately 1 kΩ.

**Figure 2.3:** (a) Relative spectral response of the CdS photocell. (b) Typical resistance vs. illumination characteristics of the CdS photocell (taken from [1]).
2.2 Driving and Acquisition Circuitry

The smart lighting experimental tested is composed of two main circuitries, the driving and the acquisition circuitry. These two main circuitries are interfaced by means of a research and development (R&D) controller board from dSPACE Inc. The dSPACE DS1104 R&D controller board is equipped with real-time interface (RTI) which can be graphically programmed in Simulink from MATLAB. The DS1104 board provides us with eight analog to digital converter (ADC) channels to interface the output of the light sensors with the controller coded in the digital computer and eight digital to analog converter (DAC) channels to interface the controller output to the light bulbs as an analog signal. Clearly, the analog signal generated by the DAC channels cannot be used to directly feed the light bulbs due to its low current output therefore an adequate driving circuitry has to be implemented.

First of all, we will look over the characteristics of the driving circuitry. In Figure 2.4, the schematic layout is given for the driving circuits in each zone of the testbed. The driving circuitry is intended to provide the necessary current for the light bulbs in each corresponding zone by using a power transistor as a common-collector amplifier (i.e., a voltage buffer). As illustrated in Figure 2.4, each zone of the experimental testbed has its independent driving circuit. Zones 1, 2, 4 and 7 use the TIP120 NPN epitaxial Darlington transistor as their voltage buffer respectively. Zones 3, 5, 6 and 8 use the TIP31A NPN epitaxial silicon transistor as their voltage buffer respectively. As already mentioned, this testbed was also used in [12, 13, 14] but there is not a clear reason why two different kind of transistors were selected. In any case, we assumed from the beginning of this research that the testbed represents a black box and that we are not allowed to change its internal parameters.
For zones 1, 2, 4 and 7 driving circuit, the TIP120 represents the amplifier transistor. Figure 2.5 contains the package layout and the internal equivalent circuit of the TIP120 NPN epitaxial Darlington transistor. Figure 2.5 (a) shows that the TIP120 has a TO-220 package layout. Figure 2.5 (b) presents the internal equivalent circuit of the TIP120 that is basically a Darlington configuration. The TIP120 is characterized by a maximum DC collector current ($I_c$) of 5 A and a typical Base-Emitter On voltage ($V_{BE(on)}$) of 2.5 V. The common-collector amplifier has an input-output curve characterized by a dead zone for input voltages below the Base-Emitter On voltage.
For the driving circuit of zones 3, 5, 6, and 8, the TIP31A represents the amplifier transistor. The TIP31A has a TO-220 package layout but is internally equivalent to a single bipolar transistor. The TIP31A is characterized by a maximum DC collector current ($I_c$) of 3 A and a typical Base-Emitter On voltage ($V_{BE(on)}$) of 1.8 V. Again, the dead zone created by the common-collector amplifier is going to be limited by input voltages below approximately 1.8 V (i.e., Base-Emitter On voltage). The dead zone is clearly illustrated when we describe the photocell sensor calibration in Section 2.4. In addition, this characteristic brings additional control challenges to our testbed because we have eight zones with eight different input-output characteristic curves which depend on the Base-Emitter On voltage and the transistor DC current gain ($h_{FE}$) that defines the slope of the input-output line (i.e., $\approx 1$ for a voltage buffer).
Secondly, we will discuss the acquisition circuitry. Figure 2.6 shows the schematic layout for the acquisition circuits in each zone of the testbed. The acquisition circuitry is designed to provide a voltage signal (i.e., an analog signal) to each one of the analog to digital converter channels. As shown in Figure 2.6, each zone of the testbed has its independent acquisition circuit. From section 2.1, we stated that the LDR changes its resistance as a function of the illumination. Thus, our acquisition circuitry is formed by a voltage divider which basically will output as much as the source voltage (i.e., $V_{cc} = 13.4 \, \text{V}$) as the illumination on the LDR increases or a smaller voltage (i.e., tending to zero) as the illumination on the LDR decreases.

Figure 2.6: Acquisition circuitry for each zone of the experimental testbed.
2.3 Overall Smart Lighting Control Loop

For our given smart lighting system, we are concerned about regulating the actual light levels across the floor of each zone at a desired reference value. In order to achieve this goal, a smart lighting control loop has to be implemented and coded via MATLAB/Simulink which provides the necessary interfacing to work with the dSPACE DS1104 R&D controller board installed in our research lab. The overall control loop diagram implemented for our smart lighting system is illustrated in Figure 2.7. Clearly, the control loop is formed by four main parts: ADC acquisition, sensor calibration, control algorithm, and DAC output. Each part of the control loop plays a key role in our system and both the ADC acquisition and DAC output let us create the interface between the digital and analog world. Both the sensor calibration and the control algorithms are coded into the digital computer.

![Overall control loop diagram](image)

Figure 2.7: Overall control loop diagram.
In general, the control loop works as follows. The ADC acquisition stage uses the eight analog to digital converter channels of the DS1104 controller board to measure in real time the analog output of the voltage dividers from the acquisition circuits. This acquired raw data is processed by the sensor calibration stage that adjust the measured data in order to work within a linear region. The sensor calibration will be described in more detail in the following section.

Next, the calibrated data is feed into the control algorithm stage that computes the error compared to the given desired reference level and the integral of the error to then produce a control output. Figure 2.8 shows the eight independent controllers implemented in MATLAB/Simulink for each zone respectively. From Figure 2.8, we can observe that the controller has four inputs which includes the voltage reference, output of the plant, error, and the integral of the error and one output given by the control signal to the plant. This entire process is performed in real time with a sampling period of 1 ms that is fast enough to maintain the testbed illumination at the desired value.

Last but not least, the DAC output stage takes the digital control signal computed by the control algorithm state and converts that given digital signal into an analog control signal via a digital to analog conversion. Before applying the analog control signal to the plant, this analog signal is limited between 0 V to 10 V by an output saturator in order to protect the DAC channels of the DS1104 controller board and all the light bulbs. Each of the DAC channels is wired to its corresponding transistor base as described by the driving circuitry in Figure 2.4.
2.4 Photocell Sensor Calibration

Due to the given design of the acquisition circuitry and the intrinsic characteristic of the light dependent resistors, there would be an evident difference between the input-output voltage response of the sensors for each zone. This response difference is directly connected to the fact that two different type of power transistors (i.e., TIP120 and TIP31A) are being used as voltage buffers. Figure 2.9 presents the open loop raw sensor data after applying a voltage sweep from 0 V to 10 V to the base of the power transistors in each corresponding zone with half height partition setting.
Firstly, the typical Base-Emitter On voltage of the power transistors will not be exactly the same even though if the transistors are from the same family. This is illustrated in Figure 2.9 by the length of the dead zone in each corresponding zone which slightly varies (i.e., few milivolts) even in the zones where the same power transistor is used. Secondly, the typical transistor DC current gain will also be different for each power transistor due to small differences during their construction process. This is shown in Figure 2.9 by the different slopes of the lines from zones 1 to 8.

Clearly, a photocell sensor calibration is a fundamental necessity for our smart lighting system because we are concerned about getting the same output voltage under the same input voltage for each zone respectively. A similar input-output response for each zone is critical for the performance of each control loop.

![Figure 2.9: Raw sensor data after applying a voltage sweep from 0 V to 10 V.](image)

16
2.4.1 Smart Light Calibration Algorithm

The sensor calibration algorithm implemented on our smart light testbed is adapted from [12]. This algorithm is an off-line sensor calibration that uses raw sensor data directly obtained from the experimental testbed. In this section, we will discuss how the sensor calibration is performed and provide a set of plots with calibrated sensor data for illustration.

From Figure 2.9, we can point out that the input-output voltage response of each zone presents a linear region for input voltages between 4 Volts and 7 Volts. Based on this assumption, the slope of the lines for each zone is calculated. Let us define the slope for zone $i$ as $m_i$, given as follows

$$m_i = \frac{y_{raw}(7) - y_{raw}(4)}{7 - 4} = \frac{y_{raw}(7) - y_{raw}(4)}{3} \quad (2.1)$$

Once the slope of each line is computed from Equation (2.1), the raw sensor data $y_{raw}(V)$ is scaled by the factor $1/m_i$ in order to obtain the scaled raw sensor data $y_{scaled}(V)$. Then, a bias value is added up to the scaled raw sensor data of each zone in order to achieve an operating linear region for input and output voltages between 4 Volts and 7 Volts. Let us define the bias voltage for zone $i$ as $b_i$, the calibrated sensor data for zone $i$ given by $y_{cal_i}(V)$ can be defined as

$$y_{cal_i}(V) := \frac{y_{raw_i}(V)}{m_i} + b_i \quad (2.2)$$

where

$$b_i := 4 - y_{scaled_i}(4) \quad (2.3)$$
From Equations (2.2) and (2.3), we know that we are required to compute $m_i$, $y_{scaled,i}$, and $b_i$ in order to obtain the calibrated sensor data. These algebraic parameters are obtained from a set of raw sensor data measured off-line, and then used to properly perform the sensor calibration in our control loop algorithm.

Figure 2.10 illustrates a set of calibrated sensor data after performing the photocell sensor calibration described in this section. We can point out that each zone presents a different bias level as well as a different final output voltage. Nevertheless, an input voltage of 4 Volts perfectly matches with an output voltage of 4 Volts and an input voltage of 7 Volts perfectly matches with an output voltage of 7 Volts. This graphical result supports the described sensor calibration algorithm and its importance for the control of our smart lighting system.

Figure 2.10: Calibrated sensor data after applying a voltage sweep from 0 V to 10 V.
In Figure 2.11, all the calibrated sensor data from zones $i = 1$ to $i = 8$ are plotted together for comparison. Clearly, there is a linear operating region between 4 Volts and 7 Volts that would be our desired range of operation for using our testbed.

Figure 2.11: Comparison of calibrated sensor data.
In this chapter we present an overview of the fundamental concepts of fuzzy control that we used for our smart light system. Moreover, we will discuss the fuzzy model reference learning controller (FMRLC), which is an adaptive fuzzy control algorithm, and its application to our smart light system. This chapter is mostly based on ideas adapted from [3, 4] to our smart lighting testbed.

3.1 Fuzzy System Overview

Fuzzy control has become an alternative to classical control schemes because the controller design does not depend on a mathematical model but on the knowledge that the control engineer has about how to accurately control the plant. In Figure 3.1, the block diagram of the fuzzy controller for smart lights is presented. Notice from Figure 3.1 that a fuzzy controller is formed by four main elements which include: a rule-base, an inference mechanism, a fuzzification interface, and a defuzzification interface.

First, the rule-base component includes a fuzzy logic quantification of the control rules defined by the control engineer or the process expert on how to obtain a good control of the plant. Second, the inference mechanism imitates the control engineer’s ability to interpret and use the knowledge on how to do a good control of the plant to
take a control decision. Third, the *fuzzification interface* is intended to transform the controller inputs into information that the inference mechanism is able to use in order to decide which rules are turned “on.” Last, the defuzzification interface is intended to use the output of the inference mechanism and transform it into compatible inputs for the plant to be controlled [3]. In the following section, we will discuss each of the components of a fuzzy controller adapted for our smart light testbed.

![Fuzzy Controller Diagram]

Figure 3.1: Fuzzy controller for smart lights (adapted from [3]).

### 3.1.1 Selecting Fuzzy Controller Inputs and Outputs

Let us consider the case where a human is trying to control the illumination level in a single room. We desire to design our fuzzy controller in such a way that it will automate how the person inside the room would control the smart light system. To begin with, the room user tells us (the control engineers designing the fuzzy controller) what information about the process she or he will use as inputs to make a decision on how to control the smart lights. Suppose that for our smart lighting testbed, the room user says that she or he will use


\[ e(t) = V_{ref}(t) - V_{out}(t) \quad (3.1) \]

and

\[ \int e(t)\,dt = \int (V_{ref}(t) - V_{out}(t))\,dt \quad (3.2) \]

as the input variables to the decision-making process. Of course, there are many different choices to choose as input (e.g., the derivative of the error \( e(t) \)) but this choice makes sense in a system with a fast transient response such as our smart light testbed. The next step is to identify the controlled variable. For our smart light system, we are concerned about controlling the illumination across the floor of the room via the applied voltage to the light bulbs so the input is given by \( V_{app}(t) \) as illustrated in Figure 3.2. In general, the controller designer wants to make sure that the controller will have the sufficient information available such that good control decisions can be made and the proper control inputs can be applied to the smart light system to regulate the illumination (i.e., output voltage) at the desired level.

![Figure 3.2: Human controlling a smart light (adapted from [4]).](image)
Now that we have selected the inputs and the output of the fuzzy controller, we can select the reference input. For our smart lighting tested, we will use the desired voltage level (i.e., illumination in the testbed floor) as the reference input. Once all the inputs and outputs for the fuzzy controller have been selected, we can create a general picture of our fuzzy control system as illustrated in Figure 3.3. Notice from Figure 3.3 that the fuzzy control system for our smart light system is composed of two inputs (i.e., the error $e(t)$ and integral of the error $\int e(t)dt$) and one output (i.e, applied voltage $V_{app}(t)$). Following the given fuzzy control system structure, we next seek to obtain a description on how to control the smart light system. Selecting the proper inputs and outputs for the fuzzy controller plays a key role in the design process. If we do not provide the necessary information to the fuzzy controller then obtaining a good rule base or inference mechanism might not be possible. In addition, even though that the proper information is given to the fuzzy control system for control decision making, this will be unnecessary if the controller is not able to affect the behavior of the controlled process via the selected process input. We have to understand that the selection of the controller inputs and outputs is very important for designing a proper fuzzy controller for our smart lighting system.

Figure 3.3: Fuzzy controller for smart light illumination problem (adapted from [4]).
3.1.2 Converting Control Knowledge into Rule Bases

Let us assume that the room user presented in Figure 3.2 knows a set of rules on how best to control the smart light system in a human language such as English. Based on these human language rules, we can write down “linguistic descriptions” in order to build our rule base.

Linguistic Descriptions

The linguistic description given by the room user can be divided into several parts. We will have “linguistic variables” that are related to each of the time-varying fuzzy controller inputs and outputs. For our smart lighting testbed, we have the following:

- “error” describes $e(t)$
- “integral-of-error” describes $\int e(t)dt$
- “applied voltage” describes $V_{app}(t)$

Notice that the quotes are used to emphasize the use of words or phrases as linguistic descriptions, but more importantly, that these variables are time-varying. For instance, let us assume that $e(t)$ takes on a value of 50 mV at $t = 0.5$ (i.e, $e(0.5) = 0.050V$), we can state that linguistic variables assume “linguistic values.” These linguistic variables change dynamically their values over time. Let us suppose that for our smart light system the “error”, “integral-of-error,” and “applied voltage” can take the following values:

- “neghuge”
- “neglarge”
- “negbig”

24
“negmed”
“negsmall”
“zero”
“possmall”
“posmed”
“posbig”
“poslarge”
“poshuge”

Notice that we are abbreviating the linguistic variables. For instance, “possmall” is an abbreviation for “positive small in value” and the same logic applies for the other variables. In this way, we can keep the linguistic descriptions as short as possible and they can be understood. We could also use integer values to represent the linguistic variables as follows:

“-5” to represent “neghuge”
“-4” to represent “neglarge”
“-3” to represent “negbig”
“-2” to represent “negmed”
“-1” to represent “negsmall”
“0” to represent “zero”
“1” to represent “possmall”
“2” to represent “posmed”
“3” to represent “posbig”
“4” to represent “poslarge”
“5” to represent “poshuge”
This way of representing the linguistic variables is simple and straightforward. However, we want to point out that the numeric values associated to each of the linguistic variables does not represent the actual value of the error in Volts, this representation provides a simple way to quantify the sign of the error and also tell us the differences between the size of each linguistic value. These “linguistic-numeric values” are quite useful when it comes down to coding our fuzzy controller algorithm into MATLAB/Simulink. Moreover, the linguistic variables and values give us a set of words or phrases that the room user can use to express her or his ideas about how to control the smart light system based on a decision-making approach, using as primary information the fuzzy controller inputs and outputs. Let us assume that for our smart light system \( V_{\text{ref}}(t) = 5 \) Volts and \( e(t) = V_{\text{ref}}(t) - V_{\text{out}}(t) \), therefore we have

\[
e(t) = 5 - V_{\text{out}}(t) \quad (3.3)
\]

and

\[
\int_0^t e(t) dt = \int_0^t (5 - V_{\text{out}}(t)) dt = 5t - \int_0^t V_{\text{out}}(t) dt \quad (3.4)
\]

Assuming that we are the room user of our smart light system, we will take a look at how to quantify certain behaviors with the given linguistic variables. For our smart light testbed, using a heuristic approach we can write down the following statements for quantifying different illumination settings within a room:

- The phrase “error is poshuge” can represent the case where the output voltage is significantly smaller than the desired reference input voltage.
• The phrase “error is negmed” can represent the case where the output voltage is slightly greater than the desired reference input voltage.

