EXTENSIONAL FUZZY LOGIC CONTROLLERS FOR UNCERTAIN SYSTEMS

BY

LAI JUNWEI
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

NATIONAL UNIVERSITY OF SINGAPORE
2007
Acknowledgement

First of all, I would like to thank to my project supervisor Dr. Tan Woei Wan for her great guidance and assistance along the difficult research road. Her trust and patience are truly appreciated when I encountered difficulties in my research. Her insight into different aspects of control engineering and fuzzy logic theories has helped to solve many problems and fine-tune many important ideas. I have also learned a lot from her since joining the university.

I would also like to express my sincere and heartfelt gratitude to my wife and my son Elwin. During the long time of thesis revision, I may not be able to perform my husband role very well to take care of my wife when she was pregnant. She always gives me a good environment to concentrate on my thesis writing, even in the first month after my baby was born. I am forever grateful to my loving parents, I have to thank to their consistent support and endless love. Thanks for their assistance in taking care my wife and my son, I can settle down to concentrate on my research and thesis writing during the recent year. It is my immense pleasure to dedicate this small accomplishment to my family.

Last but definitely not least, I would like to take this opportunity to express my gratitude to my colleagues for their camaraderie and friendship. Over the four years, we have shared together and this is always one of the most enjoyable and impressionable period in my life.
Contents

List of Figures x

List of Tables xi

Summary xii

1 Introduction 1

1.1 Uncertainty in the Real World ................................. 1

1.2 Historical Review on Fuzzy Control ............................... 2

1.3 Extension to Type-1 Fuzzy Logic Theory .......................... 4

1.3.1 Non-singleton type-1 fuzzy logic systems ..................... 5

1.3.2 Type-2 fuzzy logic systems .................................. 5

1.3.3 Recent research in type-2 fuzzy controllers ................... 7

1.4 Aims and Scope of the Work .................................... 9

1.5 Organization of the Thesis .................................... 11

2 Theories on Extensional Fuzzy Logic 13

2.1 Singleton Type-1 Fuzzy Logic Systems ............................ 13

2.2 Realization of PID Control Using Type-1 FLSs .................... 17

2.3 Non-singleton Type-1 Fuzzy Logic Systems ........................ 20

2.4 Type-2 Fuzzy Logic Theories .................................. 23

2.4.1 Type-2 membership functions ................................. 24

2.4.2 Embedded type-2 and type-1 sets ............................. 27
2.4.3 Operations of type-2 fuzzy sets ........................................... 30
2.4.4 Centroid of type-2 fuzzy sets ............................................. 31
2.4.5 Properties of the centroid for an interval type-2 set ............. 33
2.4.6 Type reduction ................................................................. 37
2.4.7 Interval type-2 fuzzy logic systems ..................................... 38

3  Non-singleton Type-1 Fuzzy Controller for Noise Rejection 42
3.1 Properties of Symmetric Triangular Non-singleton Fuzzifier .... 43
  3.1.1 Case I: Support of X partially overlaps the support of S1 . . 45
  3.1.2 Case II: Support of X is a subset of the support of S1 . . . 48
  3.1.3 Case III: Support of S1 is a subset of X ....................... 49
3.2 Non-singleton Type-1 PI Fuzzy Controller .............................. 50
  3.2.1 Structure of non-singleton PI controller .......................... 50
  3.2.2 Structure of inference engine ......................................... 51
  3.2.3 Characteristics of fuzzy PI controller using symmetric non-
      singleton fuzzifier ......................................................... 52
3.3 Non-symmetric non-singleton Fuzzifier ................................ 58
3.4 Simulation Results .............................................................. 60
  3.4.1 pH process in CSTR .................................................... 60
  3.4.2 Performance of proposed controller ............................... 63
3.5 Case Study: Thermal chamber ............................................. 71
3.6 Conclusions ........................................................................ 75

4  Type-2 Fuzzy PI Controller with Adjustable Type-reduced Output 76
4.1 Realization of Type-2 Fuzzy PI Controller .............................. 78
4.2 Analysis of Type-2 Fuzzy PI Controller ................................ 82
4.3 Theorems on Properties of Centroids .................................... 83
4.4 Adaptive Algorithm for Type-reduction ................................ 88
  4.4.1 Switch point adjustment algorithm ................................. 88
  4.4.2 Derivatives of centroid with respect to switch points .......... 92
4.4.3 Algorithm initialization ............................................. 94
4.5 Simulation Results ..................................................... 98
4.6 Comparison with Fuzzy PI Gain-scheduling Control .......... 102
  4.6.1 Uncertain parameters for pH neutralization process ........ 105
  4.6.2 Simulation results for pH neutralization process with uncer-
       tain parameters ....................................................... 108
4.7 Case Study: Thermal chamber ...................................... 110
4.8 Conclusions ............................................................ 113

5 On-line Learning Algorithm for Type-2 Fuzzy-Neural Controller 115
  5.1 Type-1 and Type-2 Fuzzy-Neural Systems—General Background . 116
  5.2 Architecture of type-2 FNC ......................................... 118
  5.3 Control Scheme of Type-2 Fuzzy-Neural Control System ........ 122
  5.4 On-line Self-learning Algorithm for MF Variables and Weights .... 124
    5.4.1 Weight update rules ........................................... 126
    5.4.2 MF variables update rules .................................... 130
  5.5 Case Study: pH Neutralization Process ......................... 138
    5.5.1 Performance of type-2 FNC with online weights and MF vari-
           ables update ..................................................... 144
    5.5.2 Performance of type-1 FNC .................................... 155
  5.6 Case Study: Thermal chamber ..................................... 162
    5.6.1 Performance of type-2 FNC .................................... 163
    5.6.2 Performance of conventional PI controller and type-1 FNC
           with 12 rules .................................................... 171
  5.7 Conclusion ........................................................... 176

6 Conclusions and Future Work ........................................ 178
  6.1 Conclusions .......................................................... 178
  6.2 Suggestions for Future Work ....................................... 181
Appendix A  Relationship between FOU and control surface  183
   A.1  Control surface using type-2 triangles with uncertain base  .  .  186
   A.2  Control surface using parallel type-2 triangles  .  .  .  .  .  .  .  .  .  189
   A.3  Control surface using type-2 triangles with uncertain peak  .  .  192

Appendix B  Update rules for lower MF variables  196

Author’s Publications  199

Bibliography  200
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Examples for type-1 fuzzy set and singleton</td>
<td>14</td>
</tr>
<tr>
<td>2.2</td>
<td>The structure of type-1 FLS</td>
<td>15</td>
</tr>
<tr>
<td>2.3</td>
<td>The fuzzy sets of fuzzy PID controller</td>
<td>17</td>
</tr>
<tr>
<td>2.4</td>
<td>Structure of a Type-2 rule-based FLS</td>
<td>24</td>
</tr>
<tr>
<td>2.5</td>
<td>Type-2 membership functions</td>
<td>25</td>
</tr>
<tr>
<td>2.6</td>
<td>Example of interval type-2 membership function</td>
<td>26</td>
</tr>
<tr>
<td>2.7</td>
<td>Upper or lower membership function and embedded fuzzy set</td>
<td>28</td>
</tr>
<tr>
<td>2.8</td>
<td>Example of embedded type-2 fuzzy set</td>
<td>29</td>
</tr>
<tr>
<td>2.9</td>
<td>Switch points for calculating the centroid</td>
<td>35</td>
</tr>
<tr>
<td>3.1</td>
<td>The structure of fuzzy PI controller</td>
<td>51</td>
</tr>
<tr>
<td>3.2</td>
<td>The antecedents of PD-like FLSs</td>
<td>52</td>
</tr>
<tr>
<td>3.3</td>
<td>Triangular non-singleton fuzzifier with small spread for $e$</td>
<td>53</td>
</tr>
<tr>
<td>3.4</td>
<td>Triangular non-singleton fuzzifier with small spread for $\dot{e}$</td>
<td>54</td>
</tr>
<tr>
<td>3.5</td>
<td>Triangular non-singleton fuzzifier with large spread</td>
<td>57</td>
</tr>
<tr>
<td>3.6</td>
<td>Rectangular nonsymmetric non-singleton fuzzifier</td>
<td>59</td>
</tr>
<tr>
<td>3.7</td>
<td>Titration curve for a weak acid, strong base reaction</td>
<td>62</td>
</tr>
<tr>
<td>3.8</td>
<td>The CSTR configuration with two influent streams</td>
<td>63</td>
</tr>
<tr>
<td>3.9</td>
<td>The control scheme for CSTR</td>
<td>64</td>
</tr>
<tr>
<td>3.10</td>
<td>The details of $e$ and $\dot{e}$ of the proposed nonsymmetric non-singleton</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>fuzzy PD plus integrator fuzzy controller at the steady state pH=8.5</td>
<td></td>
</tr>
</tbody>
</table>
3.11 Comparison of singleton type-1 PI controllers with moving average filters and non-singleton fuzzy PD plus integrator controller  