• The phrase “error is zero” can represent the case the the output voltage is very close to the desired reference input. In this linguistic quantification, we will assume that any value of error close enough to \( e(t) = 0 \) can be seen linguistically as “zero” because it is a better representation than “possmall” or “negsmall”.

• The phrase “error is poshuge and integral-of-error is possmall” can represent the situation where the output voltage is significantly smaller than the desired reference input voltage but the output voltage is getting closer to the desired reference input voltage.

• The phrase “error is negmed and integral-of-error is negsmall” can represent the situation where the output voltage is slightly greater than the desired reference input voltage but the output voltage is increasing compared to the desired reference input.

Clearly, quantifying the behavior of the controlled process requires a good understanding of its characteristics. This quantification is not easy for many physical processes, however, good linguistic quantifications will lead to the design of a good fuzzy controller that would be capable of taking the right decisions on how to control the plant properly.

Rules

Based on the linguistic variables that we have defined above, a set of rules (i.e., a rule base) can be constructed to represent the room user knowledge on how to best
control the plant. Essentially, for our smart lighting system we could write down the following rules:

1. **If** error is negsmall **and** integral-of-error is possmall **Then** applied voltage is zero

   This rule refers to the situation where the output voltage is slightly smaller than the desired reference input voltage and the output voltage is approaching the desired reference input voltage; thus, near zero voltage should be applied to the room light bulb since the output voltage is moving towards the reference.

2. **If** error is zero **and** integral-of-error is negsmall **Then** applied voltage is negsmall

   This rule refers to the situation where the output voltage is very close to the desired value (zero as a linguistic qualification does not state that $e(t) = 0$ precisely) and the output voltage is increasing compared to the desired reference input voltage; thus, a small negative voltage should be applied to keep the output voltage close to zero (a positive applied voltage could produce that the output voltage overshoot the desired value).

3. **If** error is possmall **and** integral-of-error is possmall **Then** applied voltage is posmed

   This rules refers to the situation where the output voltage is slightly smaller than the desired reference input value and the output voltage is increasing compared to the desired reference input voltage; thus, a positive medium voltage should be applied so we can adjust the output voltage to the proper desired value.
In general, all the three rules mentioned above are “linguistic rules” because they are composed only of linguistic variables and values. Moreover, linguistic values are not an exact representation of the given quantities that they refer to and linguistic rules are not exact either. Both the linguistic values and rules are ideas about how to best control that will not necessarily mean the same thing for two different persons. The linguistic rules are generally expressed in the following form as

\textbf{If} premise \textbf{Then} consequent

From the “linguistic rules” listed above, we can state that the premises are connected with the fuzzy controller inputs and are usually located on the left-hand side of the rules. In the other hand, the consequents are connected with the fuzzy controller outputs and are usually located on the right-hand side of the rules.

\textbf{Rule Bases}

Following the same idea to write down additional rules, we can build a set of rules for all possible cases. Since there is only a given number of linguistic variables and linguistic values, then there would be a fixed number of possible rules. For our smart lighting system, we have two inputs and eleven linguistic values for each of the given inputs, then there would be at most $11^2 = 121$ possible rules. Typically, a tabular representation is used as a convenient way to group all the possible rules whenever there are not more than two or three inputs to the fuzzy controller. The tabular representation for the set of possible rules for our smart lighting system is presented in Table 3.1. Notice from Table 3.1 that the content is composed by the linguistic-numeric consequents of the rules, the top row represents the linguistic-numeric premise for the integral-of-error and the left column represents the linguistic-numeric premise for the
error. For instance, the \((-1, +1)\) position, where the “\(-1\)” represents the row having “\(-1\)” as the numeric-linguistic value and the “+1” represents the column having “+1” as the numeric-linguistic value, has a 0 (“zero”) in the content of the table and defines the following rule

If error is negsmall and integral-of-error is possmall Then applied voltage is zero

which is defined by the rule 1 above. Table 3.1 provides the knowledge that the room user has about how to best control the smart light system given the error and its integral as inputs.

<table>
<thead>
<tr>
<th>(e(t))</th>
<th>(\int e(t)dt)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-5</td>
<td>-5 -5 -5 -5 -5 -5 -1 -3 -2 -1 0 1 0 1 0</td>
</tr>
<tr>
<td>-4</td>
<td>-5 -5 -5 -5 -5 -5 -4 -3 -2 -1 0 1 2 1 2</td>
</tr>
<tr>
<td>-3</td>
<td>-5 -5 -5 -5 -4 -3 -2 -1 0 1 2 3 3 4 3 4</td>
</tr>
<tr>
<td>-2</td>
<td>-5 -5 -4 -3 -2 -1 0 1 2 3 4 5 5 5 5 5 5</td>
</tr>
<tr>
<td>-1</td>
<td>-5 -4 -3 -2 -1 0 1 2 3 4 5 5 5 5 5 5 5 5</td>
</tr>
<tr>
<td>0</td>
<td>-5 -4 -3 -2 -1 0 1 2 3 4 5 5 5 5 5 5 5 5</td>
</tr>
<tr>
<td>1</td>
<td>-4 -3 -2 -1 0 1 2 3 4 5 5 5 5 5 5 5 5 5</td>
</tr>
<tr>
<td>2</td>
<td>-3 -2 -1 0 1 2 3 4 5 5 5 5 5 5 5 5 5 5</td>
</tr>
<tr>
<td>3</td>
<td>-2 -1 0 1 2 3 4 5 5 5 5 5 5 5 5 5 5 5</td>
</tr>
<tr>
<td>4</td>
<td>-1 0 1 2 3 4 5 5 5 5 5 5 5 5 5 5 5 5</td>
</tr>
<tr>
<td>5</td>
<td>0 1 2 3 4 5 5 5 5 5 5 5 5 5 5 5 5 5</td>
</tr>
</tbody>
</table>

Table 3.1: Rule table for smart lighting system.

Evidently, there is a very particular pattern in the rule consequents. First, there is a diagonal of zeros. Secondly, there is an evident symmetry in the table if we compare the upper and lower triangles formed next to the main diagonal of zeros respectively. This pattern is often found in many applications [4].
3.1.3 Fuzzy Numerical Interpretation of Knowledge

In the previous section, we quantified in an abstract way the knowledge that the room user has about on how to best control the smart lighting system. In this section, we will illustrate how to use fuzzy logic to provide a numerical interpretation to the meaning of linguistic descriptions such that the fuzzy controller can be automated to take decisions based on the control rules specified by the room user.

Membership Functions

The expression “membership functions” is used in fuzzy control theory to provide a procedure for assigning a numerical interpretation to the linguistic values. For instance, let us consider the “membership function” given in Figure 3.4. The function \( \mu \) measures the certainty that the error \( e(t) \) can be linguistically expressed as “possmall”. The word “certainty” is used here to express how certain we are that a rule is activated or not, and it does not have any connection with probability theory [4].

Figure 3.4: Membership function for linguistic value “possmall” (adapted from [4]).
Let us analyze how to interpret the various values of $e(t)$ from Figure 3.4, therefore we can get a good understanding of the functionality of a membership function:

- If $e(t) = -80$, then $\mu(-80) = 0$. This indicates that we are certain that $e(t) = -80$ is not “possmall”. Clearly, it is a negative value.

- If $e(t) = 20$, then $\mu(20) = 0.5$. This indicates that we are 50% certain that $e(t) = 20$ is “possmall”.

- If $e(t) = 40$, then $\mu(40) = 1.0$. This indicates that we are 100% certain that $e(t) = 40$ is exactly a “possmall” value.

- If $e(t) = 80$, then $\mu(80) = 0$. This indicates that we are certain that $e(t) = 80$ is not “possmall”. This value has to be numerically interpreted with a different linguistic value. Next, we will identify it as “poslarge”.

The membership function provide us a way to numerically interpret if values of $e(t)$ are members or not of a set of values that are called “possmall”, and this way we can come up with a meaning for the linguistic statement “error is possmall”. Similarly, we can use the same approach for the other linguistic statements used to define the rules. There are many available shapes to represent a membership function including bell-shaped functions, trapezoids, gaussians, sharp peaks, skewed triangles, and many others, but the triangular form provides a fairly simple numerical interpretation of the knowledge of the room user for our smart lighting system as well as providing an easier implementation of the fuzzy controller. The set of values that is represented by $\mu$ as “positive small” is called a “fuzzy set”. We will denote this fuzzy set as $A$. From Figure 3.4, we are 100% certain that $e(t) = 40$ is an element of $A$, but we are less certain that $e(t) = 80$ is an element of $A$. 

32
Next, we would like to introduce the term “universe of discourse”. Notice from Figure 3.4 that the vertical axis ($\mu$) represents the certainty and the horizontal axis represent the so called universe of discourse ($e(t)$). The universe of discourse for the input $e(t)$ gives us the range of all possible values of $e(t)$ that can be numerically interpreted with linguistics and fuzzy sets. In general, a universe of discourse provides the range of all possible values that the inputs and outputs can take on for all the given inputs and outputs of a fuzzy system.

Figure 3.5: Membership functions for smart lighting system (adapted from [4]).
Up to this point, we have shown how to associate the meaning of a linguistic value with its corresponding membership function or fuzzy set. Thus, we can state all the membership functions for all 33 linguistic values, eleven values for each input \( e(t) \) and \( \int e(t) dt \) and eleven for the output \( V_{app}(t) \). The membership functions for our smart lighting system are illustrated in Figure 3.5. Notice from Figure 3.5 that both the linguistic and linguistic-numeric values for each of the membership functions are provided respectively. The scales of the axes for \( e(t) \), \( \int e(t) dt \), and \( V_{app}(t) \) were chosen based on experimental results, and thus, they represent the best tuned values found during the implementation of the fuzzy controller in the smart lighting system. Clearly, notice that the universe of discourse for \( e(t) \) is totally different for the universe of discourse for \( \int e(t) dt \).

Let us take a closer look at both the universes of discourse of \( e(t) \) and \( \int e(t) dt \). Notice from Figure 3.5 that for the inputs \( e(t) \) and \( \int e(t) dt \) the outermost membership functions “saturate” at the maximum value of one. In this case, we are putting together all large positive and negative values in a linguistic description such as “poshuge” or “neghuge” respectively. More importantly, the values of the membership functions are time dependent. For instance, the value of \( e(t) \) could change from \(-120\) to \(120\) in a certain period of time, we can observe in Figure 3.5 that several membership functions will take on zero and nonzero values providing the certainty to which the corresponding linguistic value numerically interprets the current value of \( e(t) \). Assuming that \( e(t) = -120 \), we are 100% certain that the error is “negbig,” and if the value of \( e(t) \) starts decreasing to \(-90\), we will be less certain that the
error is “negbig” and more certain that it is “negmed”. Obviously, membership functions provide quantification of linguistic phrases that are associated with time-varying signals.

Finally, let us study the membership function for the “applied voltage” $V_{app}(t)$ universe of discourse in Figure 3.5. Again, the horizontal scale was chosen based on experimental analysis performed on the smart lighting system. Notice from Figure 3.5 that for the output $V_{app}(t)$, the membership functions at the outermost edges are not saturated for our fuzzy system. Even though this output membership function does not exactly match the values for the process input, they provided the best feasible performance during experimental testing.

**Interpreting the Membership Functions and Rules**

First, we can observe from Figure 3.5 that there is a clear pattern of center positions (i.e, where the triangle peak is equal to one) for both input membership functions, and for the output membership functions this pattern is not uniform. In order to achieve a uniform distribution for the output membership function centers we can choose the center values, which we will identify as $b_i$ where $i$ is the linguistic-numeric index for each corresponding membership function, as follows

$$b_i = 100 \left( \frac{i}{5} \right)$$

where $i = -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5$.

The selection of nonuniformly distributed membership functions for the output in Figure 3.5 represents the case that for small voltage errors, when the integral-of-error is also small, the room user will not apply a big voltage to the light bulb. Based
on the way the smart lighting system behaves, if the output voltage is close enough
to the desired reference input voltage and the output voltage is not getting bigger
fast, then small corrections to the applied voltage will be more effective for output
voltage regulation (i.e., illumination regulation). We will assume that the room user
has learned about the behavior of the smart lighting system by gaining experience as
a human-in-the-loop performing the control actions. For instance, the output voltage
sensor measurement is noisy due to its low cost, therefore for small voltage errors
the noise will make it look as there is a greater voltage error when we do not have
one. Moreover, let us say that when the error is “poshuge” and the integral-of-error is
“neghuge”, the room user will also apply zero voltage to the light bulb (see Table 3.1).
The room user came up with this rule because, while the output voltage is significantly
away from the desired voltage, the integral-of-error indicates that the output voltage
is moving towards the desired condition. Now, if the error is “poshuge” and the
integral-of-error is “neglarge”, then the applied voltage is “possmall” and looking at
Figure 3.5 this represents a very small voltage correction since the room user does
not want to considerably increase the applied voltage more than necessary because it
would involve more energy consumption.

Second, notice from the content of Table 3.1 the pattern of the output membership
functions, and you see that the room user will saturate the applied voltage either
positive (lower right corner of the rule base) or negative (upper right corner of the
rule base) if the error and the integral-of-error are considerably big in magnitude.
The decision of when to apply the maximum voltage (i.e., light bulb as its maximum
illumination) is based on the room user experience on regulating the illumination
within the room. If the room user does not perform the saturation soon enough, then
the desired reference input voltage might not be achieved. However, if the room user saturates the applied voltage too quick, then for relatively small errors a steady state voltage might not be able to be achieved.

Due to the unclear connection between the interpretation of control rules in the linguistic rule base presented in Table 3.1 and the membership functions information, a common scheme is used for generating the rule base table. Instead of listing the indices for the output membership functions, the center of the appropriate output membership functions are listed divided by a scaling factor, which for our smart lighting system is 100. In order to get the actual center from the given rule base table we will need to multiply the corresponding entry by 100. Table 3.2 illustrates this common scheme for a rule base table.

<table>
<thead>
<tr>
<th>$e(t)$</th>
<th>$\int e(t)dt$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-5</td>
<td>-1 -1 -1 -1 -1 -1 -1 -0.8 -0.6 -0.4 -0.2 0 0.2</td>
</tr>
<tr>
<td>-4</td>
<td>-1 -1 -1 -1 -1 -1 -0.8 -0.6 -0.4 -0.2 0 0.2 0.4</td>
</tr>
<tr>
<td>-3</td>
<td>-1 -1 -1 -1 -1 -0.8 -0.6 -0.4 -0.2 0 0.2 0.4 0.6</td>
</tr>
<tr>
<td>-2</td>
<td>-1 -1 -1 -1 -0.8 -0.6 -0.4 -0.2 0 0.2 0.4 0.6 0.8</td>
</tr>
<tr>
<td>-1</td>
<td>-1 -1 -1 -1 -0.8 -0.6 -0.4 -0.2 0 0.2 0.4 0.6 0.8 1</td>
</tr>
<tr>
<td>0</td>
<td>-1 -0.8 -0.6 -0.4 -0.2 0 0.2 0.4 0.6 0.8 1 1 1</td>
</tr>
<tr>
<td>1</td>
<td>-0.8 -0.6 -0.4 -0.2 0 0.2 0.4 0.6 0.8 1 1 1 1</td>
</tr>
<tr>
<td>2</td>
<td>-0.6 -0.4 -0.2 0 0.2 0.4 0.6 0.8 1 1 1 1 1</td>
</tr>
<tr>
<td>3</td>
<td>-0.4 -0.2 0 0.2 0.4 0.6 0.8 1 1 1 1 1 1</td>
</tr>
<tr>
<td>4</td>
<td>-0.2 0 0.2 0.4 0.6 0.8 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>5</td>
<td>0 0.2 0.4 0.6 0.8 1 1 1 1 1 1 1 1</td>
</tr>
</tbody>
</table>

Table 3.2: Rule table for smart lighting system (the content represents the output membership function centers where each element has to be scaled by 100).
The knowledge about voltage regulation can be directly subtracted from Table 3.2 in the following way:

1. If the voltage error and the integral-of-error are both significantly big (upper left and lower right corners of the rule base illustrated in Table 3.2), then apply the appropriate maximum voltage.

2. For zero $e(t)$ and $\int e(t)dt$, the applied voltage should be zero. If $e(t)$ and $\int e(t)dt$ move positive, then the applied voltage should be positive as well. Similarly, if $e(t)$ and $\int e(t)dt$ is negative, then the applied voltage should be negative. This particular case is included in the rule base for illustration purposes but it is not physically possible in the smart lighting system. For the case that $e(t)$ and $\int e(t)dt$ have opposite signs and depending on the magnitude of the error and the integral-of-error, the applied voltage will be either positive or negative.

3. For small values of $e(t)$ and $\int e(t)dt$, careful changes to the applied voltage has to be changed such that the corrections may not introduce significant changes in the output voltage (i.e., lowering the controller “gain” around zero so the noise is not amplified). Additionally, if the output voltage is changing fast enough to significantly reduce the voltage error, then careful changes to the applied voltage have to be performed in order to keep the control energy as low as possible, and thus, reducing energy consumption in the smart lights.

**Fuzzification**

The fuzzification process is the most simple step within a fuzzy controller. This process consists of obtaining a value for an input variable (e.g., $e(t)$ or $\int e(t)dt$) and computing the numeric values of the membership function(s) that are related to that
given input variable. Let us assume that $e(t) = 40$ and $\int e(t)dt = 1$, then via the fuzzification process we can find the values of the corresponding values of the input membership functions for these given inputs. For this case, we will have the following

$$\mu_{\text{possmall}}(e(t)) = 1$$  \hspace{1cm} (3.6)

with all others membership functions being zero and

$$\mu_{\text{zero}} \left( \int e(t)dt \right) = \mu_{\text{possmall}} \left( \int e(t)dt \right) = 0.5$$  \hspace{1cm} (3.7)

This procedure is similar to an "encoding" process performed on the fuzzy controller numeric input values based on the membership functions values [4]. This encoded information is used by the fuzzy inference process that is the next step within the fuzzy controller framework.

### 3.1.4 Deciding Which Rules to Use

In the following subsection, we will discuss about the inference mechanism illustrated in Figure 3.1 as a process of the fuzzy controller. The inference mechanism is generally formed by two steps [4]:

1. The premises of all the rules are compared to the fuzzy controller inputs to select which rules are associated with the current situation. This comparison process involves computing the certainty that each rule applies, and we will take more into account those rules that we are more certain apply to the current situation.

2. The control actions are decided using the rules that have been determined to apply at the current time. The control actions are connected with a fuzzy set
(or sets) that provide the certainty that the input to the smart light takes on several values.

The first step is discussed in this subsection, and the second step will be covered in the following subsection.

**Premise Numerical Interpretation via Fuzzy Logic**

In order to do the inference we must first provide a numerical interpretation for each of the rules with fuzzy logic. First of all, we have to quantify the meaning of the premises of the rules that are formed by more than one term, where each term is associated with a different fuzzy controller input. From Figure 3.6, we illustrate the membership functions associated with the corresponding premise term of the rule

**If** error is zero **and** integral-of-error is possmall **Then** applied voltage is possmall

![Membership functions of premise terms](image)

**Figure 3.6:** Membership functions of premise terms (adapted from [4]).
In the previous subsection, we considered on how to quantify the meaning of the linguistic terms “error is zero” and “integral-of-error is possmall” independently via the membership functions presented in Figure 3.5. Next, we are interested in how to quantify the linguistic premise “error is zero and integral-of-error is possmall.” As a consequence, we are concerned with how to provide a numerical interpretation for the logical “and” operation that put together the two linguistic terms. Let us illustrate how to provide a numerical interpretation for the “and” operation by assuming that $e(t) = 20$ and $\int e(t)dt = 0.5$. From Figures 3.5 and 3.6 we have

$\mu_{\text{zero}}(e(t)) = 0.5 \quad (3.8)$

and

$\mu_{\text{possmall}}(e(t)) = 0.25 \quad (3.9)$

Let us denote the certainty by $\mu_{\text{premise}}$. There are two usual ways to define the certainty of a premise:

- **Minimum**: define $\mu_{\text{premise}} = \min\{0.5, 0.25\} = 0.25$. This means that we are selecting the minimum of the two membership values.

- **Product**: define $\mu_{\text{premise}} = (0.5)(0.25) = 0.125$. This means that we are multiplying the two membership values.

In our implementation we used the *minimum* to compute the certainty of the premise. Clearly, both methods of quantifying the “and” operation in the premise show that we cannot get more certain about the interconnection of the two statements than about each individual term that form them up (i.e, $0 \leq \mu_{\text{premise}} \leq 1$).
Generally speaking, there would be a different premise membership function for each of the 121 rules of our given rule base, and each of these premises will be a function of \( e(t) \) and \( \int e(t) dt \) so for specific values of \( e(t) \) and \( \int e(t) dt \), we achieve a numerical interpretation of the certainty that each rule in the rule base applies to the current situation. Notice that both \( e(t) \) and \( \int e(t) dt \) change dynamically over time, therefore the values of \( \mu_{\text{premise}}(e(t), \int e(t) dt) \) for each rule will also dynamically change over time, and thus, the applicability of each rule base for specifying the applied voltage to the smart light will also change with time.

**Deciding Which Rules Are On**

In order to decide which rules are *on* or *not*, we have to state that a rule is “on at time \( t \)” if the corresponding premise membership function \( \mu_{\text{premise}}(e(t), \int e(t) dt) > 0 \). As a consequence, the inference mechanism attempts to find out which rules are *on* and then decides which rules are important to the current situation. After this step, the inference mechanism will look forward to interconnect the recommendations of all the rules that are on to provide a single conclusion.

For our smart light system, let us assume that \( e(t) = 0 \) and \( \int e(t) dt = 1.5 \). Notice from Figure 3.7 that the thick black vertical lines show the values for the inputs \( e(t) \) and \( \int e(t) dt \) in the their corresponding membership functions. Clearly, the certainty for \( e(t) \) is given by \( \mu_{\text{zero}}(e(t)) = 1 \) but for all other membership functions of the input \( e(t) \) the certainty is zero (i.e., all the other rules are “off”). For the \( \int e(t) dt \), we observe that \( \mu_{\text{zero}}(\int e(t) dt) = 0.25 \) and \( \mu_{\text{posssmall}}(\int e(t) dt) = 0.75 \), and that all the other membership functions of the input \( \int e(t) dt \) are *off*. From this information, we know that the rules that have the premise terms...
“error is zero”

“integral-of-error is zero”

“integral-of-error is possmall”

are on. For all other rules, we have that $\mu_{\text{premise}}(e(t), \int e(t)dt) = 0$. From Table 3.2, we know that the following rules are on:

1. **If** error is zero **and** integral-of-error is zero **Then** applied voltage is zero

2. **If** error is zero **and** integral-of-error is possmall **Then** applied voltage is possmall

Figure 3.7: Input membership functions with input values (adapted from [4]).
Obviously, Figure 3.7 provides a very useful graphical explanation of which rules are *on*. However, sometimes it is better to take a look at the rule base table to have a better idea of what rules are *on* and their corresponding consequences. From the two rules listed above, we have that Table 3.3 (a copy of Table 3.2) illustrates the consequents of the two given rules that are *on* by boxes drawn around them.

### 3.1.5 Inference Step: Deciding Conclusions

The inference mechanism is the intermediate step within the fuzzy controller framework presented in Figure 3.1. In this subsection, we look forward to provide mechanisms to decide which conclusions should be reached when the rules that are *on* are applied to determine what the applied voltage to the light bulb of the smart light system should be. In order to do so, we will take a look at the recommendations of each rule independently and then we will seek to combine all the given recommendations from all the rules to find out the applied voltage to the light bulb in the smart light system.

**Recommendations from One Rule**

Let us consider the case of the conclusion reached by the following rule

If error is zero and integral-of-error is zero Then applied voltage is zero

which for our explanation will be referred as “rule (1).” Applying the minimum to represent the premise, we have

\[
\mu_{\text{premise}(1)} = \min\{1, 0.25\} = 0.25
\]  

(3.10)
\[
\int e(t) dt
\]

<table>
<thead>
<tr>
<th>( e(t) )</th>
<th>(-5)</th>
<th>(-4)</th>
<th>(-3)</th>
<th>(-2)</th>
<th>(-1)</th>
<th>(0)</th>
<th>(1)</th>
<th>(2)</th>
<th>(3)</th>
<th>(4)</th>
<th>(5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(-5)</td>
<td>(-1)</td>
<td>(-1)</td>
<td>(-1)</td>
<td>(-1)</td>
<td>(-1)</td>
<td>(-0.8)</td>
<td>(-0.6)</td>
<td>(-0.4)</td>
<td>(-0.2)</td>
<td>(0)</td>
<td>(0.2)</td>
</tr>
<tr>
<td>(-4)</td>
<td>(-1)</td>
<td>(-1)</td>
<td>(-1)</td>
<td>(-1)</td>
<td>(-1)</td>
<td>(-0.8)</td>
<td>(-0.6)</td>
<td>(-0.4)</td>
<td>(-0.2)</td>
<td>(0)</td>
<td>(0.2)</td>
</tr>
<tr>
<td>(-3)</td>
<td>(-1)</td>
<td>(-1)</td>
<td>(-1)</td>
<td>(-1)</td>
<td>(-1)</td>
<td>(-0.8)</td>
<td>(-0.6)</td>
<td>(-0.4)</td>
<td>(-0.2)</td>
<td>(0)</td>
<td>(0.2)</td>
</tr>
<tr>
<td>(-2)</td>
<td>(-1)</td>
<td>(-1)</td>
<td>(-1)</td>
<td>(-1)</td>
<td>(-1)</td>
<td>(-0.8)</td>
<td>(-0.6)</td>
<td>(-0.4)</td>
<td>(-0.2)</td>
<td>(0)</td>
<td>(0.2)</td>
</tr>
<tr>
<td>(-1)</td>
<td>(-1)</td>
<td>(-1)</td>
<td>(-1)</td>
<td>(-1)</td>
<td>(-1)</td>
<td>(-0.8)</td>
<td>(-0.6)</td>
<td>(-0.4)</td>
<td>(-0.2)</td>
<td>(0)</td>
<td>(0.2)</td>
</tr>
<tr>
<td>(0)</td>
<td>(-1)</td>
<td>(-0.8)</td>
<td>(-0.6)</td>
<td>(-0.4)</td>
<td>(-0.2)</td>
<td>(0)</td>
<td>(0.2)</td>
<td>(0.4)</td>
<td>(0.6)</td>
<td>(0.8)</td>
<td>(-1)</td>
</tr>
<tr>
<td>(1)</td>
<td>(-0.8)</td>
<td>(-0.6)</td>
<td>(-0.4)</td>
<td>(-0.2)</td>
<td>(0)</td>
<td>(0.2)</td>
<td>(0.4)</td>
<td>(0.6)</td>
<td>(0.8)</td>
<td>(1)</td>
<td>(1)</td>
</tr>
<tr>
<td>(2)</td>
<td>(-0.6)</td>
<td>(-0.4)</td>
<td>(-0.2)</td>
<td>(0)</td>
<td>(0.2)</td>
<td>(0.4)</td>
<td>(0.6)</td>
<td>(0.8)</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
</tr>
<tr>
<td>(3)</td>
<td>(-0.4)</td>
<td>(-0.2)</td>
<td>(0)</td>
<td>(0.2)</td>
<td>(0.4)</td>
<td>(0.6)</td>
<td>(0.8)</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
</tr>
<tr>
<td>(4)</td>
<td>(-0.2)</td>
<td>(0)</td>
<td>(0.2)</td>
<td>(0.4)</td>
<td>(0.6)</td>
<td>(0.8)</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
</tr>
<tr>
<td>(5)</td>
<td>(0)</td>
<td>(0.2)</td>
<td>(0.4)</td>
<td>(0.6)</td>
<td>(0.8)</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
</tr>
</tbody>
</table>

Table 3.3: Rule table for the smart lighting system with rules that are “on” (square boxed). (The content of the table holds the output membership function centers were each element should be multiplied by 100).

From Equation (3.10), the notation \( \mu_{\text{premise}(1)} \) represents the certainty of the premise (i.e., \( \mu_{\text{premise}} \)) for rule (1). Therefore, we are 0.25 certain that this rule applies to the current situation. The rule tells us that if its premise is true, then the action provided by its consequent should be applied. For our given rule (1), the consequent is that “applied voltage is zero”. The membership function for this consequent is illustrated in Figure 3.8(a). The membership function for the conclusion provided by rule (1) that is denoted by \( \mu(1) \), is presented in Figure 3.8(b) and is defined as follows

\[
\mu(1)(V_{\text{app}}(t)) = \min\{\mu_{\text{premise}(1)}, \mu_{\text{zero}}(V_{\text{app}}(t))\}
\]  

(3.11)

where \( \mu_{\text{premise}(1)} = 0.25 \) is given in Equation (3.10). This membership function describes the “implied fuzzy set” for rule (1). This is nothing more than the conclusion
provided by rule (1). The reason for the use of the minimum operator to represent the implication is that we can be no more certain about our consequent than our premise [4].

![Figure 3.8](image_url)

**Figure 3.8:** (a) Consequent membership function and (b) implied fuzzy set with membership function $\mu_{(1)}(V_{app})$ for rule (1) (adapted from [4]).

Notice from Figure 3.8(b) that the membership function $\mu_{(1)}(V_{app}(t))$ is a function of $V_{app}(t)$ and that the minimum operation will typically “chop off the top” of the $\mu_{zero}(V_{app}(t))$ membership function to generate $\mu_{(1)}(V_{app}(t))$. If we have different values of $e(t)$ and $\int e(t)dt$, then there will be different values of the premise certainty $\mu_{premise(1)}$, therefore chopping off the top of the triangle at different points. Evidently, the membership function $\mu_{(1)}(V_{app}(t))$ is in general a time-varying function that provides a numerical interpretation on how certain rule (1) is that the applied voltage $V_{app}(t)$ should take on certain values. From Figure 3.8(b), we observe that rule (1) is most certain that the applied voltage should lie in a region around zero and tells us that it is certain that the applied voltage should not be too large in either the
positive or negative direction, which makes sense if we look closely to the linguistic meaning of the given rule. Generally speaking, the membership function $\mu_{(1)}(V_{app}(t))$ provides a numerical interpretation for the conclusion given by only rule (1) and only for the current $e(t)$ and $\int e(t)dt$. This numerical interpretation will vary over time as the inputs $e(t)$ and $\int e(t)dt$ change over time.

**Recommendation from Another Rule**

Let us consider the conclusion reached by the second rule that is

**If** error is zero and integral-of-error is possmall **Then** applied voltage is possmall

which for our explanation we will refer to as “rule (2).” Applying the minimum to represent the premise, we have

$$
\mu_{premise(2)} = \min\{1, 0.75\} = 0.75
$$

From Equation (3.12), the notation $\mu_{premise(2)}$ represents the certainty of the premise (i.e., $\mu_{premise}$) for rule (2). Therefore, we are 0.75 certain that this rule applies to the current situation. Note that we are much more certain that rule (2) is more related to the current situation than rule (1). For our given rule (2), the consequent is that “applied voltage is possmall”. The membership function for this consequent is illustrated in Figure 3.9(a). The membership function for the conclusion provided by rule (2) that is denoted by $\mu_{(2)}(V_{app}(t))$, is presented in Figure 3.9(b) and defined as

$$
\mu_{(2)}(V_{app}(t)) = \min\{\mu_{premise(2)} , \mu_{possmall}(V_{app}(t))\}
$$
where $\mu_{\text{premise}(2)} = 0.75$ as given in Equation (3.12). This membership function describes the implied fuzzy set for rule (2). This is nothing more than the conclusion provided by rule (2). Once again, as $e(t)$ and $\int e(t)dt$ vary over time so will the values of $\mu_{\text{premise}(2)}$ (i.e., different functions of $\mu(2)(V_{\text{app}}(t))$ will be obtained). Clearly, rule (2) is more certain that the control output should be a small positive value. As rule (2) has a premise membership function with a higher certainty compared to rule (1), then we will be more certain of the conclusion reached by rule (2).