3.12 The pH responses of singleton PI controller and proposed non-singleton fuzzy controllers at different setpoints  

3.13 The responses of proposed non-singleton fuzzy controllers with different $v$  

3.14 The responses of proposed non-singleton fuzzy controllers with different $\alpha_f$  

3.15 The responses of proposed non-singleton fuzzy controllers with different $B_v$  

3.16 Diagram of a thermal chamber  

3.17 The responses of proposed non-singleton controller and conventional singleton controller  

3.18 Control signals of proposed non-singleton controller and conventional singleton controller  

4.1 The input and output fuzzy sets  

4.2 Lower and upper bounds of type-reduced output set  

4.3 An example of type-2 fuzzy set  

4.4 An example of two Theorems for the particular type-2 fuzzy set  

4.5 The standard fuzzy set used in this chapter  

4.6 Illustration of equivalent gains for type-2 PI using the algorithm  

4.7 ITAEs of type-2 fuzzy PI controller in Monte Carlo uncertainty analysis  

4.8 Histogram of ITAEs of type-2 fuzzy PI controller  

4.9 ITAEs of type-1 fuzzy PI controller in Monte Carlo uncertainty analysis  

4.10 Histogram of ITAEs of type-1 fuzzy PI controller  

4.11 Responses of three control systems when $K = 0.9$, $\tau = 4.5$. (a) The first step response; (b) Step response after adaptation.
4.12 Responses of three control systems when $K = 1.1$, $\tau = 5.5$. (a) The first step response; (b) Step response after adaptation. .................. 104
4.13 Antecedents of fuzzy PI gain-scheduling controller .......................... 105
4.14 Consequents for $K_p^*$ and $K_i^*$ of fuzzy PI gain-scheduling controller 106
4.15 Performances of proposed type-2 PI and conventional gain-scheduling controllers .......................... 109
4.16 Histogram of the ISEs between the reference and responses of type-2 PI controller ........................................... 110
4.17 Histogram of the ISEs between the reference and responses of gain scheduling PI controller .......................... 111
4.18 Responses of different control systems when the fan speed is 30% of full speed .................................................. 113
4.19 Responses of different control systems when the fan speed is 80% of full speed .................................................. 114
5.1 Fuzzy-neural network implementing a fuzzy inference procedure .......................... 119
5.2 Creation of Footprint of Uncertainty (FOU) from type-1 fuzzy sets .................. 120
5.3 Feed-forward feedback FNC ........................................... 123
5.4 Flow of full update algorithm ........................................... 139
5.5 Schematic diagram of pH system ........................................... 140
5.6 Titration curve of the pH neutralization process ........................................... 143
5.7 Reference trajectory of simulation for modelling ability .......................... 145
5.8 Type-2 antecedent fuzzy sets of type-2 FNC with 9 rules .......................... 146
5.9 Response of type-2 FNC with 9 rules in 1st iterations .......................... 148
5.10 Response of type-2 FNC with 9 rules after 50 iterations .......................... 148
5.11 ISEs of the performance of type-2 FNC with 9 rules .......................... 149
5.12 Weights of type-2 FNC with 9 rules during the 50 learning iterations 149
5.13 Weights of type-2 FNC with 9 rules at the 50th iteration .......................... 150
5.14 MF variables for lower MFs of input \( r \) of type-2 FNC with 9 rules at the 50th iteration ............................................. 150
5.15 MF variables for upper MFs of input \( r \) of type-2 FNC with 9 rules at the 50th iteration ............................................. 151
5.16 MF variables for input \( \dot{r} \) of type-2 FNC with 9 rules at the 50th iteration ......................................................... 151
5.17 Optimized antecedents of type-2 FNC with 9 rules after 50 learning iterations ......................................................... 153
5.18 Control surface of type-2 FNC with 9 rules and optimized weights and MF variables .................................................. 154
5.19 Control slice when \( \dot{r} = 0 \) of type-2 FNC with 9 rules and optimized weights and MF variables ......................... 154
5.20 Antecedent fuzzy sets of type-1 FNC with 9 rules ................................................................. 156
5.21 Control slice with three key points at initialization and titration curve 156
5.22 ISEs of the performance of type-1 FNC with 9 rules ................................................................. 157
5.23 Weights of type-1 FNC with 9 rules during the 50 learning iterations 158
5.24 Weights of type-1 FNC with 9 rules at the 50th iteration ................................................................. 158
5.25 Antecedent fuzzy sets of type-1 FNC with 25 rules ................................................................. 159
5.26 Control slice with five key points at initialization and titration curve 160
5.27 ISEs of the performance of type-1 FNC with 25 rules ................................................................. 161
5.28 Weights of type-1 FNC with 25 rules during the 50 learning iterations 161
5.29 Weights of type-1 FNC with 25 rules at the 50th iteration ................................................................. 162
5.30 Disturbance from the fan with uncertain rotation speed ................................................................. 163
5.31 Type-2 antecedent fuzzy sets of type-2 FNC with 4 rules ................................................................. 164
5.32 Response of type-2 FNC with 4 rules in the last iteration ................................................................. 165
5.33 ISEs of the performance of type-2 FNC with 4 rules ................................................................. 165
5.34 Weights of type-2 FNC with 4 rules during the learning iterations ................................................................. 166
5.35 Weights of type-2 FNC with 4 rules at the last iteration (dashed lines as estimated average values) ................................................................. 167
5.36 MF variables for input \( r \) of type-2 FNC with 4 rules ................. 168
5.37 Deviation of MF variables learning trajectories from the mean for input $r$ .................................................. 168
5.38 MF variables for input $r$ of type-2 FNC with 4 rules at the 50th iteration 169
5.39 MF variables for input $\dot{r}$ of type-2 FNC with 4 rules .................. 169
5.40 Deviation of MF variables learning trajectories from the mean for input $\dot{r}$ .................................................. 170
5.41 MF variables for input $\dot{r}$ of type-2 FNC with 4 rules at the 50th iteration 170
5.42 Response of conventional PI controller in the last iteration ............ 172
5.43 ISEs of conventional PI controller ........................................ 172
5.44 Type-1 antecedent fuzzy sets of type-1 FNC with 12 rules ............. 173
5.45 Response of type-1 FNC with 12 rules in the last iteration ............. 173
5.46 ISEs of the performance of type-1 FNC with 12 rules ................. 174
5.47 Weights of type-1 FNC with 12 rules during the learning iterations .... 175
5.48 Weights of type-1 FNC with 12 rules at the last iteration .............. 175
5.49 Comparison of performances for the three controllers .............. 176
A-1 MF variables for type-2 triangles with uncertain base ............... 184
A-2 MF variables for parallel type-2 triangles ................................. 184
A-3 MF variables for type-2 triangles with uncertain peak ............... 185
A-4 Control surface of the type-1 FLS ........................................ 186
A-5 Control surface of type-2 FLS using type-2 triangles with uncertain base for X1 only .............................................. 187
A-6 Control slice of type-2 FLS using type-2 triangles with uncertain base for X1 only .............................................. 188
A-7 Control surface of type-2 FLS using type-2 triangles with uncertain base for X1 only–A new case of MF variables’ combination ......... 189
A-8 Control slice of type-2 FLS using type-2 triangles with uncertain base for X1 only–A new case of MF variables’ combination ............. 190
A-9 Control surface of type-2 FLS using parallel type-2 triangles for X1 only .................................................. 191
A-10 Control slice of type-2 FLS using parallel type-2 triangles for X1 only 191
A-11 Control slice of type-2 FLS using type-2 triangles with uncertain peak for X1 only ................................. 193
A-12 Control surface of type-2 FLS using type-2 triangles with uncertain peak for X1 only—Large upper MF variables .................................................. 193
A-13 Control slice of type-2 FLS using type-2 triangles with uncertain peak for X1 only—Large upper MF variables .................................................. 195
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Partial overlap between the input and antecedent $S_1$ sets</td>
<td>46</td>
</tr>
<tr>
<td>3.2</td>
<td>Fuzzified input base is a subset of $S_1$ base</td>
<td>49</td>
</tr>
<tr>
<td>3.3</td>
<td>$S_1$ base is a subset of fuzzified input base</td>
<td>50</td>
</tr>
<tr>
<td>3.4</td>
<td>Consequent singletons of the fuzzy PD FLS</td>
<td>52</td>
</tr>
<tr>
<td>3.5</td>
<td>Parameters of the nonsymmetric non-singleton fuzzy PD plus integrator controller for pH setpoint at 8.5</td>
<td>65</td>
</tr>
<tr>
<td>3.6</td>
<td>Noise level at different pH setpoint and recommended control parameters</td>
<td>66</td>
</tr>
<tr>
<td>3.7</td>
<td>Mean-squared errors and standard deviations of singleton and non-singleton fuzzy PI controllers for noise rejection during steady state period</td>
<td>67</td>
</tr>
<tr>
<td>3.8</td>
<td>Parameters of the nonsymmetric non-singleton fuzzy PD plus integrator controller for thermal chamber temperature control</td>
<td>73</td>
</tr>
<tr>
<td>4.1</td>
<td>ITAEs of responses in figures after several adaptation iterations</td>
<td>102</td>
</tr>
<tr>
<td>4.2</td>
<td>Fuzzy tuning rules for $K_p$</td>
<td>106</td>
</tr>
<tr>
<td>4.3</td>
<td>Fuzzy tuning rules for $K_i$</td>
<td>106</td>
</tr>
<tr>
<td>5.1</td>
<td>Parameter and initial conditions of the pH plant</td>
<td>144</td>
</tr>
<tr>
<td>5.2</td>
<td>Simulation parameters of the type-2 FNC</td>
<td>147</td>
</tr>
<tr>
<td>5.3</td>
<td>Simulation parameters of the type-1 FNC</td>
<td>157</td>
</tr>
<tr>
<td>5.4</td>
<td>Experiment parameters of the type-2 FNC</td>
<td>166</td>
</tr>
</tbody>
</table>
Summary