![Figure 3.9: (a) Consequent membership function and (b) implied fuzzy set with membership function $\mu(2)(V_{\text{app}}(t))$ for rule (2) (adapted from [4]).](image)

By computing the premise certainty of for each rule, we have performed the operation of the inference mechanism presented in Figure 3.1. Evidently, the input to the inference process is given by the set of rules that are on and its output is defined by the set of implied fuzzy sets that describe the conclusions reached by all the rules that are on (i.e., rule (1) and (2) in our example).
3.1.6 Transforming Decisions into Actions

The defuzzication operation is the last step within the fuzzy controller framework shown in Figure 3.1. The defuzzification process operates on the implied fuzzy sets generated by the inference mechanism and combines their effects to provide the “the most certain” controller output (process input). The defuzzification operation can be seen as a “decoding” of the fuzzy set information created by the inference process (i.e., the implied fuzzy sets) into numeric fuzzy controller outputs that can be applied to the plant.

The best way to understand how the defuzzification process works is to draw all the implied fuzzy sets obtained from the inference process on one axis. Figure 3.10 shows the two implied fuzzy sets obtained in the previous subsection. Our objective is to determine the fuzzy controller output which we will denote by $V_{crisp}^{app}$. This controller output will be the best representation of the conclusions of the fuzzy controller that are described with the implied fuzzy sets.

![Figure 3.10: Implied fuzzy sets from rule (1) and (2) (adapted from [4]).](image)

Figure 3.10: Implied fuzzy sets from rule (1) and (2) (adapted from [4]).
There are many methods to actually perform the defuzzification process. In this subsection, we will describe the two methods tested in our smart light testbed. Firstly, we will introduce the “center of gravity” (COG) defuzzification method, and secondly, we will present the “center-average” defuzzification method.

**Combining Recommendations**

First, let us consider the “center of gravity” (COG) defuzzification method for combining the recommendations described by the implied fuzzy sets from all the rules that *on*. Let $b_i$ denote the center of the membership function for the implied fuzzy set for the $i^{th}$ rule. For our given example we have

$$b_1 = 0.0(100) = 0.0$$

and

$$b_2 = 0.2(100) = 20$$

as illustrated in Figure 3.10. Let us define

$$\int \mu_{(i)}$$

which represents the area under the membership function $\mu_{(i)}$. The COG defuzzification method computes the output $V_{app}^{\text{crisp}}$ as follows

$$V_{app}^{\text{crisp}} = \frac{\sum_i b_i \int \mu_{(i)}}{\sum_i \int \mu_{(i)}}$$

Equation (3.17) represents a classical formula to compute the center of gravity.
Second, let us consider the “center-average” defuzzification method. For this method we defined the controller output $V_{\text{crisp}}$ as

$$V_{\text{crisp}} = \frac{\sum_i b_i \mu_{\text{premise}(i)}}{\sum_i \mu_{\text{premise}(i)}}$$

(3.18)

where $b_i$ also represents the center of the membership function for the implied fuzzy set for the $i^{th}$ rule. Remember that for our design we have used the minimum to compute the premise certainty $\mu_{\text{premise}(i)}$. Equation (3.18) refers to the “center-average” defuzzification method because it represents a weighted average of the center values of the membership functions of the implied fuzzy sets which are also considered the output membership function centers. In general, the center-average method substitutes the areas of the implied fuzzy sets that used in the COG method with the values of $\mu_{\text{premise}(i)}$. This replacement is acceptable because the area of the implied fuzzy sets is typically proportional to $\mu_{\text{premise}(i)}$ because $\mu_{\text{premise}(i)}$ is used to define which area under the triangle we are interested when COG is implemented for our design. There is an ambiguity on which defuzzification method should be selected since there are not clear guidelines in the literature. From the implementation results, we have decided to use the center-average for our smart lighting system.

3.2 Design Case: Direct PI Fuzzy Control for Smart Lights

Since the design of a fuzzy controller has to be performed in an ad-hoc manner, we will present for our smart light system a typical procedure used in the design as well as redesign of a fuzzy controller.
3.2.1 Implementation of a Direct PI Fuzzy Controller

Generally, the best way to test a fuzzy controller design is via simulation. For our smart lighting system, we do not have a mathematical model that describes the behavior of the plant. As a consequence, we implemented our designed fuzzy controller directly on the plant and proceeded to adjust the controller gain during implementation. We programmed our fuzzy controller in MATLAB/Simulink.

Normalization and Scaling

The fuzzy controller implemented in our experimental testbed is illustrated in Figure 3.11. Notice from Figure 3.11 that we have added the gains $g_1$ and $g_2$ at the inputs and $g_0$ at the output of the fuzzy controller. These gains have been added because they are useful in tuning the fuzzy controller. The gains can scale the horizontal input and output axes of the fuzzy controller. In our implementation, we “normalized” the input and output universe of discourse. For our smart lighting system, we changed the membership functions to those illustrated in Figure 3.12 (i.e., normalize to an interval $\pm 1$). Using the scaling gains given in Figure 3.12, we will be able to implement the membership functions presented in Figure 3.5.

Figure 3.11: PI fuzzy controller for smart lights with scaling gains.
The scaling gains play an important key role in the performance of the fuzzy controller. Notice that the scaling gain $g_1$ in the input is equivalent to scaling the horizontal axis of the $e(t)$ universe of discourse by $1/g_1$. The scaling $g_1$ will introduce the following effects:

1. If $g_1 = 1$, the membership functions are not changed, therefore there is no change on the meaning of the linguistic values.

2. If $g_1 < 1$, the membership functions are uniformly “spread out” by a factor of $1/g_1$. This introduces some changes in the meaning of the linguistics. For instance, “poshuge” is now described by a membership function that represents much larger numbers.

3. If $g_1 > 1$, the membership functions are uniformly “contracted.” This introduces some changes in the meaning of the linguistics. For example, “poshuge” is now described by a membership function that represents much smaller numbers.

Similarly, the scaling gain $g_2$ will have the same effects for the $\int e(t)dt$ universe of discourse. For the output universe of discourse, the scaling gain $g_0$ is a multiplying factor to the horizontal $V_{app}(t)$ axis. For our smart light system, we found after several trials and errors that the following scaling gains provide the best tuned performance

$$g_1 = 0.005 \quad g_2 = 0.10 \quad g_0 = 100 \quad (3.19)$$

The code was implemented in such a way that the execution time for the fuzzy control algorithm is minimized to guarantee real-time execution in the experimental testbed.
3.2.2 Direct PI Fuzzy Controller Mapping Surface

One of the most important characteristics of a fuzzy controller compared to a linear controller is its control surface. Fuzzy controllers can generate a nonlinear control surface which is desired for some control applications.

Figure 3.12: Normalized universes of discourse for fuzzy controller for smart light system (scaling gains are given in boxed values).
Figure 3.13 shows the nonlinear control surface implemented by the direct PI fuzzy controller. This control surface provides another way to view the room user expertise on how to control the output voltage (i.e., room illumination). For the case of a simple proportional-integral (PI) controller, the control surface is described by a plane in three dimensions. By properly tuning the PI controller gains, this linear PI controller can replicate the same shape as the fuzzy controller surface near the origin. Thus, the fuzzy controller will behave similarly to the PI controller provided its inputs are small and this was achieved during implementation of a simple PI controller for the experimental testbed. In general, it is not possible that a linear PI controller can replicate the nonlinear control surface presented in Figure 3.13.

Figure 3.13: Nonlinear control surface by direct PI fuzzy controller for smart lights.
3.3 Adaptive Fuzzy Control: Emulating Expertise Learning

The performance of a fuzzy controller (and most controllers) is very dependent on the plant and controller parameters. As we discussed in Chapter 2, we look forward to test our fuzzy controller under three partition settings (i.e., full, half, and no partition). The idea is to come up with a fuzzy controller that will automatically adapt the controller parameters (i.e., rule-base for a fuzzy controller) to those possible changes in the testbed. In addition, the smart light testbed is characterized by its simplicity and low cost. Therefore, rejection of sensor noise and plant disturbances will be a key characteristic of our fuzzy controller. Generally speaking, there are two main approaches to adaptive control: direct and indirect adaptive control schemes. We will implement for our smart light system a case of a direct adaptive control scheme as the one shown in Figure 3.14.

![Figure 3.14: Direct adaptive control block diagram (adapted from [3]).](image)
Notice from Figure 3.14 that in the direct approach, the “adaptation mechanism” looks to the controller output ($V_{app}$) and the plant output ($V_{out}$) and adjusts the parameters of the controller to maintain the performance even if there are changes in the smart light testbed. Typically, the desired system performance is characterized with a “reference model” and the idea is that the fuzzy controller will make the closed-loop system behave as the reference model even though there are plant parameter changes.

### 3.3.1 Fuzzy Model Reference Learning Control

In this subsection, we will present the fuzzy model reference learning control (FMRLC) which represents a “learning system” intended to tune direct fuzzy controllers. The block diagram of the FMRLC is presented in Figure 3.15. Notice from Figure 3.15 that the FMRLC has four main components: the plant, the fuzzy controller to be tuned, the reference model, and the learning (i.e., adaptation) mechanism. The FMRLC observes the data from the fuzzy control system, which means the reference input $r(kT)$ and the plant output $y(kT)$ where $T$ is the sampling period of the digital computer. From the measured numerical data, the FMLRC studies the fuzzy control system’s current performance and automatically adjusts (i.e., tunes) the fuzzy controller such that the closed-loop system (mapping from $r(kT)$ to $y(kT)$) behaves like the given reference model (mapping $r(kT)$ to $y_m(kT)$). In general, the fuzzy controller loop (the bottom of Figure 3.15) makes $y(kT)$ track $r(kT)$ by adjusting $u(kT)$ and the learning control loop (the top of Figure 3.15) makes the output of the plant $y(kT)$ track the output of the reference model $y_m(kT)$ by tuning the fuzzy controller parameters.
The Fuzzy Controller

For our smart lighting system, the inputs to the fuzzy controller are the error $e(kT) = r(kT) - y(kT)$ and the integral-of-error $c(kT)$.

$$c(kT) = c(kT - T) + T \cdot e(kT - T)$$

so that we have a PI fuzzy controller. Looking at Figure 3.15, we have the scaling gains $g_e$, $g_c$, and $g_u$ for the error $e(kT)$, integral-of-error $c(kT)$, and the controller output $u(kT)$ (plant input), respectively. These scaling gains correspond to the scaling gains $g_1$, $g_2$, and $g_0$ that we already discussed earlier in this chapter.
Rule Base: The rule base for the fuzzy controller has rules that can be expressed as

\[ \text{If } \tilde{e} \text{ is } \tilde{E}^j \text{ and } \tilde{c} \text{ is } \tilde{C}^l \text{ Then } \tilde{u} \text{ is } \tilde{U}^m \]

where \( \tilde{e} \) and \( \tilde{c} \) represent the linguistic variables connected with the controller inputs \( e(kT) \) and \( c(kT) \) respectively, \( \tilde{u} \) represents the linguistic variable connected with the controller output \( u(kT) \), \( \tilde{E}^j \) and \( \tilde{C}^l \) represent the \( j^{th} \) (\( l^{th} \)) linguistic value connected with \( \tilde{e} \) (\( \tilde{c} \)), and \( \tilde{U}^m \) represent the consequent linguistic value connected with \( \tilde{u} \). As a general example, one rule for the fuzzy controller could be

\[ \text{If error is positive-small and integral-of-error is negative-large} \]

\[ \text{Then plant input is positive-huge} \]

for this example case \( \tilde{e} = \text{“error”} \), \( \tilde{E}^1 = \text{“positive-small”} \), \( \tilde{c} = \text{“integral-of-error”} \), \( \tilde{C}^{-3} = \text{“negative-large”} \), \( \tilde{u} = \text{“plant-input”} \), and \( \tilde{U}^5 = \text{“positive-huge”} \). As presented for the direct fuzzy controller, we will use for the FMRLC the same standard selection of all the membership functions for all the input universes of discourse as the one illustrated in Figure 3.16. Again, we implemented 11 membership functions on each
of the two input universe of discourse, where we would have 121 rules in the rule base. Also, we used the minimum to represent the interconnection between the premise and the implication and the standard center-average defuzzification method due to the computational advantages of this approach.

**Rule Base Initialization:** The input membership functions are used to describe the premises of the rules that characterize the various situations in which rules should be applied or not. The input membership functions are left fixed and are not adjusted by the FMRLC. There are two main choices about how to initialize the membership functions for the output universe of discourse since this is assumed to be unknown because this is what the FMRLC will automatically tune as it “learns” about the process. First, we can choose as initial values for the output universe of discourse between $[-1, 1]$ to be triangular-shaped membership functions with base widths of 0.4 and all centered at the origin. This selection represents the case where the fuzzy controller does not know anything about how to control the plant so for all the rules the output will be zero ($u = 0$). Since in our implementation we first tried a direct PI fuzzy controller in the plant, we know how to initialize the output universe of discourse such that the fuzzy controller will be “more knowledgeable” about initially controlling the plant.

**The Learning Mechanism**

The learning mechanism seeks to adjust the rule base of the direct fuzzy controller (bottom part of Figure 3.15) such that the closed-loop system follows the reference model. In our implementation we used a first order transfer function as our reference model. The rule base adjusting process proceeds by acquiring data from the controlled process, the reference model, and the direct fuzzy controller. The learning
mechanism is formed by two main components: a “fuzzy inverse model” and a “rule base modifier.” The fuzzy inverse model does the mapping of \( y_e(kT) \), that represents the error of the plant out \( y(kT) \) with the desired behavior, with changes in the process inputs \( p(kT) \) that are computed to force \( y_e(kT) \) to zero. The rule modifier takes care of updating the fuzzy controller’s rule base such that the required changes to the process inputs are introduced.

**Fuzzy Inverse Model**: The fuzzy system presented in the top of Figure 3.15 is called the “fuzzy inverse model” because information about the plant inverse dynamics is implemented in its specification. Generally speaking, this “fuzzy inverse model” can be seen as a standard fuzzy system part of the adaptation loop in Figure 3.15 that is trying to select the best \( p(kT) \) such that the error \( (y_e(kT)) \) is minimized.

Notice from Figure 3.15 that the fuzzy inverse model also presents scaling gains which are denoted with \( g_{y_e} \), \( g_{y_c} \), and \( g_p \). We will discuss the selection of these gains below. The inputs to the fuzzy inverse model are given by the error \( y_e(kT) \) and the integral-of-error \( y_c(kT) \); therefore the rule base has rules of the following form

\[
\text{If } \tilde{y}_e \text{ is } Y_e^j \text{ and } \tilde{y}_c \text{ is } Y_c^l \text{ Then } \tilde{p} \text{ is } \tilde{P}^m
\]

where \( Y_e^j \) and \( Y_c^l \) represent the linguistic values and \( \tilde{P}^m \) represents the linguistic value connected with the \( m^{th} \) output fuzzy set. For our fuzzy inverse model, we used membership functions for the input universe of discourse illustrated in Figure 3.16, symmetric triangular-shaped membership functions for the output universe of discourse, minimum to compute the premise and implication, and center-average defuzzification method.

In general, the selection of the fuzzy inverse model is dependent in the application where the FMRLC is implemented. However, there are some guidelines that we can
follow. Typically, the fuzzy inverse model has a similar structure as the direct fuzzy controller. Usually, the initial guess for the input scaling gains are obtained in a similar manner as for the standard fuzzy controller. For the case of $g_p$, there are two possibilities. First, we can select $g_p = 0$ and try to tune the fuzzy controller to get a reasonable performance for the nominal system. In our smart light system, we were able to actually tune the fuzzy controller by selecting $g_p = 0$. In some cases, it is recommended to start with a small value of $g_p$ (which represents the “adaptation gain”) and tune the fuzzy control system gains following conventional controller tuning guidelines. Once a desirable performance is achieved, the scaling gain $g_p$ can be slowly increased until a good adaptation speed is obtained.