The objective of the thesis is to use extensional fuzzy theories to develop fuzzy controllers that are capable of maintaining the desired performance of uncertain or nonlinear systems and to investigate the advantages on handling uncertainties offered by the extra freedom inside the type-2 fuzzy sets.

First, the possibility of using non-singleton FLS to better handle sensor noise is investigated. Since existing singleton and non-singleton fuzzifiers do not fully utilize the ability of fuzzy sets to handle input uncertainties, a new non-singleton fuzzifier is proposed. The fuzzification strategy is designed to have minimal impact on the system dynamics and to reduce the steady-state fluctuations caused by the presence of noise.

Non-singleton type-1 FLS cannot handle other kinds of uncertainties. This shortcoming leads to the introduction of expanded fuzzy sets, known as type-2 fuzzy sets, that have an extra dimension for modelling uncertainties. In order to better understand type-2 FLS, a type-2 fuzzy PI controller whose control surface is bounded based on the uncertainty is constructed to control systems with uncertain but bounded parameters. An adaptive algorithm for adjusting the switch points to obtain variable centroids is proposed to generate a suitable output surface within the pre-determined control surface range to maintain the desired performance. Finally, by utilizing the extra dimension in the type-2 fuzzy sets, an on-line self-learning scheme is proposed for a type-2 fuzzy-neural control systems. The objective is to investigate the capability of the extra degrees of freedoms (FOU) in the type-2 FLS in modelling complex input-output relationship.
Chapter 1

Introduction

1.1 Uncertainty in the Real World

Uncertainty is ubiquitous in the real world to make things different from one another. When dealing with real-world problems, uncertainty can be rarely avoided. At the empirical level, uncertainty is an inseparable companion of almost any measurement, resulting from a combination of inevitable measurement errors and resolution limits of measuring instruments. At the cognitive level, it emerges from the vagueness and ambiguity inherent in natural language. At the social level, uncertainty has even strategic uses and it is often created and maintained by people for different purposes (privacy, secrecy, propriety)[36].

Over many years, a variety of strategies have been developed to deal with different kinds of uncertainties, where dealing with the uncertainties means to minimize the deleterious effects of these uncertainties[57]. It has been pointed out that uncertainty is a result of some information deficiency. Information (pertaining to the model within which the situation is conceptualized) may be incomplete, fragmentary, not fully reliable, vague, contradictory, or deficient in some other way[36]. In addition to a lack of complete information, uncertainty may also reflect incompleteness, imprecision, missing information, or randomness in data and a process[7]. Moreover, there are also linguistic uncertainties as words mean different things to different
people[63] and experts do not always agree on the design of the controllers[64].

A general discussion about uncertainty is not the aim of this thesis. The motivation is to develop strategies to handle or control different kinds of uncertainty, which is usually encountered in control engineering problems. In real control problems, people often encounter situations of inadequate system models. When controlling complex systems, a large quantity of sensory measurements may be difficult to interpret accurately. Efficient computational power for control actions to achieve a desired performance of the systems may also possibly be lacking. Fuzzy sets, the foundation of fuzzy theory, were introduced forty years ago as a way of expressing non-probabilistic uncertainties[97]. Since then, fuzzy theory has been applied to construct different kinds of fuzzy controllers to control systems where tradition methods may not have good results.

1.2 Historical Review on Fuzzy Control

Zadeh proposed fuzzy theory more than 40 years ago because the real world is too complicated for precise descriptions to be obtained, therefore approximation (or fuzziness) must be introduced in order to obtain a reasonable, yet trackable, model [92]. As early as 1962, Zadeh wrote that to handle biological systems “we need a radically different kind of mathematics, the mathematics of fuzzy or cloudy quantities which are not describable in terms of probability distributions” [96]. Later, Zadeh formalized these ideas into the paper “Fuzzy Sets”. The fuzzy logic theory introduced by Zadeh is also termed type-1 fuzzy logic. Since then, fuzzy logic theory has developed and found applications in database management, operations analysis, decision support systems, signal processing, data classifications, computer vision, etc[11].

The most significant applications, however, have concentrated on control problems since the birth of fuzzy controllers for real systems in 1975[92]. Mamdani and Assilian first established the basic framework of fuzzy controller based on Mam-
Dani fuzzy logic system (FLS) and applied the fuzzy controller to control a steam engine[52]. Control of cement kilns was another early industrial application[21]. Since the first consumer product using fuzzy logic was marketed in 1987, the use of fuzzy control has increased substantially. A number of CAD environments for fuzzy control design have emerged together with VLSI hardware for fast execution[2]. Early work in fuzzy control utilized the linguistic nature of fuzzy control that makes it possible to express process knowledge concerning how the process should be controlled or how the process behaves. The fuzzy controllers can provide smooth interpolation between discrete controller outputs since fuzzy systems are often regarded as smooth function approximation schemes. The main contribution of fuzzy control is its ability to handle many practical problems that cannot be adequately managed by conventional control techniques. At the same time, the results of fuzzy control theory are consistent with the existing classical ones when the system under control reduces from fuzzy to non-fuzzy. The aim of fuzzy control systems theory is to extend the existing successful conventional control systems techniques and methods as much as possible, and to develop many new and special-purposed ones, for a much larger class of complex, complicated, and ill-modelled systems — fuzzy systems[11].

The early fuzzy controllers used the system error and its rate of change as inputs to determine the desired change in the control value setting via the heuristic knowledge embedded in a linguistic rule base. This architecture closely resembles the versatile PID control strategy used extensively in industries. Research work has shown that conventional PID controllers can be realized by singleton type-1 fuzzy controllers using product t-norm for fuzzy inference engine and height defuzzification[68]. However, a fundamental problem of linguistic fuzzy controllers is that the entire design is often guided only by the designer’s experiences about the process.

In order to formulate a systematic design procedure and to reduce the dependence on expert knowledge, a promising approach that combines neural networks and fuzzy logic systems into an integrated system was proposed in the 1990s[47].
Neural networks proposed by J. J. Hopfield in the early 1980’s has been applied to classify, store, recall and associate information or patterns. “Back-propagation Algorithm” by Rumelhart, Hinton, and William further extended the learning capability and improved the learning ability of neural networks. This concept of trainable neural networks further strengthens the idea of utilizing the learning ability of neural networks to learn the fuzzy control rules and the membership functions of a fuzzy logic control system. The combination brings the low-level computational power and learning ability of neural networks into fuzzy logic systems to automate and realize the design of fuzzy logic control systems; it also provides the high level IF-THEN rule thinking and reasoning of fuzzy logic systems into neural networks.