**Rule Base Modifier:** Using the information given by the output of the inverse fuzzy model $p(kT)$, the rule base modifier adjusts the rules of the direct fuzzy controller such that the previously used control action will be changed by the amount $p(kT)$. Let us consider the previously calculated control action $u(kT - T)$ and assume that it produced a current good or bad system performance such that the plant output $y(kT)$ did not match the reference model output $y_m(kT)$. For illustration purposes, we will assume that it will take one step (i.e., $d = 1$) for the plant input to affect the plant output. For our particular example, $e(kT - T)$ and $c(kT - T)$ would have been the error and the integral of error that were input to the fuzzy controller at that moment. By adjusting the direct fuzzy controller’s rule base, we seek to force the fuzzy controller to generate a desired output $u(kT - T) + p(kT)$ that *we should have put in at time $kT - T$ to force $y_e(kT)$ to be smaller*. If we do get similar values for the error and the integral of error, then the input to the plant will be one such that it will reduce the error between the reference model and the plant output.
In our smart light system we are using symmetric output membership functions for the fuzzy controller, therefore $b_m$ represents the center of the membership function connected with $\tilde{U}^m$. The rule base modification is produced by shifting the centers $b_m$ of the membership functions of the output linguistic value $\tilde{U}^m$ that are connected with the fuzzy controller rules that recommended the previous control action $u(kT - T)$. This involves a two-step procedure:

1. Find all the rules in the fuzzy controller with positive premise certainty, which means

$$\mu_i(e(kT - T), c(kT - T)) > 0$$

(3.21)

and identify this as the “active set” of rules at time $kT - T$. We can build this active set by looking at the indices of the input membership functions of each rule that is on.

2. Let $b_m(kT)$ represent the center of the $m^{th}$ output membership function at time $kT$. For all rules in the active set, use the following update formula

$$b_m(kT) = b_m(kT - T) + p(kT)$$

(3.22)

which modifies the output membership function centers. Rules that are not in the active set will not have their output membership functions modified.

**Heuristic Robustification**

In order to achieve a stable adaptive control, it is necessary to implement a “robustification” mechanism that can guarantee a stable operation of the adaptive system for uncertain plants. For our smart light system, we implemented an heuristic method to try to obtain robustification. Practically speaking, the inverse model has
to be defined in such a way that when the response of the plant \((y(kT))\) is following
the output of the reference model \((y_m(kT))\), the fuzzy inverse model shuts down the
learning (adaptation) mechanism. Following this idea, once the error input \((y_e(kT))\)
gets close enough to zero, the output of the fuzzy inverse model becomes zero. By
implementing this characteristic in our fuzzy inverse model we can guarantee the sta-
bility of the overall closed-loop system. This strategy can be coded in the following
way

\[
\text{If } |p(kT)| < \epsilon_p \text{ Then } p(kT) = 0 \hspace{1cm} (3.23)
\]

where \(\epsilon_p > 0\) is a small number specified during the initialization of the fuzzy inverse
model.

### 3.4 Adaptive PI Fuzzy Control for Smart Lights

In this section, we show the implementation of an adaptive PI fuzzy controller
for our smart light system. We will provide some general details about the design of
a FMRLC for the smart lights and illustrate the adaptation ability of the adaptive
fuzzy controller via the implemented nonlinear control surface for two case scenarios.

#### 3.4.1 Implementation of the FMRLC for Smart Lights

For our smart light system, we have selected the direct fuzzy controller inputs to
be the following

\[
e(kT) = V_{ref}(kT) - V_{out}(kT) \hspace{1cm} (3.24)
\]

and

64
\[ c(kT) = c(kT - T) + T \cdot e(kT - T) \]  \hspace{1cm} (3.25)

where \( V_{ref}(kT) \) is the desired input reference voltage. We selected \( T = 0.001 \) seconds for our implementation in the dSPACE board. As we know, the controller output is given by the output voltage \( V_{out}(kT) \) of the smart light. We implemented 11 uniformly spaced triangular membership function for each controller input (i.e, for \( e(kT) \) and \( c(kT) \)) as illustrated in Figure 3.16, which represents a normalized universe of discourse as the one discussed for the direct fuzzy controller in Section 3.2. We selected the scaling gains for the direct fuzzy controller to be \( g_e = 0.005 \), \( g_c = 0.10 \), and \( g_u = 100 \). The output membership functions were defined to be symmetric and triangular shaped with a base width of 0.4 on a normalized output universe of discourse as well. The reference model is selected to be

\[ \frac{V_m(s)}{V_{ref}(s)} = \frac{200}{s + 200} \]  \hspace{1cm} (3.26)

where \( V_m(s) \) indicates the desired system performance for the output voltage of the smart light \( V_{out}(t) \). For the fuzzy inverse model, the inputs are selected to be

\[ V_e(kT) = V_m(kT) - V_{out}(kT) \]  \hspace{1cm} (3.27)

and

\[ V_c(kT) = V_e(kT) + T \cdot V_e(kT - T) \]  \hspace{1cm} (3.28)

Similarly to the fuzzy model, we implemented 11 fuzzy sets with symmetric and triangular shaped membership functions, which are also evenly distributed on the
corresponding universes of discourse as given in Figure 3.16. For the inverse fuzzy model, we will have rules of the following form

\[
\text{If } \tilde{V}_e \text{ is } \tilde{V}_e^i \text{ and } \tilde{V}_c \text{ is } \tilde{V}_c^j \text{ Then } \tilde{p} \text{ is } \tilde{P}^m
\]

The output membership function centers are denoted by \( c_{i,j} \) for the \( i^{th} \) membership function of \( \tilde{V}_e \) and the \( j^{th} \) membership function of \( \tilde{V}_c \). The rule base for the inverse fuzzy model is defined in Table 3.4.

<table>
<thead>
<tr>
<th>( c_{i,j} )</th>
<th>( \tilde{V}_c^j )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V_e^i )</td>
<td>-5</td>
</tr>
<tr>
<td>-5</td>
<td>-1</td>
</tr>
<tr>
<td>-4</td>
<td>-1</td>
</tr>
<tr>
<td>-3</td>
<td>-1</td>
</tr>
<tr>
<td>-2</td>
<td>-1</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>0</td>
<td>-0.8</td>
</tr>
<tr>
<td>1</td>
<td>-0.6</td>
</tr>
<tr>
<td>2</td>
<td>-0.4</td>
</tr>
<tr>
<td>3</td>
<td>-0.4</td>
</tr>
<tr>
<td>4</td>
<td>-0.2</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3.4: Rule table for smart lights fuzzy inverse model.

Notice from Table 3.4 that \( V_e^i \) represents the \( i^{th} \) fuzzy set connected with the error signal \( V_e \), and \( V_c^j \) represents the fuzzy set connected with the integral of error signal \( V_c \). The content of the table provides the center values of the symmetric triangular membership functions \( c_{i,j} \) with base widths 0.4 for output fuzzy set \( \tilde{P}^m \) on the normalized universe of discourse.
The scaling gains for the fuzzy inverse model were selected as \( g_{y_e} = 0.0005 \), \( g_{y_c} = 0.001 \), and \( g_p = 1 \) respectively, following the ideas for tuning these gains presented in [5]. We used Equation (3.22) to compute the update of the base of the output membership functions. Also, we selected \( d = 1 \) and implemented the heuristic robustification principle where if

\[
|p(kT)| < 0.01g_p
\]  \hspace{1cm} (3.29)

then we set \( p(kT) = 0 \). In this sense, we seek to shut down the learning mechanism when the error between the plant out and the reference model is small. Notice from Equation (3.29) that the robustification term depends in the scaling gain \( g_p \), therefore as we tune the fuzzy controller we are also limiting the updates that can be introduced to the direct fuzzy controller.
Chapter 4: Simple Fault Tolerance Test for Smart Lights

In this chapter, we develop a series of real-time experiments in order to perform a simple fault tolerance test for both the direct PI fuzzy and direct adaptive PI fuzzy controllers described in the previous chapter. This test is being performed in order to study the controller’s performance under a specific fault test with no partitions between zones such that a proper operation of the smart lights can be guaranteed for a commercial application. Moreover, we are more concerned in the capability of the “learning” control algorithm (i.e., FMRLC or direct adaptive fuzzy) to perform on-line adaptation of the fuzzy rule-base to overcome a system failure compared to the performance of the nonadaptive control algorithm (i.e., direct fuzzy). Implementation results for a simple fault tolerance are presented in two main cases: single zone and multiple zones light bulb failure. For multiple zones light bulb failure, one case for three zones failure and two cases for four zones failure are presented in this chapter.

4.1 Single Zone Light Bulb Failure

In this section, we present the results obtained by introducing a simple failure characterized by shutting down the light bulb at a single zone in the testbed with no partitions between zones. As an initial approach, we study this case to analyze the capability of the two implemented fuzzy control algorithms to overcome a simple
system failure. For this failure, the two fuzzy controllers are able to regulate the desired light levels in the remaining zones. From this point, more developed failures are implemented to test the smart lights system near to complete failure.

### 4.1.1 Direct PI Fuzzy Controller Results

When a light bulb failure is introduced in the experimental testbed at $t = 30$ seconds in zone 1 (i.e., upper left corner), the direct PI fuzzy controller is able to maintain the uniform lighting in the remaining seven zones as illustrated in Figure 4.1. Clearly, the light level in zone 1 drops to a fixed value given by the cross-illumination effect imposed by the neighboring zones.

![Graphs showing light levels at each zone for the direct PI fuzzy controller with no partition between zones under zone 1 light bulb failure.](image)

**Figure 4.1:** Light levels at each zone for the direct PI fuzzy controller with no partition between zones under zone 1 light bulb failure.
Notice in Figure 4.1 that very small undershoots can be seen in zones 2, 3, 4, 5, and 7 due to the presence of the failure produced in zone 1. By looking at the applied voltage level to the light bulbs (i.e., control signal) on each zone for this particular case given in Figure 4.2, we can observe how the controller reacts to the zone 1 failure after \( t = 30 \) seconds. Zones 2, 3, 4, 5, and 7 control signals show a significant change in the shape of the applied voltage which is connected with the presence of the undershoots mentioned above. These undershoots are not desirable but they are so small and fast that it will not even be perceptible by a human eye. Moreover, this unbalance introduced by the zone 1 failure is clearly not perturbing the light sensors in zones 6 and 8 showing the disturbance rejection of the direct fuzzy controller.

Figure 4.2: Applied voltage level to the light bulbs at each zone for the direct PI fuzzy controller with no partition between zones under zone 1 light bulb failure.
### 4.1.2 Direct Adaptive PI Fuzzy Controller Results

As in the previous subsection, a light bulb failure is generated in zone 1 at $t = 30$ seconds. The direct adaptive PI fuzzy controller is capable to keep the uniform lighting in the remaining seven zones as shown in Figure 4.3.

![Figure 4.3: Light levels at each zone for the direct adaptive PI fuzzy controller with no partition between zones under zone 1 light bulb failure.](image)

From Figure 4.3, the presence of a learning control algorithm is evident. The existence of a very fast overshoot for the raising reference and a very fast undershoot for the falling reference is caused by the way the inverse fuzzy model gains are tuned. This does not mean that this is a poor performance, in fact, the adaptive fuzzy controller...
controller is tuned in such a way that is fast enough to adapt to abrupt changes in the plant parameters (e.g., light bulb failures) as is presented in the following sections.

The applied voltage level to the light bulbs on each zone for this particular case is given in Figure 4.4. Again, it is noticeable the faster response of the adaptive fuzzy controller as well as a better disturbance rejection. Compared with the results for the direct fuzzy controller (i.e., Figure 4.1), very fast undershoots are only presented in the zones 2, 3, and 4 (i.e., immediate neighboring zones of zone 1). This result points out that a properly tuned adaptive control algorithm is a good candidate for controlling a smart lights system. However, further exploration on fault tolerance has to be performed in order to support the selection of a robust control algorithm.

Figure 4.4: Applied voltage level to the light bulbs at each zone for the direct adaptive PI fuzzy controller with no partition between zones under zone 1 light bulb failure.
4.2 Multiple Zones Light Bulb Failure

In this section, we introduce the implementation results obtained by performing a simple bulb-off failure in multiple zones of the testbed with no partitions between zones. This failure is generated by shutting down the light bulb in multiple zones for three selected cases. First, the case when three zones simultaneously fail is presented. Then, two different cases are given when four zones simultaneously fail. A detailed response of the two fuzzy algorithms when the failure happens is also illustrated for comparison. For this multiple zone failure, the learning control algorithm is expected to track the desired light level in the remaining controlled zones, and shows a faster response to the system disturbance as well as a smaller undershoot.

4.2.1 Direct PI Fuzzy Controller Results

In this subsection, we present the results for the direct PI fuzzy controller. As stated before, a light bulb failure is generated in the experimental testbed at \( t = 30 \) seconds in multiple zones depending in the case on study. For all the given cases, the direct PI fuzzy controller is able to keep the uniform lighting in the remaining controlled zones.

First, a light bulb failure is introduced in zones 1, 3, and 4 at \( t = 30 \) seconds. For this case, the direct PI fuzzy controller keeps track of the desired reference lighting level even after three light bulb failures are generated in the experimental smart lights testbed as illustrated in Figure 4.5.

From Figure 4.5, we can appreciate how the voltage level in the uncontrolled zones (i.e., zones 1, 3, and 4) drop to a fixed value which is not the same for each zone because of the different cross-illumination effects introduced by the controlled zones
(i.e., zones 2, 5, 6, 7 and 8). Clearly, zone 1 presents the lowest light level because it is farther away from the lights with the major cross-illumination contributions (i.e., zones 5, 6, 7, and 8). Then, assuming that the cross-illumination from zones 5, 6, 7 and 8 has the same contribution on zones 3 and 4, this justifies why zone 4 has the highest fixed light level (followed by zone 3) due to its proximity to zone 2. Furthermore, there are visibly undershoots in the controlled zones just when the three zones failure occurs. The size of the undershoot is related to how close the controlled zone is to an uncontrolled zone and these undershoots are higher than the ones obtained for the previous single zone case. This illustrate the high coupling between zones due to no partitions between each zone in the tested.

Figure 4.5: Light levels at each zone for the direct PI fuzzy controller with no partition between zones under zones 1, 3, and 4 light bulb failure.
In Figure 4.6, the applied voltage level to the light bulbs on each zone for this particular case of multiple zones failure is presented. This illustrates the occurrence of the light bulb failure in zones 1, 3, and 4 and how the direct fuzzy controller reacts to the disturbance. Obviously, the high peak of the undershoots is reflected on this figure by an abrupt increase of the applied voltage by the direct fuzzy controller.

Figure 4.6: Applied voltage level to the light bulbs at each zone for the direct PI fuzzy controller with no partition between zones under zones 1, 3, and 4 light bulb failure.

A detailed view of the light level in the controlled zones (i.e., zones 2, 5, 6, 7 and 8) is shown in Figure 4.7. In the worst case, an undershoot with a peak of approximately 0.45 Volts is observed in zones 2 and 6, and the zones 5, 7 and 8 shows a peak of less than 0.25 Volts of undershoot. In addition, zones 7 and 8 have a very small overshoot.
(less than 1%) respectively. It takes approximately 0.625 seconds for the direct fuzzy controller to overcome the undershoot in zones 2, 5 and 6. For zones 7 and 8, it takes approximately 0.3125 seconds for the controller to recover from the undershoot, and approximately 0.625 seconds and 0.3125 seconds to return from the overshoot in zone 7 and 8 respectively.

Figure 4.7: Light level in controlled zones 2, 5, 6, 7, and 8 for the direct PI fuzzy controller with no partition between zones under zones 1, 3, and 4 light bulb failure.

Secondly, a light bulb failure is produced in zones 1, 4, 5, and 8 at \( t = 30 \) seconds. Again, for this specific case the direct PI fuzzy controller is able to keep track of the desired lighting level even after four failures are introduced in the experimental testbed as shown in Figure 4.8.
From Figure 4.8 is evident how the cross-illumination effect contributed by the controlled zones (i.e., zones 2, 3, 6, and 7) sets a fixed light level in the uncontrolled zones (i.e., zones 1, 4, 5, and 8). Moreover, zone 8 presents the case with the highest cross-illumination effect because its two neighboring zones (i.e., zones 6 and 7) have the highest applied voltage to each light bulb respectively right after the failure takes place as shown in Figure 4.9. Thus, this higher applied voltage also provides an explanation for the greater undershoot peak in the light level of zones 6 and 7 just when the failure happens. Obviously, smaller undershoot peaks are also presented in zones 2 and 3 due to the smaller change in the applied light bulb voltage in these zones respectively. We look at this behavior in more detail as shown in Figure 4.10.

Figure 4.8: Light levels at each zone for the direct PI fuzzy controller with no partition between zones under zones 1, 4, 5, and 8 light bulb failure.
When the light bulb failure occurs in zones 1, 4, 5 and 8 at $t = 30$ seconds, the failure is given by an immediate drop in the applied voltage level in the corresponding failing zones as shown in Figure 4.9. Following the light bulb failure, the remaining controlled zones react to the plant parameter change by immediately increasing their corresponding applied voltage level to maintain the uniform lighting.

![Graph showing applied voltage levels for zones 1 to 8](image)

Figure 4.9: Applied voltage level to the light bulbs at each zone for the direct PI fuzzy controller with no partition between zones under zones 1, 4, 5, and 8 light bulb failure.

Figure 4.10 shows a zoomed view of the light level in zones 2, 3, 6, and 7 in a $t = 27.5$ to $t = 32.5$ seconds window. A nice critically damped system behavior is observed as a response from the plant disturbance. Zone 7 has the highest undershoot peak at approximately 0.812 Volts below the reference voltage. Additionally, zones
2 and 3 have a settling time of approximately 0.625 seconds and zones 6 and 7 have a greater settling time of approximately 1.2 seconds. Since the same local direct PI fuzzy controller is implemented in each zone, it is logical to obtain an increase in the settling time for the zones with higher undershoot peaks.

Figure 4.10: Light level in controlled zones 2, 3, 6, and 7 for the direct PI fuzzy controller with no partition between zones under zones 1, 4, 5, and 8 light bulb failure.

Thirdly, a light bulb failure is created in zones 3, 4, 5, and 6 at $t = 30$ seconds. Once more, the direct PI fuzzy controller keeps track of the desired light level for this particular type of four zone simultaneous failure as illustrated in Figure 4.11. A typical reaction of the controller (i.e., undershoot at failure time) to reject the plant disturbance is observed in zones 1, 2, 7, and 8 respectively.
Figure 4.11: Light levels at each zone for the direct PI fuzzy controller with no partition between zones under zones 3, 4, 5, and 6 light bulb failure.

As presented in Figure 4.11, the uncontrolled zones 3, 4, 5 and 6 have a fixed light level after the failure occurs. In average, all the uncontrolled zones have the same light level between $t = 30$ seconds and $t = 40$ seconds and slightly different light levels between $t = 40$ seconds and $t = 60$ seconds because of the reduction of the desired reference level, and hence, a decrease in the cross-illumination effect due to the remaining controlled zones (i.e., zones 1, 2, 7, and 8). This is an interesting type of failure for two reasons: first, the introduced disturbance to the plant has a greater impact because the four failing zones are neighboring zones to each other, and secondly, two separated areas (i.e., zones 1-2 and zones 7-8) are created by assuming that they are far enough that there are no cross-illumination effects between them.
Figure 4.12: Applied voltage level to the light bulbs at each zone for the direct PI fuzzy controller with no partition between zones under zones 3, 4, 5, and 6 light bulb failure.

As given in Figure 4.12, the applied voltage levels drop to zero in zones 3, 4, 5, and 6 when the failure is activated. Once the failure is active, the zones 1, 2, 7, and 8 instantly increase the applied voltage level to the light bulbs so that they can compensate the failure. One more time, this fast voltage increase is reflected in the light level for the controlled zones by means of an undershoot as presented in Figure 4.11 for this failure case. Moreover, a closer look at these undershoots will provide additional information on how the direct PI fuzzy controller is able to reject this specific disturbance. Results prove that a similar system response is obtained for this type of failure as compared with the other failures already discussed in this chapter.
Figure 4.13: Light level in controlled zones 1, 2, 7, and 8 for the direct PI fuzzy controller with no partition between zones under zones 3, 4, 5, and 6 light bulb failure.

From Figure 4.13, a critically damped system response is illustrated in all controlled zones, similar to the ones already obtained in the previously discussed failures. Furthermore, zone 7 exhibits the highest undershoot peak at approximately 0.75 Volts below the desired reference level and a settling time of approximately 1.2 seconds. Zone 8 shows the second highest undershoot peak at approximately 0.5 Volts below the desired light reference level and a settling time of approximately 1 second. Zones 1 and 2 present the same undershoot peak at 0.375 Volts below the desired light reference level and a settling time of approximately 0.625 seconds.
4.2.2 Direct Adaptive PI Fuzzy Controller Results

In this subsection, we present the results for the direct adaptive PI fuzzy controller. Now that we have discussed the performance of the direct PI fuzzy controller under three distinct cases of multiple zones failures, we can proceed to compare the system response for both the direct nonadaptive and direct adaptive fuzzy controllers.

Figure 4.14: Light levels at each zone for the direct adaptive PI fuzzy controller with no partition between zones under zones 1, 3, and 4 light bulb failure.

First of all, a light bulb failure is generated in zones 1, 3, and 4 at $t = 30$ seconds. Notice in Figure 4.14 that the direct adaptive PI fuzzy controller maintains the uniform illumination across the remaining controlled zones 2, 5, 6, 7, and 8. Recalling the system light level performance for the direct PI fuzzy controller (i.e.,
Figure 4.5), we observe from Figure 4.14 that the adaptive control algorithm shows lower undershoot peaks as well as faster system response. In fact, the system response after the light bulb failure happens is similar to an under-damped (i.e., with oscillatory response) system behavior as illustrated in Figure 4.16.

![Figure 4.15: Applied voltage level to the light bulbs at each zone for the direct adaptive PI fuzzy controller with no partition between zones under zones 1, 3, and 4 light bulb failure.](image)

From Figure 4.15, we observe the applied voltage level given by the direct PI adaptive fuzzy controller. Clearly, a faster controller response is supported by the applied voltage in zones 2, 5, 6, 7, and 8. Thus, a lower applied voltage is achieved after a failure occurs which results in lower undershoot peaks as seen in Figure 4.16.
Besides the faster response, zone 7 shows the learning control algorithm ability to achieve sensor noise rejection that is ideal for this type of experimental testbed (i.e., low cost sensors and ADC data acquisition).

![Figure 4.16: Light level in controlled zones 2, 5, 6, 7, and 8 for the direct adaptive PI fuzzy controller with no partition between zones under zones 1, 3, and 4 light bulb failure.](image_url)

Implementation results shown in Figure 4.16 illustrate the direct adaptive PI fuzzy controller detailed response for the light bulb failure. Clearly, the learning control algorithm presents a faster response to the failure with an approximately settling time of 0.15 seconds for all the controlled zones (i.e., zones 2, 5, 6, 7, and 8). Notice the under-damped system response to the failure with the highest undershoot peak at
0.3125 Volts below the desired reference in zones 2 and 6 and the highest overshoot peak at 0.1875 Volts above the desired reference in zones 7 and 8.

Second of all, a light bulb failure is produced in zones 1, 4, 5, 8 at $t = 30$ seconds. For this case of four light bulb failures, the adaptive control algorithm continues to maintain the desired light reference level even after four light bulb failure are introduced in the testbed. As expected, there is a visible increase in the undershoot and overshoot peaks due to the additional failuring zone. However, the learning control algorithm reacts fast enough that is available to compensate the effect introduced by the plant disturbance.

Figure 4.17: Light levels in each zone for the direct adaptive PI fuzzy controller with no partition between zones under zones 1, 4, 5, and 8 light bulb failure.
Notice in Figure 4.17 that the uncontrolled zones exhibit the same fixed light level between \( t = 30 \) and \( t = 40 \) seconds, and then it is slightly different between \( t = 40 \) and \( t = 60 \) seconds because of the lower desired reference level, and hence, a reduced cross-illumination effect within neighboring zones.

From Figure 4.18 is evident that the applied voltage level by the adaptive control algorithm resembles the performance given by the nonadaptive control algorithm for all the controlled zones (i.e., zones 2, 3, 6, and 7) as compared to Figure 4.9 with undershoots and overshoots that does not affect the adaptive controller performance.
In Figure 4.19 above, we show the results of the adaptive PI fuzzy controller when the light bulb failure just occurs (i.e., a time window of 5 seconds). The highest undershoot peak is given by zone 7 at 0.562 Volts below the desired reference level which represents approximately a 30.78% peak reduction compared to the nonadaptive control algorithm (i.e., Figure 4.10). The highest overshoot peak is shown by zone 7 at 0.25 Volts above the desired reference level. Clearly, a faster settling of approximately 0.15 seconds is observed from all the controlled zones as compared to the nonadaptive control algorithm which constitute a 52% reduction in settling time. As a result, the learning control algorithm shows its robustness to a plant disturbance.

Figure 4.19: Light level in controlled zones 2, 3, 6, and 7 for the direct adaptive PI fuzzy controller with no partition between zones under zones 1, 4, 5, and 8 light bulb failure.
Third of all, a light bulb failure is initiated in zones 3, 4, 5, and 6 at \( t = 30 \) seconds. Implementation results given in Figure 4.20 illustrate how the learning control algorithm keeps track of the desired light level reference after introducing a light bulb failure in four neighboring zones (i.e., increased plant disturbance effect). Notice from Figure 4.20 that the fixed light level in the uncontrolled zones is comparable to the ones obtained for the nonadaptive fuzzy controller. The overall system response of both controllers is very similar until we take a closer look to the disturbance rejection as illustrated in Figure 4.22. The adaptive fuzzy controller exhibits very fast overshoots and undershoots whenever a change in the desired reference light level occurs in the plant but those peaks can be neglected due to very fast convergence.

Figure 4.20: Light levels at each zone for the direct adaptive PI fuzzy controller with no partition between zones under zones 3, 4, 5, and 6 light bulb failure.
As shown in Figure 4.21 above, the applied voltage level by the adaptive control algorithm is slightly below (few millivolts) as compared with the nonadaptive control algorithm for all the controlled zones (i.e., zones 1, 2, 7, and 8) given in Figure 4.12.

Implementation results of the direct adaptive PI fuzzy controller during the occurrence of the light bulb failure in zones 3, 4, 5, and 6 are presented in Figure 4.22. The highest undershoot peak is defined by zone 7 at 0.5 Volts below the desired reference level which constitutes approximately a 33.33% peak reduction compared to the nonadaptive control algorithm (i.e., Figure 4.13). The highest overshoot peak is given by zones 7 and 8 at approximately 0.187 Volts above the desired reference level.
Certainly, a faster settling time of approximately 0.15 seconds is observed from all the controlled zones as compared to the direct PI fuzzy controller that represents a 70% reduction in settling time compared with the faster settling time zones given in Figure 4.13 (i.e., zones 1 and 2).

![Diagram](image)

Figure 4.22: Light level in controlled zones 1, 2, 7, and 8 for the direct adaptive PI fuzzy controller with no partition between zones under zones 3, 4, 5, and 6 light bulb failure.

In the Section 3.2, we illustrated the ability of the direct fuzzy controller to achieve a nonlinear control surface. In order to show the learning capacity of the FMRLC, we present the nonlinear control surface achieved when a light bulb failure is introduced in zones 3, 4, 5, and 6 at $t = 30$ seconds. Figure 4.23 shows the nonlinear control
surface implemented by the direct adaptive PI fuzzy controller in zone 1 at the end of the simulation (i.e, \( t = 60 \) seconds). Notice from Figure 4.23 how the FMLRC algorithm introduced some significant changes in the control surface if we compare with the direct fuzzy nonlinear control surface given in Figure 3.13.

![Figure 4.23: Nonlinear control surface implemented by the direct PI fuzzy controller in zone 1 at \( t = 60 \) seconds under zones 3, 4, 5, and 6 light bulb failure.](image)

In general, all the implementation results discussed in the present chapter illustrate the effectiveness of the FMRLC algorithm for controlling an experimental testbed for smart lights under critical disturbance conditions. We are not only testing the plant
under several types of light bulb failures (i.e., single and multiple zones) but also maximizing the cross-illumination effect by means of no partitions between each zone in the entire testbed. As a consequence, a direct adaptive fuzzy controller provides a flexible alternative for the smart lights testbed by means of an heuristic design.

In conclusion, we have presented a significant amount of implementation results such that we could show the performance of both a direct PI fuzzy and a direct adaptive PI fuzzy controller. These experimental results have proven the capability of both controlling algorithms to react to a small and even a great plant disturbance so that the uniform illumination is maintained across the remaining controlled zones (i.e., the zones without light bulb failure). Furthermore, the adaptive PI fuzzy controller has shown an under-damped system response characterized with a lower undershoot peak and faster settling time as well as sensor noise rejection. In the next chapter, we look forward to introduce a more complex type of failure consisting of an on/off light bulb failure. This is not a common type of failure but it can be seen in rural locations (i.e., depending in renewable energies or commercial generators) or in developing countries with an electricity generation quota very close to the country’s electricity demand where our smart lights system approach can be used to save energy and maintain the quality of service.
Chapter 5: On/Off Fault Tolerance Test for Smart Lights

In this chapter, we carry out further exploration about the disturbance tolerance of both the direct PI fuzzy and direct adaptive PI fuzzy controllers by means of real-time implementations in the experimental testbed for smart lights. For this case, we introduced an on/off fault tolerance test in single zone and multiple zones scenarios with no partitions between zones. This consists of an on/off light bulb failure of five seconds long. We are interested in extensively studying the ability of the FMRLC to reject plant disturbances by learning about the new plant parameters. On the other hand, we are also concerned with comparing both adaptive and nonadaptive control algorithm at the same implementation setting. As in the previous chapter, we provide results for an on/off fault in two main cases: single zone and multiple zones on/off light bulb failure. Two main scenarios are presented for the multiple zones: one case for three and two cases for four zones on/off light bulb failure.

5.1 Single Zone On/Off Light Bulb Failure

In this section, we illustrate the implementation results obtained by generating an on/off light bulb failure in the testbed. This type of failure is performed by turning off the light bulb in a single zone for 5 seconds.
5.1.1 Direct PI Fuzzy Controller Results

The following set of results illustrate the system response for the direct PI fuzzy controller when an on/off light bulb failure is generated in the experimental testbed at \( t = 30 \) seconds in zone 1 (i.e., upper left corner). As shown in Figure 5.1, at \( t = 27.5 \) seconds the light bulb failure is initiated in zone 1 and then at \( t = 32.5 \) seconds the control is given back to the nonadaptive control algorithm. Notice in Figure 5.1 that a considerable overshoot peak at approximately 1.75 Volts above the desired light reference level is observed in zone 1 and is propagated to a lesser extent in the other remaining zones as we move away from zone 1. This phenomena presents the high coupling between the zones due to the maximized cross-illumination effect.

Figure 5.1: Light levels at each zone for the direct PI fuzzy controller with no partition between zones under zone 1 on/off light bulb failure.
In spite of the significant overshoot produced in zone 1 by the on/off failure, the direct PI fuzzy controller maintains the uniform illumination across the entire smart lights testbed as illustrated in Figure 5.1. This behavior again illustrates the disturbance rejection capability of a fuzzy controller.

From Figure 5.2 we observe the applied voltage level defined by the nonadaptive control algorithm. In zone 1 is clear that the on/off failure is introduced between 27.5 and 32.5 seconds of implementation time. Once the control is returned to the controller, the control output is saturated at 10 Volts for few seconds until it is decreased to a steady value. Zone 3 (followed by zone 2) exhibits the highest undershoot peak due to its close proximity to zone 1 (as defined in Figure 2.1).

Figure 5.2: Applied voltage level to the light bulbs at each zone for the direct PI fuzzy controller with no partition between zones under zone 1 on/off light bulb failure.
5.1.2 Direct Adaptive PI Fuzzy Controller Results

Implementation results from Figure 5.3 show the capability of the learning control algorithm to quickly adapt to a significant plant disturbance (i.e., the on/off failure). Comparing with the results given in the previous subsection, the adaptive algorithm shows a faster speed of convergence to the desired reference level and smaller overshoots and undershoots as compared to the nonadaptive algorithm. Also, good reference light level tracking is achieved by the adaptive control algorithm.

![Graphs showing light levels at each zone for the direct adaptive PI fuzzy controller with no partition between zones under zone 1 on/off light bulb failure.](image)

Figure 5.3: Light levels at each zone for the direct adaptive PI fuzzy controller with no partition between zones under zone 1 on/off light bulb failure.

In Figure 5.4, the applied voltage level defined by the adaptive control algorithm is illustrated. Notice from Figure 5.4 that the adaptive control algorithm does not
resemble the same control signal as given by the nonadaptive algorithm (i.e., Figure 5.2). Clearly, the learning process forces some significant changes in the control signal as given in zones 2 and 3 respectively which shuts down the light bulbs in these zones until zone 1 controller is able to start tracking the reference light level again.

Figure 5.4: Applied voltage level to the light bulbs at each zone for the direct adaptive PI fuzzy controller with no partition between zones under zone 1 on/off light bulb failure.