1.3 Extension to Type-1 Fuzzy Logic Theory

In spite of the many applications utilizing type-1 fuzzy controllers, type-1 fuzzy set and fuzzy logic system (FLS) is not adequate for handling all kinds of uncertainty when constructing rule-based FLS. It is known that the uncertain knowledge used to construct a FLS may arise from the following sources: 1) the words used in the antecedents and the consequents of rules can mean different things to different people, 2) consequents obtained by polling a group of experts may differ, 3) the training data are noisy, and 4) the measurements that activate the FLS are noisy. Conventional (Type-1) fuzzy sets are a generalization of crisp sets which can only state that the output is either true or false. Even though the word fuzzy has the connotation of uncertainty, Klir and Floger pointed out “…it may seem problematical, if not paradoxical, that a representation of fuzziness is made using membership grades that are themselves precise real numbers.” Since research has shown that the usefulness of Type-1 fuzzy sets is limited by its crisp membership grades, there are efforts made to extend conventional fuzzy sets and fuzzy logic theory so that the extensional FLS may handle more uncertainty.
1.3.1 Non-singleton type-1 fuzzy logic systems

The conventional type-1 FLS, with a singleton fuzzifier, may not always be adequate when noise is present in the training data or in the data processed by the system. Non-singleton fuzzifier was thus proposed to account for uncertainty in the data. A non-singleton type-1 FLS is a type-1 FLS whose inputs are modelled as type-1 fuzzy number. Hence, it can be used to handle uncertainties that occur when uncertain inputs are applied to a type-1 FLS. The early forms of non-singleton input had been applied for many years. Muyaram utilized fuzzy numbers in empirical rules to optimize the fuel consumption rate of a marine diesel engine [75]. Later, Balazinski used vector of fuzzy sets both to train a fuzzy neural network and as input during processing[6]. These methods were more flexible and faster than conventional singleton fuzzy controller and they both introduced the idea of expressing the data as fuzzy sets. Finally, Mendel and Mouzouris extended this idea and proposed a non-singleton formulation of FLS[73] and used the non-singleton FLS in non-linear time-series analysis[74]. The results showed the non-singleton FLS minimized uncertain effects of noise in the data much better than the original singleton type-1 FLS. Their system could predict the future time-series satisfactorily but there is limited study on the design method of fuzzifier. This is a severe limitation as the relationship between shapes of non-singleton fuzzifier and minimizing effect of noise should be very useful to design a suitable fuzzifier for noisy inputs. In addition, little application of non-singleton fuzzy logic system in the control field is found in literature. Hence, the topic of shaping non-singleton fuzzifier could be further investigated to design a suitable non-singleton fuzzifier for a fuzzy controller.

1.3.2 Type-2 fuzzy logic systems

Although the non-singleton FLS is able to handle uncertainties in the input signals, it does not explicitly handle the other kinds of uncertainty mentioned in the first paragraph in this section. A new type of fuzzy set was introduced by Zadeh in 1975
It is called type-2 fuzzy set in order to differentiate from its ordinary type-1 counterpart. A type-2 fuzzy set is defined as one that has a fuzzy membership function, i.e. the membership grade is a fuzzy set in the unit interval [0,1], rather than a point in [0,1]. Such fuzzy sets are useful in situations where the shape or the parameters of the membership functions are uncertain. Although the notion of type-2 fuzzy set has been introduced for a long time, very little work was published about it until the mid nineties. Also, due to its complexity, type-2 fuzzy logic theory was not formally formulated until recently initial research works focused on the properties of type-2 fuzzy set. Mizumoto and Tanaka studied the set theoretic operations of type-2 fuzzy sets and properties of membership grades of such sets. They also examined type-2 fuzzy sets under the operations of algebraic product and algebraic sum. Nieminen provided more detail about the algebraic structure of type-2 fuzzy sets. Dubois and Prade discussed fuzzy valued logic and provided a formula for the composition of type-2 relations as an extension of the type-1 sup-star composition. All these works laid the foundation for type-2 fuzzy logic theory, and they demonstrated the flexibility of type-2 fuzzy sets which can accommodate more uncertain information.

The watershed for the field occurred when Mendel and Karnik extended the works of Mizumoto and Tanaka with practical algorithms for performing union, intersection, and complement of a type-2 fuzzy set. By using Zadeh’s Extension Principle, Karnik and Mendel proposed a general formula for the extended sup-star composition of type-2 relations. It can be viewed as a nonlinear mapping of a type-2 input fuzzy set into another type-2 output fuzzy set where the calculations are based on the operations of union and intersection for type-2 fuzzy sets. Karnik and Mendel also developed the concept of the centroid of a type-2 fuzzy set and the accompanying computational algorithm. Later, they proposed type-reduction methods that map a type-2 set into a type-1 fuzzy set, based on computing the centroid of the combined type-2 fuzzy set. From the type-reduced fuzzy set, a defuzzified output for the type-2 FLS can then be easily derived us-
ing different defuzzification methods. Based on these results, Karnik, Mendel and Liang established a complete type-2 fuzzy logic theory [33]. A general type-2 FLS is too complicated for applications because of its higher dimension in membership function of type-2 fuzzy sets. Hence, Liang and Mendel proposed the theory and design of interval type-2 FLSs [43] which are less computationally complex in related operations. Mendel also pointed out that only interval type-2 fuzzy sets are practical for type-2 FLSs because the computations of union, intersection, extended sup-star composition and type-reduction are less complicated [59]. Nearly all the applications of type-2 FLSs up to now are using interval type-2 FLSs.

Type-2 fuzzy sets provide us with more design degrees of freedom, so using Type-2 fuzzy sets has the potential to outperform systems using Type-1 fuzzy sets, especially in uncertain environments. Since the type-2 FLS can better handle numerical and linguistic uncertainties via an extra degree of freedom [57], type-2 FLSs have been successfully applied to more and more fields, including but not limited to

- **Signal processing**: [44, 42, 81],
- **decision making**: [78, 77],
- **finance**: [41, 5],
- **clustering**: [24],
- **time-series forecasting**: [30],
- **survey processing**: [29, 4],
- **pattern recognition**: [67, 20, 100],
- **wireless communication**: [45, 84],
- **noise cancellation**: [9],
- **system identification**: [40],
- **embedded agent**: [12],
- **health care**: [37, 23],
- **robotics**: [3, 90, 53],
- **marine engine control**: [50],
- **power engineering**: [1, 72],
- **quality control**: [54],
- **plant diagnostics**: [8, 10] and
- **hidden markov models**: [99]

### 1.3.3 Recent research in type-2 fuzzy controllers

Much research is continuing on interval type-2 FLSs and some research are starting to employing general type-2 FLSs [58, 86]. Researchers from all over the world work on developing different kinds of type-2 FLSs, although the number and growth rate of applications are still not comparable to its conventional counterpart. Control engineering, which is the original most widely applied field for type-1 FLSs, has
now gradually become a major focus of attention for interval type-2 FLSs since 2003[55]. Hagras proposed a novel hierarchical type-2 fuzzy architecture for the real time control of mobile robots navigating in changing and dynamic unstructured indoor and outdoor environments[16]; he also proposed an incremental adaptive life long learning approach for type-2 fuzzy embedded agents in ambient intelligent environments[17]. Phokharatkul and Phaiboon also applied a type-2 FLS to control a mobile robot’s direction for obstacle avoidance and corridor following[80]. Wu and Tan proposed a simplified architecture of type-2 FLS and applied it in real-time control of coupled tank system[94]. Figuero and et al applied a type-2 fuzzy controller for tracking mobile objects in the context of robotic soccer games[15]. Sepulveda and et al examined the ability of type-2 fuzzy controller in handling uncertainty[83]. Lin and et al designed a type-2 fuzzy logic controller for buck DC-DC converters[48].

There are some other works that utilized neural based system to learn the parameters of type-2 fuzzy controllers since type-1 fuzzy neural systems have been successfully developed and applied in last decade. Melin and Castillo designed an adaptive controller for non-linear plants using Type-2 fuzzy logic and neural networks[56]. Lee and Lin applied type-2 fuzzy neural systems with adaptive filter to nonlinear uncertain systems[39]. Singh and et al also proposed a type-2 fuzzy neural model based controller for a nonlinear system[85]. Wang, Chen and Lee developed a type-2 fuzzy neural network to handle uncertainty with dynamical optimal learning[91]. Excellent results were obtained for the truck backing-up control and the identification of nonlinear system, which yield more improved performance than those using type-1 FNN. The advantage of introducing neural network is that the consequent weights can be updated automatically by the BP method. However, they only used GA to generate suitable membership functions and did not study the BP update algorithm for parameters of membership functions. Lynch et al. recently presented the result of using uncertainty bounds in the design of embedded real-time type-2 neuro-fuzzy speed controller for marine diesel engines[50, 51]. The main contribution
of his study is to use an approximation algorithm for type-reduction and thus this simpler method made it possible to derive a BP update algorithm for parameters of membership functions. Nonetheless, both of these works of type-2 fuzzy-neuro systems are only for off-time learning. Therefore, on-line algorithm using type-2 fuzzy-neuro systems with original type-reduction and membership function update algorithm remains an interesting topic to investigate.

1.4 Aims and Scope of the Work

More attentions have been paid to apply type-2 FLSs to control uncertain systems, but the number of papers on the topic is still small compared to the thousands of papers on applications of type-1 fuzzy controllers. Hence, the area of type-2 fuzzy control is still a fertile field for research. For completeness, the first step of my research is to investigate the relationship between the shape of non-singleton fuzzifier and effect on modelling uncertain input, and hopefully it may provide some guidelines to develop a non-singleton fuzzy controller and enhance its noise rejection performance.

Another motivation of my research is to develop type-2 fuzzy controllers that can handle different uncertainties and provide a suitable control surface. Centroid is a very important concept since it is used in type-reduction to provide the range of control surface that models the uncertain information. Under the motivation of seeking the relationship between centroid and uncertainty, Mendel and Wu have done some work to show the properties of centroid of an interval type-2 fuzzy set[66]. Paradoxically, the study found that when only interval symmetrical type-2 fuzzy sets are used to perform operations (e.g. arithmetic, set-theoretic and nonlinear function on them), the results will also be symmetrical interval type-2 fuzzy sets. Hence, the result of combined centroid plus defuzzification procedures could be the same as those of particular symmetrical type-1 fuzzy sets with same defuzzification method[66]. Thus, the uncertain information included in centroid may not be well
utilized when performing type-reduction and defuzzification. This property will
definitely reduce the practical usefulness of type-2 FL and the improvement for
a type-2 FLS under such condition is strongly needed. As a result, an adaptive
type-reduction method based on information of uncertainty is desirable and very
practical to generate an adaptive output surface corresponding with different type-1
FL system to handle different cases of uncertainty.

On the other hand, type-2 fuzzy set has an extra dimension in fuzzy set[57]
and this extra freedom may provide the type-2 FLS with more freedom to model
uncertain or complex relationship. The last main motivation of my research is to
develop on-line update algorithms for type-2 fuzzy-neuro systems to accomplish the
task of modelling uncertain or complex relationship and examine the advantage of
the extra freedom in type-2 fuzzy sets.

This thesis seeks to develop controllers utilizing extensional fuzzy logic theo-
ries, namely non-singleton fuzzy logic and type-2 fuzzy logic and evaluate these
controllers’ performance on handling different kinds of uncertainty. In view of the
above discussion, the specific objectives are as follows:

1. To examine the efficiency of original non-singleton fuzzifier on modeling the
noisy input and to design a new non-singleton fuzzifier to improve the perfor-
mane on minimizing the effect of uncertain information in the input.

2. To develop an adaptive type-reduction method based on properties of centroid
for an interval type-2 fuzzy logic controller to obtain a variable control sur-
face. To evaluate the performance of such a type-2 fuzzy logic controller with
variable control surface to track a reference trajectory when the system are
uncertain but the parameters of the system are bounded.

3. To construct a type-2 fuzzy-neuro controller (FNC) using BP algorithm for
updating the consequent and antecedent parameters online. To evaluate the
online performance of a type-2 fuzzy-neuro controller when it is applied to a
nonlinear and uncertain systems.
Fuzzy neural network (FNN) system can be tuned both for neuron parameters and the structure of network, but the uncertain information is mainly described in the fuzzy neurons. Hence, the update of the structure of network in the FNC is beyond the scope of this study. Structure will then remain the same during learning iterations and only parameters of fuzzy neurons are updated. The purpose is to study only how the FOU[62], or updated parameters of membership function affect the performance of the type-2 FNN, and the benefit of updating the structure of the network has to be eliminated.

1.5 Organization of the Thesis

In order to provide readers with a solid understanding of the extensional fuzzy logic theories, Chapter 2 provides a prime theoretical introduction for non-singleton FLSs and type-2 FLSs. Chapter 3 presents a new type of non-singleton type-1 fuzzy controller for improving the performance of handling uncertainties, such as noise rejection in nonlinear control system. The contribution of Chapter 3 is that the new type of non-singleton fuzzifier improves the noise rejection performance compared with the traditional non-singleton fuzzifier. Chapter 4 develops a type-2 fuzzy PI controller to control a process whose parameters are uncertain such that the performance of the proposed controller can be maintained even when the system parameters deviate from their nominal values. The contribution is that it provides a systematic framework to set up a desired type-2 fuzzy controller whose control surface are bounded within pre-determined range. The theories developed in Chapter 4 provide the base of adaptive algorithm to generate variable control surface to minimize the effect of uncertainties and maintain desired performance. Chapter 5 proposes an online learning algorithm for tuning the parameters of a type-2 fuzzy-neuro controller(T2FNC). It utilizes the feedback error signal and its derivatives to train the FNC. Both the consequents and antecedents are adjusted simultaneously. The main contributions of this chapter are to derive an online
update algorithm for the first time that is suitable for any rule-base fuzzy systems and provide investigation on the ability of FOU in type-2 sets to model uncertainty and nonlinear relationship. Finally, Chapter 6 will be the concluding remarks of the previous research work and some discussion about the future work.
Chapter 2

Theories on Extensional Fuzzy Logic

This chapter will briefly introduce some of the important concepts and theorems for the extensional fuzzy logic, namely non-singleton type-1 FLSs and type-2 FLSs. To set the background, the simplest fuzzy logic system—singleton type-1 FLS and the relationship between conventional PID controller and type-1 FLS will be introduced first. After that, the structures of non-singleton type-1 FLSs and type-2 FLSs will be introduced comparing with that of the singleton type-1 FLSs.

2.1 Singleton Type-1 Fuzzy Logic Systems

A crisp set $A$ in a universe of discourse $X$ can be defined by a zero-one membership function:

$$
A \Rightarrow \mu_A(x) = \begin{cases} 
1 & \text{if } x \in A \\
0 & \text{if } x \notin A 
\end{cases}
$$

(2.1)

where $A$ can be defined as $A=\{x|x$ meets some condition(s)$\}$. A type-1 fuzzy set $F$ is a generalization of a crisp set. It is defined on a universe of discourse $X$ and is characterized by a membership function $\mu_F(x)$ whose grades are in the interval $[0,1]$. The grade of a membership function gives a degree of similarity of a member
in $X$ to the fuzzy set. When $X$ is continuous, $F$ can be presented as

$$F = \int_X \mu_F(x) / x$$

where the integral sign does not mean integration but collection of all points $x \in X$ with associated membership function $\mu_F(x)$. When $X$ is discrete, $F$ can be presented as

$$F = \sum_X \mu_F(x) / x$$

where the summation sign does not mean arithmetic addition but collection of all points $x \in X$ with associated membership function $\mu_F(x)$. The support of a fuzzy set $F$ is the crisp set of all points $x$ in $X$ whose $\mu_F(x)$ is non-zero. The spread $S$ is the distance from the point $x$ which has the maximum membership grade to one end of the base. A fuzzy set whose support is a single point in $X$ is called a type-1 fuzzy singleton. Examples of type-1 fuzzy set with its left spread and singleton are shown in Figure 2.1.

![Figure 2.1. Examples for type-1 fuzzy set and singleton](image)

After the type-1 fuzzy set appeared, large volumes of literature has blossomed about it in a wide number of fields. Applications can also be found in many areas, e.g. medicine[25], finance[41], computational linguistic[89] and car control[46]. In a majority of type-1 fuzzy set applications, rule-based fuzzy logic systems(FLSs) is
the most powerful and popular design methodology. The rule-based FLSs contain four components—rules, fuzzifier, inference engine and output processor. The FLSs can be regarded as a mapping from inputs to outputs.

![Diagram of type-1 FLS](image)

**Figure 2.2. The structure of type-1 FLS**

The simplest rule-based FLS is called singleton type-1 FLS which is also known as Mamdani FLS. The structure of such a type-1 FLS is shown in Figure 2.2. All the fuzzy sets are type-1 and the measurements are perfect and treated as crisp values[57]. In the FLSs, crisp inputs are first fuzzified into fuzzy input sets in order to activate the inference engine. The fuzzifier maps a crisp point \( x = (x_1, \cdots, x_p)^T \in X_1 \times X_2 \times \cdots \times X_p \equiv X \) into a fuzzy set \( A_x \) in \( X \). In singleton type-1 FLSs, singleton fuzzification is applied and the whole computation process is the easiest. The singleton fuzzifier is just a fuzzy singleton:

\[
A_x \text{ is a fuzzy singleton with support } x' \text{ such that } \mu_{A_x}(x) = 1 \text{ for } x = x' \text{ and } \mu_{A_x}(x) = 0 \text{ for all other } x \in X \text{ with } x \neq x'
\]

Consider a type-1 FLS having \( p \) inputs \( x_1, \cdots, x_p \in X_p \) and one output \( y \in Y \). Suppose that it has \( M \) rules, then the \( l \)th rule has the following form:

\[
R^l : IF \ x_1 \ is \ F_1^l \ and \ \cdots \ and \ x_p \ is \ F_p^l , \ THEN \ y \ is \ G^l \ l = 1, \cdots, M
\]
This rule represents a type-1 fuzzy relation between the input space $X_1 \times \cdots \times X_p$ and the output space, $Y$, of the FLS. In a type-1 FLS, $F^l_i$, $i = 1, \cdots, p$ and $C^l$ $l = 1, \cdots, M$ are all type-1 fuzzy sets.