5.2 Multiple Zones On/Off Light Bulb Failure

In this section, we continue our exploration to further study the fault tolerance capability of both direct PI and direct adaptive PI fuzzy controllers. Real-time implementations introducing an on/off light bulb failure in multiple zones of the testbed
with no partition between zones are accomplished. We generate the on/off failure in the time window between \( t = 27.5 \) and \( t = 32.5 \) seconds. The same implementation scenarios discussed in the previous chapter are considered in this section. First, an on/off light bulb failure is generated in zones 1, 3 and 4 simultaneously which represents a three zone failure case. Then, two cases are presented for the on/off light bulb failure in four zones: a) zones 1, 4, 5, and 8, and b) zones 3, 4, 5, and 6. Again, we expect that the learning control algorithm outperforms the nonadaptive control in all the real-time implementations by means of recovering the desired light reference level in as many of the zones as possible.

### 5.2.1 Direct PI Fuzzy Controller Results

In this subsection, results for the direct PI fuzzy controller under multiple zones light bulb failure are presented. For all the given scenarios, the direct PI fuzzy controller is not able to maintain the uniform lighting in all the zones after the on/off failure is completed during the real-time implementation.

In the first place, an on/off light bulb failure is originated in zones 1, 3, and 4 between \( t = 27.5 \) and \( t = 32.5 \) seconds. From Figure 5.5, the nonadaptive control algorithm has the ability to recover the desired reference level in all the zones but zones 2 and 6. For zone 2 is evident that this particular zone will be the most affected by the system reaction to the on/off failure because it is surrounded by all failing zones, and hence, the direct fuzzy controller is not fast enough to respond before a change in the desired reference input. Similarly, the zone 6 light bulb remains turned off until the controller starts increasing the applied voltage but it is not able to reach the reference as fast as the neighboring zones (i.e., zones 3, 4, 5, 7, and 8).
Figure 5.5: Light levels at each zone for the direct PI fuzzy controller with no partition between zones under zones 1, 3, and 4 on/off light bulb failure.

From Figure 5.6, we can observe the applied voltage level defined by the nonadaptive control algorithm. Notice in Figure 5.6 that in zones 1, 3, and 4 the light bulbs are turned off between $t = 27.5$ and $t = 32.5$ seconds (i.e., for 5 seconds). After the failure is over, the direct PI fuzzy controllers applies a control signal of 10 Volts for few seconds until is able to get back to the desired light reference level. On the other hand, all the remaining zones experience a shut down of its corresponding light bulb until the controller is able to track the reference light level in zones 5, 7, and 8 before $t = 40$ seconds (i.e., when the reference input is changed). Also, the abrupt changes in the applied voltage produces the significant overshoot seen in Figure 5.5.
In the second place, an on/off light bulb failure is initiated in zones 1, 4, 5, and 8 between $t = 27.5$ and $t = 32.5$ seconds. For this scenario shown in Figure 5.7, the direct PI fuzzy controller fails to keep track of the desired reference level in zones 6 and 7 not only before but also after the reference input is lowered to 5 Volts at $t = 40$ seconds. Be aware that in zones 6 and 7 there is a really small (i.e., few millivolts) of steady state error with the final desired reference input after $t = 40$ seconds. This steady state error is explained by the fact that the light bulb is turned off in zone 7 after $t = 32.5$ seconds and is also turned off between $t = 32.5$ and $t = 50$ seconds in zone 6.
Implementation results given by Figure 5.8 shows the applied voltage level by the direct PI fuzzy controller for an on/off failure in zones 1, 4, 5 and 8. Clearly, the light bulbs remain turned off in the failing zones between $t = 27.5$ and $t = 32.5$ seconds. As compared with the previous three zones failure, the effect of the on/off failure is more distributed throughout the entire testbed since neither of the failing zones is either a horizontal or vertical neighbor of another failing zone. As a matter of fact, we observe a better response of the PI direct fuzzy controller compared to the previous three zones failure case but a turned off light bulb as in zone 7 is not desirable because the controller is seen to have no capacity of either recovering the tracking performance of the desired reference input or keeping the light bulb on.
Figure 5.8: Applied voltage level to the light bulbs at each zone for the direct PI fuzzy controller with no partition between zones under zones 1, 4, 5 and 8 on/off light bulb failure.

In the third place, an on/off light bulb failure is produced in zones 3, 4, 5, and 6 between \( t = 27.5 \) and \( t = 32.5 \) seconds. One more time, results prove the poor system performance with the direct PI fuzzy controller as shown in Figure 5.9. Clearly, the nonadaptive control algorithm fails to maintain the desired reference light level in zones 2, 7 and 8. Actually, only zone 1 (i.e., the only remaining not failing zone) is able to track the reference input only few seconds before the reference input is lowered to 5 Volts. Also, notice in Figure 5.9 that zone 7 exhibits a steady state error and zones 2, 7, and 8 present an undershoot after \( t = 40 \) seconds (i.e., reference input of 5 Volts).
As illustrated in Figure 5.10, the applied voltage levels drop to zero in zones 3, 4, 5, and 6 when the on/off failure is activated. Once the failure has taken place, the zone 3, 4, 5, and 6 controllers quickly react to apply the necessary voltage to the light bulbs such that the reference input is tracked. This fast response in the perturbed zones is reflected into the not failing zones (i.e., zones 1, 2, 7 and 8) by means of the light bulb being turned off by its corresponding direct PI fuzzy controller. Evidently, the undershoots present in zones 2, 7 and 8 are connected to the delay of the nonadaptive controller to react to the change in the reference input and in zone 7 the steady state error is connected to the fact that the cross-illumination effect has eliminated any possible controller reaction to the plant disturbance.
Figure 5.10: Applied voltage level to the light bulbs at each zone for the direct PI fuzzy controller with no partition between zones under zones 3, 4, 5 and 6 on/off light bulb failure.

5.2.2 Direct Adaptive PI Fuzzy Controller Results

In this subsection, results for the direct adaptive PI fuzzy controller under a multiple zone on/off light bulb failure are discussed and compared to the direct PI fuzzy controller results. For all the given cases, the learning control algorithm is capable to improve the system performance compared to the nonadaptive control algorithm. Nevertheless, the light unbalance introduced by the on/off failure is expected to drive the adaptive control algorithm close to failure in at least one zone but keeping all light bulbs turned on for the last desired reference input.
Firstly, an on/off light bulb failure is created in zones 1, 3, and 4 between $t = 27.5$ and $t = 32.5$ seconds. From Figure 5.11 is clear that the direct adaptive PI fuzzy controller maintains the uniform illumination across all the zones after the on/off failure is finished at $t = 32.5$ seconds and the control algorithm is able to track the reference in the failing zones. Also, notice in Figure 5.11 that it takes approximately 2 seconds for the adaptive control algorithm to recover the control in zones 1, 3, and 4 which is a similar performance as compared to the nonadaptive control algorithm. Once the direct adaptive PI fuzzy controller tracks back the desired reference light level in zones 1, 3, and 4, then it is capable to recover the reference tracking in zones 2, 5, 6, 7, and 8 in an average time of 2 seconds.

Figure 5.11: Light levels at each zone for the direct adaptive PI fuzzy controller with no partition between zones under zones 1, 3, and 4 on/off light bulb failure.
Recall from the previous subsection that the direct PI fuzzy controller failed to track the desired reference input in zones 2 and 6 between $t = 35$ and $t = 40$ seconds (i.e., Figure 5.5). As already mentioned, this system performance is related to the fact that the nonadaptive algorithm is not able to turn on the light bulbs in zone 2 and 6 respectively. Figure 5.12 illustrates the applied voltage defined by the direct adaptive PI fuzzy controller. Evidently, we can point out from Figure 5.12 that the adaptive control algorithm turns on the light bulbs in zones 2, 5, 6, 7, and 8 at different instances (few milliseconds apart) but this proves the capacity of the adaptive controller to reject the plant disturbances and to retain successful operation for this particular real-time implementation scenario.

Figure 5.12: Applied voltage level to the light bulbs at each zone for the direct adaptive PI fuzzy controller with no partition between zones under zones 1, 3, and 4 on/off light bulb failure.
Secondly, an on/off light bulb failure is generated in zones 1, 4, 5, and 8 between $t = 27.5$ and $t = 32.5$ seconds. This implementation scenario represents the first four light bulbs failure case as illustrated in Figure 5.13. Comparing to the results given for the direct PI fuzzy controller (i.e., Figure 5.7), we observe that the adaptive control algorithm performs in a similar way between $t = 32.5$ and $t = 40$ seconds. This means that the adaptive control algorithm also fails to track the desired reference light level for few millivolts in both zones 6 and 7. On the other hand, after $t = 40$ seconds the learning control algorithm is able to eliminate the steady state error which is presented in the nonadaptive control algorithm results. Again, this provides another situation where the adaptation overcomes a significant problem.

Figure 5.13: Light levels at each zone for the direct adaptive PI fuzzy controller with no partition between zones under zones 1, 4, 5 and 8 on/off light bulb failure.
In general, the already mentioned situation where the adaptive control algorithm is capable to turn back on the light bulbs in more than one zone (i.e., zones 6 and 7) is shown in Figure 5.14. Notice from Figure 5.14 that it takes approximately 6 seconds for the adaptive control algorithm to turn on the light bulb in zone 7 after the reference input is lowered to 5 Volts. This means that several updates of the fuzzy rule-base has been done by the learning mechanism until a positive voltage is applied to the light bulb in zone 7 and the uniform illumination is recovered across the entire experiment testbed. Once again, this behavior shows the robustness and adaptation capability of the adaptive control algorithm for a smart lighting control problem.

Figure 5.14: Applied voltage level to the light bulbs at each zone for the direct adaptive PI fuzzy controller with no partition between zones under zones 1, 4, 5 and 8 on/off light bulb failure.
Thirdly, an on/off light bulb failure is produced in zones 3, 4, 5, and 6 between $t = 27.5$ and $t = 32.5$ seconds. This implementation scenario represents the case where the plant disturbance due to the on/off failure is maximized. Recalling from previous subsection, the direct PI fuzzy controller failed to recover the tracking in zones 2, 7, and 8 between $t = 32.5$ and $t = 40$ seconds (i.e., Figure 5.9). From Figure 5.15 is apparent that the direct adaptive fuzzy controller is able to recover the tracking in zones 2 and 6 but there is still a small steady state error (few millivolts) in zone 7. As shown in Figure 5.16, the light bulb in zone 7 remains turned off between $t = 32.5$ and $t = 40$ seconds which justify the presence of the steady state error. The given voltage in zone 7 is defined by the cross-illumination effect in the testbed.

![Figure 5.15: Light levels at each zone for the direct adaptive PI fuzzy controller with no partition between zones under zones 3, 4, 5 and 6 on/off light bulb failure.](image-url)
In spite of the incapacity of the direct adaptive PI fuzzy controller to regulate zone 7 between $t = 27.5$ and $t = 32.5$ seconds, when the reference input is lowered to 5 Volts at $t = 40$ seconds the adaptive control algorithm is able to track the desired reference light level in all the zones, eliminating the undershoot in zones 2, 7, and 8 and the steady state error in zone 7. The applied voltage level for this four zone on/off failure is shown in Figure 5.16. Notice from Figure 5.16 that the adaptive algorithm has the competence to maintain all the light bulbs turn on immediately after the last reference input is introduced at $t = 40$ seconds. One more time, this is a clear evidence of the learning characteristics used to reject plant disturbances by the FMRLC implemented on this research.

Figure 5.16: Applied voltage level to the light bulbs at each zone for the direct adaptive PI fuzzy controller with no partition between zones under zones 3, 4, 5 and 6 on/off light bulb failure.
In the subsection 4.2.2, we illustrated the ability of the FMRLC to adapt the nonlinear control surface during a multiple zone light bulb failure. Again, we want to emphasize the learning capacity of the FMRLC by presenting the nonlinear control surface achieved when an on/off light bulb failure is introduced in zones 3, 4, 5, and 6 between $t = 27.5$ and $t = 32.5$ seconds. Notice from Figure 5.17 how the FMRLC quickly adapts the nonlinear control surface compared to the multiple zone light bulb failure case given in Figure 4.23.

Figure 5.17: Nonlinear control surface implemented by the direct adaptive PI fuzzy controller in zone 1 at $t = 35$ seconds under an on/off light bulb failure in zones 3, 4, 5 and 6.
Moreover, notice from Figure 5.18 how the FMRLC kept adapting the control surface by the end of the simulation (i.e., \( t = 60 \) seconds). This clearly proves the benefits of having an adaptive control algorithm in order to obtain a satisfactory performance under several scenarios.

![Nonlinear control surface implemented by the direct adaptive PI fuzzy controller in zone 1 at \( t = 60 \) seconds under an on/off light bulb failure in zones 3, 4, 5 and 6.](image)

Figure 5.18: Nonlinear control surface implemented by the direct adaptive PI fuzzy controller in zone 1 at \( t = 60 \) seconds under an on/off light bulb failure in zones 3, 4, 5 and 6.
As a concluding remark, we demonstrated the capacity of the learning control algorithm to outperform the nonadaptive control algorithm in terms of the smart lights system response by means of conducting online adaptation of the fuzzy rule-base (i.e., modifying the fuzzy knowledge-base). Recall that the learning mechanism only requires as inputs a reference model and the plant output which are both accessible in our experimental testbed for smart lighting. In general, the FMRLC clearly shows its capability to maintain the uniform illumination across the entire testbed, rejection of sensor noise and uncertainties under maximized cross-illumination effects (i.e., no partitions within the zones) and several combinations of on/off light bulb failures.
Chapter 6: Sensor Failure Test for Smart Lights

In this chapter we present results from real-time experiments when there are sensor failure in the smart lights testbed. We compare the performance for both the PI fuzzy and direct adaptive PI fuzzy controllers for several sensor failure conditions. The sensor failure test provides a case of study where the illumination is maximized (i.e., light bulb is turned on at maximum applied voltage) in the zone where the failure occurs and an over-illumination unbalance is introduced to the testbed via the cross-illumination between zones. This test is performed for half-height partitions between zones in order to attenuate the cross-illumination effect so that the impact of the sensor failure can be analyzed and compensated by the fuzzy controller. Implementation results for sensor failures are presented in two main cases: single zone and multiple zones sensor failure. For multiple zones sensor failure, two cases for two zone failures are presented in this chapter.

6.1 Single Zone Sensor Failure

In this section, we discuss the results achieved by introducing a sensor failure by assuming that the sensor measurement is always zero (i.e., zero illumination) at a single zone in the testbed with half-height partition between zones. We study this type of failure in order to analyze the ability of the nonadaptive and adaptive fuzzy
controllers to compensate a sensor failure. For this single zone sensor failure, the two fuzzy algorithms can regulate the desired light levels quickly after the sensor failure occurs in the remaining controlled zones.

6.1.1 Direct PI Fuzzy Controller Results

When a sensor failure is generated in the experimental testbed at $t = 30$ seconds in zone 1 (i.e., the upper left corner), the direct PI fuzzy controller is able to maintain the uniform lighting in zones 4, 5, 6, 7 and 8, and is unable to track the voltage reference input in zones 2 and 3 after $t = 40$ seconds (i.e., when the voltage reference level is lowered from 7V to 5V). See Figure 6.1.

![Figure 6.1: Light levels at each zone for the PI fuzzy controller with half partitions between zones under zone 1 sensor failure.](image)
From Figure 6.1, we can see that the cross-illumination effect is maximized in the neighboring zones (i.e., zone 2 and 3) and that the fuzzy controller is unable to track the lower reference input. Clearly, overshoots can be seen in zones 2 and 3, and very small overshoots are presented in zones 4, 5, and 7 as the failure propagates through the testbed. The applied voltages to the light bulbs on each zone for this single zone failure are illustrated in Figure 6.2. We can observe from Figure 6.2 how the fuzzy controller reacts to the zone sensor failure at \( t = 30 \) seconds. In zone 1 the control signal clamps at 10 Volts due to the sensor failure, in zones 2 and 3 the control signal goes to 0 Volts (i.e., light bulb is turned “off ”) after \( t = 40 \) seconds, and in the remaining zones the control signal adjusts to the perturbation.

![Figure 6.2: Applied voltage level to the light bulbs at each zone for the PI fuzzy controller with half partition between zones under zone 1 sensor failure.](image-url)
6.1.2 Direct Adaptive PI Fuzzy Controller Results

In a similar way as in the previous subsection, a sensor failure is produced in zone 1 at $t = 30$ seconds. The direct adaptive PI fuzzy controller is able to maintain uniform lighting in zones 4, 5, 6, 7 and 8, and is unable to maintain the uniform lighting in zones 3 and 4 after $t = 40$ seconds as shown in Figure 6.3.