In the fuzzy inference engine, fuzzy logic principles are used to combine the fuzzy IF–THEN rules that are activated to produce a mapping from the input fuzzy sets in $X_1 \times \cdots \times X_p$ to fuzzy output sets in $Y$. Each rule can be interpreted as a fuzzy implication. Hence, the fuzzy inference engine can be interpreted as a system that maps fuzzy sets into fuzzy sets by means of the sup–star composition:

$$\mu_{B^l}(y) = \mu_{G^l}(y) \star [\sup_{x_1 \in X_1} \mu_{x_1}(x_1) \star \mu_{F^l_1}(x_1)] \star \cdots \star [\sup_{x_p \in X_p} \mu_{x_p}(x_p) \star \mu_{F^l_p}(x_p)], \quad y \in Y$$

where $\mu_{G^l}(y)$ is the membership function of the consequent fuzzy set and the bracketed term is the firing level for the $l$th fuzzy rule. The final fuzzy set, $B$, is determined by all $M$ rules and is obtained by combining $B^l$ and its membership function $\mu_{B^l}(y)$ for all $l = 1, \cdots, M$.

For singleton fuzzification, the supremum operation in the sup-star composition is very easy to evaluate because $\mu_{x_i}(x_i)$ is non-zero only at one point $x_i = x'_i$; hence

$$\mu_{B^l}(y) = \mu_{G^l}(y) \star [\sup_{x_1 \in X_1} \mu_{x_1}(x_1) \star \mu_{F^l_1}(x_1)] \star \cdots \star [\sup_{x_p \in X_p} \mu_{x_p}(x_p) \star \mu_{F^l_p}(x_p)]$$

$$= \mu_{G^l}(y) \star [\mu_{x_1}(x'_1) \star \mu_{F^l_1}(x'_1)] \star \cdots \star [\mu_{x_p}(x'_p) \star \mu_{F^l_p}(x'_p)]$$

$$= \mu_{G^l}(y) \star [\mu_{F^l_1}(x'_1) \cdots \star \mu_{F^l_p}(x'_p)], \quad y \in Y$$

The term in the bracket on the last line of Equation (2.6) is referred as the firing level. Actually $\mu_{B^l}(y)$ is a membership function and it depends on $x = x'$; change $x'$ and $\mu_{B^l}(y)$ changes. Singleton fuzzification will greatly simplify the sup-star composition in (2.6).

Defuzzification produces a crisp output from the fuzzy sets that appear at the
output of the inference block in Figure 2.2. One of the computationally simplest
defuzzifiers is \textit{height defuzzifier} which is also called the \textit{center average defuzzifier}.
The height defuzzifier replaces each rule output fuzzy set by a singleton at the point
having maximum membership in that output set, and then calculates the centroid
of the type-1 set comprised of these singletons. The output of a height defuzzifier is
given as

\begin{equation}
y_h(x) = \frac{\sum_{i=1}^{M} y_i^l \mu_{B^l}(y_i^l)}{\sum_{i=1}^{M} \mu_{B^l}(y_i^l)}
\end{equation}

where $y_i^l$ is the point having maximum membership in the $l$th output set (if there
is more than one such point, their average can be taken as $y_i^l$), and its membership
grade in the $l$th output set is $\mu_{B^l}(y_i^l)$.

\section{Realization of PID Control Using Type-1 FLSs}

The most significant applications of FLSs have concentrated on control problems
since the birth of fuzzy controllers for real systems in 1975\cite{92}. In 1975, Mam-
dani and Assilian first established the basic framework of fuzzy controller based on
Mamdani FLS and applied the fuzzy controller to control a steam engine\cite{52}.

Conventional PID controllers are perhaps the most well-known and most widely
used controllers in the modern industries: statistics has shown that most controllers
used in industries today are PID or PID-type of controllers. PID controllers are
simple reliable and effective. Research work has shown that conventional PID con-
trollers can be realized by singleton type-1 fuzzy controllers using product t-norm
for fuzzy inference engine and height defuzzification\cite{68}.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure23.png}
\caption{The fuzzy sets of fuzzy PID controller}
\end{figure}
Consider a PID controller whose output is:

\[ u = \alpha e + \beta \dot{e} + \gamma \int e \, dt \]  

(2.8)

where \( e \) is the error; \( \dot{e} \) as derivative of error; \( \int e \, dt \) as integral of error; and \( \alpha \) is the propositional coefficient, \( \beta \) is the derivative coefficient, and \( \gamma \) is the integral coefficient for the controller.

Figure 2.3 shows the antecedent fuzzy sets of a fuzzy controller which is equivalent to a conventional PID controller. There are two antecedent fuzzy sets each for \( e \) or \( \dot{e} \) or \( \int e \, dt \). Let \( e_1 \) and \( e_2 \) be the minimum and maximum values for possible error \( e \); and let \( de_1 \) and \( de_2 \) be the minimum and maximum values for possible derivative of error \( \dot{e} \); and let \( ie_1 \) and \( ie_2 \) be the minimum and maximum values for possible integral of error \( \int e \, dt \), i.e.

\[ e_1 \leq e \leq e_2 \quad de_1 \leq \dot{e} \leq de_2 \quad ie_1 \leq \int e \, dt \leq ie_2 \]  

(2.9)

The rules for the equivalent fuzzy PID controller are:

\begin{align*}
\text{Rule 1} : & \quad e_1 \text{ and } de_1 \text{ and } ie_1 \Rightarrow u_1 \\
\text{Rule 2} : & \quad e_1 \text{ and } de_1 \text{ and } ie_2 \Rightarrow u_2 \\
\text{Rule 3} : & \quad e_1 \text{ and } de_2 \text{ and } ie_1 \Rightarrow u_3 \\
\text{Rule 4} : & \quad e_1 \text{ and } de_2 \text{ and } ie_2 \Rightarrow u_4 \\
\text{Rule 5} : & \quad e_2 \text{ and } de_1 \text{ and } ie_1 \Rightarrow u_5 \\
\text{Rule 6} : & \quad e_2 \text{ and } de_1 \text{ and } ie_2 \Rightarrow u_6 \\
\text{Rule 7} : & \quad e_2 \text{ and } de_2 \text{ and } ie_1 \Rightarrow u_7 \\
\text{Rule 8} : & \quad e_2 \text{ and } de_2 \text{ and } ie_2 \Rightarrow u_8 \\
\end{align*}  

(2.10)

where the fuzzy sets for the output space are fuzzy singletons associated with the
following real numbers:

\[
\begin{align*}
  u_1 &= \alpha e_1 + \beta de_1 + \gamma ie_1 \\
  u_2 &= \alpha e_1 + \beta de_1 + \gamma ie_2 \\
  u_3 &= \alpha e_1 + \beta de_2 + \gamma ie_1 \\
  u_4 &= \alpha e_1 + \beta de_2 + \gamma ie_2 \\
  u_5 &= \alpha e_2 + \beta de_1 + \gamma ie_1 \\
  u_6 &= \alpha e_2 + \beta de_1 + \gamma ie_2 \\
  u_7 &= \alpha e_2 + \beta de_2 + \gamma ie_1 \\
  u_8 &= \alpha e_2 + \beta de_2 + \gamma ie_2 
\end{align*}
\] (2.11)

which are the controller output of Equation (2.8) at the crisp input pairs \((e_1, de_1, ie_1), \ldots, (e_2, de_2, ie_2)\).

The control action \(u\) of fuzzy PID controller for certain crisp input pair \((e_0, \dot{e}_0, \int e \, dt')\) if using product t-norm and height defuzzification is given as:

\[
\begin{align*}
  u &= abc u_1 + ab(1 - c)u_2 + a(1 - b)cu_3 + a(1 - b)(1 - c)u_4 + (1 - a)bcu_5 \\
   &\quad + (1 - a)b(1 - c)u_6 + (1 - a)(1 - b)cu_7 + (1 - a)(1 - b)(1 - c)u_8 \\
   &\quad = abc u_1 + ab(1 - c)u_2 + a(1 - b)cu_3 + a(1 - b)(1 - c)u_4 + (1 - a)bcu_5 \\
   &\quad + (1 - a)b(1 - c)u_6 + (1 - a)(1 - b)cu_7 + (1 - a)(1 - b)(1 - c)u_8 \\
   &= \alpha e' + \beta \dot{e}' + \gamma \int e \, dt' \\
\end{align*}
\] (2.12)

where

\[
\begin{align*}
  a &= \mu_{e_1}'(e') = \frac{e_2 - e'}{e_2 - e_1}, & b &= \mu_{de_1}(\dot{e}') = \frac{de_2 - \dot{e}'}{de_2 - de_1}, & c &= \mu_{ie_1}(\int e \, dt') = \frac{ie_2 - \int e \, dt'}{ie_2 - ie_1} \\
\end{align*}
\] (2.13)

as shown in Figure 2.3.

Therefore, it is shown that PID controller can be constructed by singleton type-1 fuzzy controller with product t-norm and height defuzzification.
However, if using other t-norm, defuzzification method or other membership function for antecedents and consequents, the resulting fuzzy PID controller may not be exactly same as the linear conventional PID controller. The fuzzy PID controller is now actually non-linear version of conventional PID controller. The fuzzy PID controllers are generally superior to the conventional ones, particularly for higher-order, time-valued, and nonlinear systems, and for those systems that have only vague mathematical models which are difficult, if not impossible, for a conventional PID to handle. Such nonlinear fuzzy PID controllers contain variable control gains in contrast to the conventional PID controllers where the control gains are constant.

2.3 Non-singleton Type-1 Fuzzy Logic Systems

When there are uncertainties that occur at the inputs of FLS (e.g. noise measurements), a non-singleton type-1 fuzzy logic system (FLS) can be utilized to handle the uncertainties.[57] A non-singleton type-1 FLS is described by the same diagram as Figure 2.2. The rules of a non-singleton type-1 FLS are the same as those for a singleton type-1 FLS. The difference is the fuzzifier, where input signals are modelled as type-1 fuzzy numbers; i.e. the membership function is associated with the crisp input.

A non-singleton fuzzifier is one for which $\mu_{X_i}(x'_i) = 1 \ (i = 1, \cdots, p)$

and $\mu_{X_i}(x_i)$ decreases from unity as $x_i$ moves away from $x'_i$.

The membership function for $X_i$, $\mu_{x_i}$, indicates that the sensor reading $x$ is the most likely to be the true value, while the adjacent points are also possible but to a lesser degree because the inputs are corrupted by noise.[73] Hence, a non-singleton fuzzy logic controller system is a generalization of singleton fuzzy logic system in order to provide a more flexible way to handle input uncertainties.

Consider a type-1 fuzzy logic system which has $p$ inputs $x_1 \in X_1, \cdots, x_p \in X_p$ and only one output $y \in Y$. The type-1 fuzzy controller has $M$ rules, where the $l$th
rule is
\[ R_l: \text{IF } x_1 \text{ is } F^l_1 \text{ and } \ldots \text{ and } x_p \text{ is } F^l_p, \text{ Then } y \text{ is } G^l \quad (l = 1, \ldots, M) \]

In the fuzzy inference engine, fuzzy logic principles are used to combine fuzzy IF–THEN rules from the fuzzy rule base into a mapping from fuzzy input sets in \( X_1 \times \cdots \times X_p \) to fuzzy output sets in \( Y \). Each rule is interpreted as a fuzzy implication. With reference to (2.4), let \( F^l_1 \times \cdots \times F^l_p = A^l \); then, (2.4) can be re-expressed as
\[ R_l : F^l_1 \times \cdots \times F^l_p \rightarrow G^l = A^l \rightarrow G^l \quad l = 1, \ldots, M \quad (2.14) \]

\( R^l \) is described by the membership function \( \mu_{R^l}(x, y) \), where
\[ \mu_{R^l}(x, y) = \mu_{A^l \rightarrow G^l}(x, y) \quad (2.15) \]

and \( x = (x_1, \ldots, x_p)^T \). Hence, \( \mu_{R^l}(x, y) = \mu_{R^l}(x_1, \ldots, x_p, y) \) and
\[ \mu_{R^l}(x, y) = \mu_{A^l \rightarrow G^l}(x, y) = \mu_{F^l_1 \times \cdots \times F^l_p \rightarrow G^l}(x, y) \]
\[ = \mu_{F^l_1 \times \cdots \times F^l_p}(x) \bigstar \mu_{G^l}(y) \]
\[ = \mu_{F^l_1}(x_1) \bigstar \cdots \bigstar \mu_{F^l_p}(x_p) \bigstar \mu_{G^l}(y) \]
\[ = [T_{i=1}^p \mu_{F^l_i}(x_i)] \bigstar \mu_{G^l}(y) \quad (2.16) \]

where it has been assumed that Mamdani implications are used and \( \bigstar \) and \( T \) denote t-norms.

The \( p \)-dimensional input to \( R^l \) is given by the fuzzy set \( A_x \) whose membership function is defined as
\[ \mu_{A_x}(x) = \mu_{X_1}(x_1) \bigstar \cdots \bigstar \mu_{X_p}(x_p) = T_{i=1}^p \mu_{X_i}(x_i) \quad (2.17) \]
where \( X_i (i = 1, \ldots, p) \) are the labels of the fuzzy sets describing the inputs. Each rule \( R^l \) determines a fuzzy set \( B^l = A_x \circ R^l \) in \( Y \) such that

\[
\mu_{B^l}(y) = \mu_{A_x \circ R^l}(y) = \sup_{x \in X} [\mu_{A_x}(x) \star \mu_{A^l \rightarrow G^l}(x, y)], \quad y \in Y
\]  

(2.18)

This equation is the input-output relationship between the fuzzy set that excites a one-rule inference engine and the fuzzy set at the output of that engine. The sup-star composition is a highly nonlinear mapping from the input vector \( x \) into a scalar output fuzzy set \( \mu_{B^l}(y) \). Substituting Equation (2.16) and (2.17) into (2.18), then

\[
\mu_{B^l}(y) = \sup_{x \in X} [\mu_{A_x}(x) \star \mu_{A^l \rightarrow G^l}(x, y)]
\]

(2.19)

The last line follows from the commutativity of a t-norm and the fact that \( \mu_{x_i}(x_i) \) is only a function of \( x_i \), then each supremum in Equation (2.19) is just a scalar variable. The final fuzzy output set, \( B \), is obtained by combining \( B^l \) of all \( M \) rules and its membership function \( \mu_{B^l}(y) \) for \( l = 1, \ldots, M \).

For singleton fuzzification, the firing level can be easily calculated because each \( \mu_{x_i}(x_i) \) is non-zero only at \( x \). When non-singleton fuzzification is utilized, \( \mu_{x_i}(x_i) \) is a type-1 fuzzy set so the firing level is the supremum of \( \mu_{Q^l_k} \equiv \mu_{x_k}(x_k) \star \mu_{F^l_k}(x_k) \). Denoting this firing level as \( x_{k, \text{max}}^l \), the fuzzy inference engine then can be re-expressed as:

\[
\mu_{B^l}(y) = \mu_{G^l}(y) \star [T_{k=1}^{p} \mu_{Q^l_k}(x_{k, \text{max}}^l)]
\]  

(2.20)
where $T_{k=1}^p$ represents a sequence of $p$ t-norm operations. Using height defuzzifier, the output of the non-singleton type-1 fuzzy controller may be expressed as:

$$y_h(x) = \frac{\sum_{l=1}^{M} y_l \mu_{B_l}(\overline{y}_l)}{\sum_{l=1}^{M} \mu_{B_l}(\overline{y}_l)} = \frac{\sum_{l=1}^{M} y_l \prod_{k=1}^{p} \mu_{Q_k^l}(x_{k,max}^l)}{\sum_{l=1}^{M} \prod_{k=1}^{p} \mu_{Q_k^l}(x_{k,max}^l)}$$

(2.21)

where $\overline{y}_l$ is the point having maximum membership in the $l$th output set and its membership grade in the $l$th output set is $\mu_{B_l}(\overline{y}_l)$. Equations (2.20) and (2.21) indicate that the non-singleton fuzzifier effectively transforms the input $x$ into $x_{l,max}^l$. Hence, the non-singleton fuzzification can be viewed as a pre-filter that estimates the true input value in order to account for effect of input uncertainties.