Notice from Figure 6.3 that the neighboring zones (i.e., zones 2 and 3) present smaller overshoots after the sensor failure occurs at $t = 30$ seconds due to the “learning mechanism” of the adaptive fuzzy controller compared to the fuzzy controller (i.e., see Figure 6.1). Also, the adaptive fuzzy controller eliminates the overshoots in zones 4,
5, and 7 that are present in Figure 6.1 for the nonadaptive fuzzy controller. Moreover, the adaptive fuzzy controller is tuned in the same way as for the simple and on/off light bulb failures presented in Chapters 4 and 5 respectively in order to be consistent with our adaptive controller design.

The applied voltage levels to the light bulbs on each zone for this single zone sensor failure are illustrated in Figure 6.4. By comparing with the applied voltage given by the fuzzy controller (i.e., see Figure 6.2), a significant adaptation is only presented in zones 2 and 3 (i.e., immediate neighboring zones of zone 1) and the remaining controlled zones present smaller adaptations as we move further away from zone 1.

![Figure 6.4: Applied voltage level to the light bulbs at each zone for the direct adaptive PI fuzzy controller with half partition between zones under zone 1 sensor failure.](image-url)
6.2 Two-Zone Sensor Failure

In this section, we show the implementation results obtained by generating a sensor failure in two zones of the testbed with half-height partition between zones. This sensor failure is introduced by assuming that the sensor measurement is always zero (i.e., zero illumination) in two zones for two selected cases. First, the case when zones 1 and 8 simultaneously fail is presented. Then, the case when zones 3 and 6 simultaneously fail is given. For comparison, the detailed response of the two fuzzy algorithms when the sensor failure occurs is illustrated. For this two-zone sensor failure, the learning control algorithm is expected to show uniform lighting in the remaining controlled zones as well as faster convergence to the desired voltage reference input under the disturbance given by the sensor failure.

6.2.1 PI Fuzzy Controller Results

In this section, we provide the results for the PI fuzzy controller for two cases of two-zone sensor failure. Depending on the case studied, a sensor failure is generated in the experimental testbed at \( t = 30 \) seconds for two separated zones and for two neighboring zones. For the two cases, the PI fuzzy controller is able to track the voltage reference input until a change in the reference input is introduced.

Firstly, a sensor failure is generated in zones 1 and 8 at \( t = 30 \) seconds. For this sensor failure case, the PI fuzzy controller is able to track the desired reference lighting level in zones 2, 3, 4, 5, 6, and 7 until \( t = 40 \) seconds as shown in Figure 6.5. Once the reference input is lowered to 5 Volts, the fuzzy controller cannot track the voltage reference input in zones 2 and 3 (i.e., neighboring zones of zone 1), and zones 6 and 7 (i.e., neighboring zones of zone 8) due to the maximized cross-illumination.
From Figure 6.5, we can see how the voltage level in the uncontrolled zones (i.e., zones 1 and 8) drops to a fixed value which is given by the bias voltage defined in the sensor calibration described in Chapter 2. Notice from Figure 6.5 how the neighboring zones for zone 1 (i.e., zones 2 and 3) and 8 (i.e., zones 6 and 7) show higher overshoot right after the sensor failure happens and the incapacity of tracking the reference input when it is lowered from 7 Volts to 5 Volts. Clearly, the PI fuzzy controller is able to track the reference input in zones 4 and 5 and we can also observe small overshoots due to the propagation of the disturbance through the entire testbed. The size of the overshoot is connected to how close the controlled zone is to a zone where a zone failure just occurred due to the coupling between zones.

Figure 6.5: Light levels at each zone for the PI fuzzy controller with half partition between zones under zones 1 and 8 sensor failure.
In Figure 6.6, the applied voltages to the light bulbs on each zone for the sensor failure in zones 1 and 8 are presented. Figure 6.6 shows the occurrence of the sensor failure in zones 1 and 8 by the clamping at 10 Volts of the applied voltage by the fuzzy controller. The overshoots in the controlled zones given in Figure 6.5 are connected with a decrease in the applied voltage of those controlled zones to compensate the illumination unbalance introduced by the two failures. Clearly, we also observe how the light bulbs in zones 2, 3, 6, and 7 are turned off due to the inability of the fuzzy controller to track the reference input after $t = 40$ seconds.

Figure 6.6: Applied voltage level to the light bulbs at each zone for the PI fuzzy controller with half partition between zones under zones 1 and 8 sensor failure.
A zoomed view of the light level in the controlled zones (i.e., zones 2, 3, 4, 5, 6 and 7) during the occurrence of the sensor failure is shown in Figure 6.7. In the worst case, an overshoot with a peak of approximately 0.30 Volts is observed in zones 3, 6, and 7, zone 2 shows a peak of approximately 0.25 Volts of overshoot, and zones 4 and 5 present a peak of less than 0.25 Volts of overshoot. It takes approximately 1 second for the fuzzy controller to overcome the overshoot in zones 3, 6, and 7. For zone 2, it takes approximately 0.85 seconds for the controller to recover from the overshoot, and for zones 4 and 5, it takes approximately 0.5 seconds to return from the overshoot respectively. As expected, zones 2 and 3, and zones 6 and 7 show a similar transient response to the sensor failure because they are neighboring zones of zones 1 and 8 respectively.

Figure 6.7: Light level in controlled zones 2, 3, 4, 5, 6, and 7 for the PI fuzzy controller with half partition between zones under zones 1 and 8 sensor failure.
Secondly, a sensor failure is generated in zones 3 and 6 at $t = 30$ seconds. For this sensor failure case, the PI fuzzy controller is able to track the desired reference lighting level in zones 1, 2, 4, 5, 7, and 8 until $t = 40$ seconds as shown in Figure 6.8. Once the reference input is lowered to 5 Volts, the fuzzy controller cannot track the voltage reference input in all the remaining controlled zones because each controlled zone is a neighboring zone of a failing zone. From Figure 6.8 is evident how the voltage level in the uncontrolled zones (i.e., zones 1 and 3) drops to a fixed constant value. Evidently, each failing zone has more than two neighboring zones which support the fact that the fuzzy controller is unable to keep the uniform lighting after $t = 40$ seconds.

Figure 6.8: Light levels at each zone for the PI fuzzy controller with half partition between zones under zones 3 and 6 sensor failure.
When the sensor failure occurs in zones 3 and 6 at $t = 30$ seconds, the failure is given by a clamp at 10 Volts of the applied voltage in the corresponding failing zones as illustrated in Figure 6.9. Following the sensor failures, the remaining controlled zones react to the sensor failure by immediately decreasing their corresponding applied voltage level in order to maintain uniform lighting. Thus, an overshoot is generated in each controlled zone (as given in Figure 6.8) and it is related to the decrease in the applied voltage shown in Figure 6.9. After $t = 40$ seconds, all the light bulbs in the remaining controlled zones (i.e., zones 1, 2, 4, 5, 7, and 8) are turned off and the fuzzy controller is unable to keep track of the desired voltage reference input.

Figure 6.9: Applied voltage level to the light bulbs at each zone for the PI fuzzy controller with half partition between zones under zones 3 and 6 sensor failure.
Figure 6.10 shows a detailed view of the light level in the controlled zones (i.e., zones 1, 2, 4, 5, 7 and 8) in a $t = 27.5$ to $t = 32.5$ seconds window. Zone 5 has the highest overshoot peak at approximately 0.40 Volts above the reference voltage and zone 2 has the smallest overshoot peak at approximately 0.13 Volts above the reference voltage. In addition, zone 5 has the longest settling time of approximately 1.2 seconds and zone 2 has the shortest settling time of approximately 0.5 seconds. Clearly, zone 5 is the most affected zone by the increased cross-illumination effect that was generated by the sensor failure in zones 2 and 3, and following in decreasing order zones 8, 4, 1, 7 and 2. Since the zones in the testbed are not symmetrically distributed (see Figure 2.1), the behavior given in Figure 6.10 is expected.

Figure 6.10: Light level in controlled zones 1, 2, 4, 5, 7, and 8 for the direct PI fuzzy controller with half partition between zones under zones 3 and 6 sensor failure.
6.2.2 Direct Adaptive PI Fuzzy Controller Results

In this subsection, we illustrate the results for the direct adaptive PI fuzzy controller for two-zone sensor failures. For the PI fuzzy controller, we discussed the performance for two distinct cases of two-zone failures so we can now compare it with the system response of the adaptive fuzzy controller.

First, a sensor failure is introduced in zones 1 and 8 at \( t = 30 \) seconds. From Figure 6.11, we notice that the direct adaptive PI fuzzy controller maintains the uniform illumination in the controlled zones 4 and 5, and for the controlled zones 2, 3, 6 and 7 until \( t = 40 \) seconds (i.e., before a change in the reference input).

Figure 6.11: Light levels at each zone for the direct adaptive PI fuzzy controller with half partition between zones under zones 1 and 8 sensor failure.
Comparing with the system light level performance for the PI fuzzy controller (i.e., see Figure 6.5), we can see from Figure 6.11 that the adaptive control algorithm quickly rejects the disturbance by attenuating the overshoot peaks and achieving faster convergence to the desired voltage reference input. Additionally, Figure 6.13 presents the same fast under-damped transient response given by the adaptive fuzzy algorithm for the simple and on/off light bulb failures presented in Chapters 4 and 5 respectively in a $t = 27.5$ to $t = 32.5$ seconds window.

![Graphs showing applied voltage level to light bulbs at each zone](image)

**Figure 6.12**: Applied voltage level to the light bulbs at each zone for the direct adaptive PI fuzzy controller with half partition between zones under zones 1 and 8 sensor failure.

In Figure 6.12 we see the applied voltage level defined by the direct PI adaptive fuzzy controller for sensor failures in zones 1 and 8. The smaller overshoot peaks
presented in zones 2, 3, 4, 5, 6 and 7 in Figure 6.11 are supported by the faster controller response at $t = 30$ seconds (i.e., right after the sensor failure occurred) as illustrated in Figure 6.12. Hence, the learning control algorithm shows the ability to quickly adjust the applied voltage to achieve a better response.

![Figure 6.13: Light level in controlled zones 2, 3, 4, 5, 6, and 7 for the direct adaptive PI fuzzy controller with half partition between zones under zones 1 and 8 sensor failure.](image)

A detailed view of the light level in the controlled zones (i.e., zones 2, 3, 4, 5, 6 and 7) in a $t = 27.5$ to $t = 32.5$ seconds window is shown in Figure 6.13. Clearly, the adaptive control algorithm provides a faster response to the sensor failure with an approximate settling time of 0.15 seconds for all the controlled zones that is a 85% reduction of the settling time compared to the worst case (i.e., $t_s = 1$ second) for the
nonadaptive controller and the highest overshoot peak at approximately 0.20 Volts in zones 3, 6 and 7 that is a 33.33% peak reduction compared to the worst case of the nonadaptive controller in Figure 6.7. The highest undershoot peak at approximately 0.10 Volts below the desired reference input is presented in zones 3, 6 and 7.

Second, a sensor failure is generated in zones 3 and 6 at \( t = 30 \) seconds. For this case, where the failing zones are also neighboring zones, the adaptive controller is able to maintain the uniform illumination in the tested in the controlled zones 1, 2, 4, 5, 7 and 8 until \( t = 40 \) seconds as presented in Figure 6.14. Evidently, there is a clear decrease in the overshoot and undershoot peaks for those zones closer to the failing zones compared with the nonadaptive controller (i.e., see Figure 6.8).

Figure 6.14: Light levels at each zone for the direct adaptive PI fuzzy controller with half partition between zones under zones 3 and 6 sensor failure.
Notice from Figure 6.14 that the uncontrolled zones present the same fixed light level between $t = 30$ and $t = 60$ seconds due to the sensor failure. After $t = 40$ seconds, all the remaining controlled zones exhibit different fixed light levels above the desired voltage reference input which are defined by the cross-illumination effect created by the maximum illumination (i.e., applied voltage clamp at 10 Volts) produced in the failing zones 3 and 6 as illustrated in Figure 6.15. The applied voltage level given by the adaptive controller in Figure 6.15 is similar to the performance described by the nonadaptive controller in Figure 6.9 but again characterized by a faster rejection of the sensor failure.

![Graphs of Zones 1 to 8 showing applied voltage levels over time](image)

**Figure 6.15:** Applied voltage level to the light bulbs at each zone for the direct adaptive PI fuzzy controller with half partition between zones under zones 3 and 6 sensor failure.
In Figure 6.16, a zoomed view of the performance of the adaptive PI fuzzy controller when the sensor failure just happened (i.e., a time window of 5 seconds) is presented. The highest overshoot peak is given by zone 5 at approximately 0.25 Volts above the desired voltage reference level which is a 37.5% peak reduction compared to the nonadaptive controller (i.e., Figure 6.10). The highest undershoot peak is shown by zone 4 at 0.15 Volts below the desired voltage reference level. In addition, a faster settling time of approximately 0.15 seconds is achieved in all the controlled zones which represents a 87.5% reduction in settling time compared with the worst case (i.e., $t_s = 1.2$ seconds) for the nonadaptive controller.

Figure 6.16: Light level in controlled zones 1, 2, 4, 5, 7, and 8 for the direct adaptive PI fuzzy controller with half partition between zones under zones 3 and 6 sensor failure.
To conclude, we have presented the ability of the learning control algorithm to adapt to the disturbance introduced by a sensor failure in one and two zones in the smart lights experimental testbed. We have to point out that the inability of the adaptive fuzzy controller to track a lower voltage reference input does not represent poor performance of the controller. The maximized cross-illumination between the zones extremely limits the capacity of a decentralized control approach (i.e., without communication between zones) to overcome this type of failure. Thus, these results just provide the motivation for the necessity to include a sensor failure mechanism within the controller that will automatically dim or shut down the light bulb of a zone once the failure is detected.
Chapter 7: Conclusions and Future Work

The main objective of this thesis was to implement a direct nonadaptive and a direct adaptive fuzzy controller for our smart light experimental testbed. Furthermore, in this thesis we have presented the fundamental theory behind fuzzy control and a design methodology for implementing this type of controller. Compared with the previous research conducted in the smart light testbed [12, 13, 14], we have been able to demonstrate the potential of fuzzy control over other conventional methods for our smart lighting system in achieving uniform illumination across the testbed despite plant variations (i.e., changes in the partition settings), single zone and multiple zones light bulb failures, and single zone and two-zone sensor failures.

Based on the implementation results presented in this thesis, we have proven the benefits of the fuzzy model reference learning controller algorithm for controlling our smart light experimental testbed. The FMRLC should be considered as a smart light control algorithm for the following reasons: a detailed mathematical model of the plant to be controlled is not necessary, the learning mechanism provides an automatic way to adjust the fuzzy control rule-base while behaving as a desired reference model, and the continuous adaption mechanism provides additional rejection to parameters variations and/or disturbances which results in a more “robust” controller compare to the nonadaptive algorithm.
The implementation results from a simple light bulb failure showed the ability of the FMRLC to adapt to a plant disturbance while maintaining the uniform illumination across the testbed. Moreover, the FMRLC provided a faster transient response with lower undershoot peaks and minimized the effects of the sensor noise. From the on/off light bulb failure case, the online adaption of the fuzzy rule-base provided a great benefit under the effect of a significant plant disturbance (i.e., multiple on/off light bulb failures), and also, when the cross-illumination effects have been maximized. For the case of a sensor failure, the FMRLC showed the ability to adapt to this type of failure under certain conditions where the controlled zones were not directly (i.e., not being a neighbor of a failing zone) affected by the maximized cross-illumination effect introduced in the testbed. From these cases, we have analyzed the effects of disturbances at both the actuator and sensor level.

Smart light control systems is an area with relatively little research. Nevertheless, it is a promising research area that will have a significant impact in the way that we use lighting in our buildings in the near future. In this thesis, we mainly focus on the implementation of an intelligent control algorithm for a low cost experimental testbed. Based on other smart light research works [8, 9, 10, 11], we suggest that the experimental testbed should be expanded in size as well as incorporate commercial photosensors, and intelligent and energy-efficient luminaries (i.e., LED luminaries). Additionally, an office size experimental testbed will allow for the inclusion of a new sensor such as an occupancy sensor. Occupancy sensing will allow the smart light controller to switch off the lights in areas where they are not being used, and hence, a detailed energy savings study could be performed.
Bibliography