### 2.4 Type-2 Fuzzy Logic Theories

Type-1 fuzzy logic systems (FLSs) contain four components – fuzzifier, expert rules (knowledge base), inference engine and defuzzifier. Expert rules are expressed in IF-THEN statements which are known as antecedent and consequent. The fuzzifier maps crisp inputs into type-1 fuzzy input sets. Rules in the knowledge base are then fired at varying degrees by the fuzzy input sets. The inference engine maps fuzzy sets into type-1 fuzzy output sets using the sup-star composition. Finally, a crisp output value is obtained by the defuzzifier.

Although the type-1 non-singleton FLS is able to handle uncertainty in input, it is not adequate to handle other kinds of uncertainty as stated in Chapter 1 and thus type-2 FLS is introduced to handle these kinds of uncertainty. Since fuzzy sets are associated with terms in the antecedents and consequents of the rules in the knowledge base, a type-2 FLS is one that employs at least one type-2 fuzzy set. A fuzzy set of higher type changes the nature of the membership functions and the corresponding operations, but the basic principles of a fuzzy logic system do not change. As shown in Figure 2.4, a typical type-2 FLS has the same basic components as a Type-1 FLS. The main difference is the inclusion of a type-reducer.
in the output processing to map the type-2 output sets produced by the inference engine into a type-1 fuzzy set. As a beginning, some concepts about type-2 fuzzy set will be introduced first.

2.4.1 Type-2 membership functions

A type-2 fuzzy membership function may be used to describe the strength of belief when it is difficult to determine crisp membership grades. Figure 2.5(a) shows that the membership function of a type-2 fuzzy set can be obtained by blurring the membership function of a type-1 fuzzy set to the left or the right [57]. For any specific input $x$, the membership grade is not a crisp value anymore, but may assume a number of values wherever the vertical line intersects the blurred membership function. A type-2 membership function is a three-dimensional function and the extra dimension provides the type-2 fuzzy sets with the ability to handle uncertainties.

Definition 2.1. Mathematically, a type-2 fuzzy set ($\tilde{A}$) is defined by the type-2 membership function $\mu_{\tilde{A}}(x, u)$, i.e.

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x, u)/(x, u) \quad J_x \subseteq [0, 1]$$

(2.22)

where $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$, $\int \int$ denotes union over all admissible $x$ and $u$. $J_x \subseteq [0, 1]$
is known as the primary membership of $x$.

For discrete universes of discourse $X$ and $U$, a type-2 fuzzy set ($\tilde{A}$) is defined as,

$$
\tilde{A} = \sum_{x \in X} \sum_{u \in J_x} \mu_{\tilde{A}}(x, u) / (x, u) \quad J_x \subseteq [0, 1]
$$

(2.23)

Figure 2.5. Type-2 membership functions

Figure 2.5(b) shows an example of a type-2 membership function that depicts $\mu_{\tilde{A}}(x, u)$ when $x$ and $u$ are discrete. The restriction, $J_x \subseteq [0, 1]$, is consistent with the type-1 constraint $0 \leq \mu_A(x) \leq 1$. When the blur disappears, the type-2 membership function will reduce to a type-1 membership function, in which the variable $u$ equals $\mu_A(x)$ and $0 \leq \mu_A(x) \leq 1$.

**Definition 2.2.** The 2D plane, a vertical slice of $\mu_{\tilde{A}}(x, u)$ for $x = x'$ is called a secondary membership function:

$$
\mu_{\tilde{A}}(x = x', u) \equiv \mu_{\tilde{A}}(x') = \int_{u \in J_{x'}} f_{x'}(u) / u \quad J_{x'} \subseteq [0, 1]
$$

(2.24)

where $f_{x'}(u)$ is called secondary grade and $0 \leq f_{x'}(u) \leq 1$.

Based on the concept of secondary sets, a general type-2 fuzzy set can be repre-
presented as the union of all secondary sets, or vertical slices.

\[
\tilde{A} = \int_{x \in X} \mu_{\tilde{A}}(x)/x = \int_{x \in X} \left[ \int_{u \in J_x} f_x(u)/u \right]/x \quad J_x \subseteq [0, 1] \tag{2.25}
\]

If \( X \) and \( J_x \) are both discrete, assuming \( x \) has been discretized into \( N \) points, namely \( x_1, \ldots, x_N \) and at each of these points, \( u \) has been discretized into \( M_i \) points, namely \( u_{i1}, \ldots, u_{iM_i} \).

\[
\tilde{A} = \sum_{x \in X} \left[ \sum_{u \in J_x} f_x(u)/u \right]/x = \sum_{i=1}^{N} \left[ \sum_{k=1}^{M_i} f_{x_i}(u_{ik})/u_{ik} \right]/x_i \tag{2.26}
\]

**Definition 2.3.** The domain of a secondary membership function is called the primary membership of \( x \). In Equation (2.22), \( J_x \) is the primary membership of \( x \).

**Definition 2.4.** When \( f_x(u) = 1, \ J_x \subseteq [0, 1] \), then the secondary membership functions are interval sets. If this is true for any \( x \in X \), then the type-2 fuzzy set is called interval type-2 fuzzy set. Interval secondary membership functions reflect a uniform uncertainty at the primary memberships of \( x \). Example is shown in Figure 2.6.

![Figure 2.6. Example of interval type-2 membership function](image-url)
Definition 2.5. The union of all primary memberships, which is shown as the blurred area in Figure 2.5, is defined as the footprint of uncertainty (FOU).

\[ FOU(\tilde{A}) = \bigcup_{x \in X} J_x \]  

(2.27)

FOU is very useful because it provides information about uncertainties in the shape and position of the membership function. When all the uncertainties disappear, the type-2 fuzzy sets reduce to the type-1 fuzzy sets. Although the type-2 fuzzy set may assume many shapes, interval type-2 fuzzy set is the only practical implementation because of its less complex implementation than general type-2 fuzzy set as shown in [59]. Since the secondary grades are the same, interval type-2 fuzzy sets and their uncertainties can be expressed uniquely by the FOU. In other words, they can be described using the membership functions which are associated with the upper and lower bounds of FOU. These membership functions are known as the upper and lower membership functions.

Definition 2.6. An upper membership function and a lower membership function are two type-1 membership functions that are bounds for the FOU of a type-2 fuzzy set \( \tilde{A} \). The upper membership function, denoted as \( \mu_{\tilde{A}}(x) \), is associated with the upper bound of \( FOU(\tilde{A}) \). The lower membership function, denoted as \( \underline{\mu}_{\tilde{A}}(x) \), is associated with the lower bound of \( FOU(\tilde{A}) \). Example is shown in Figure 2.7.

2.4.2 Embedded type-2 and type-1 sets

A type-2 fuzzy set \( \tilde{A} \) can be regarded as a collection of embedded type-2 fuzzy sets \( \tilde{A}_e \). [63]

Definition 2.7. For continuous universes of discourse \( X \) and \( U \), an embedded type-2 fuzzy set \( \tilde{A}_e \) is:

\[ \tilde{A}_e = \int_{x \in X} [f_x(\theta)/\theta]/x, \quad \theta \in J_x \subseteq U = [0, 1] \]  

(2.28)
The fuzzy set \( \tilde{A}_e \) is embedded in \( \tilde{A} \) and there are an uncountable number of embedded type-2 fuzzy sets.

For each value of \( x \), the fuzzy set \( \tilde{A}_e \) has only one primary membership, namely \( \theta \) and an associated secondary grade \( f_x(\theta) \).

When computing using type-2 fuzzy sets, the domains of \( X \) and \( U \) are always discretized, in which case there are a finite number of embedded type-2 sets.

**Definition 2.8.** For discrete universes of discourse \( X \) and \( U \), an **embedded type-2 fuzzy set** \( \tilde{A}_e \) has \( N \) elements, where \( \tilde{A}_e \) contains exactly one element from \( J_{x_1}, \ldots, J_{x_N} \), namely \( \theta_1, \ldots, \theta_N \), each with its associated secondary grade, namely \( f_{x_1}(\theta_1), \ldots, f_{x_N}(\theta_N) \), i.e.,

\[
\tilde{A}_e = \sum_{i=1}^{N} \left[ f_{x_i}(\theta_i)/\theta_i \right]/x_i, \quad \theta_i \in J_{x_i} \subseteq U = [0, 1]
\] (2.29)

The fuzzy set \( \tilde{A}_e \) is embedded in \( \tilde{A} \) and there are a finite number of embedded type-2 fuzzy sets. Figure 2.8 shows an example of one embedded type-2 fuzzy set in \( \tilde{A} \) of Figure 2.5(b).

**Definition 2.9.** For continuous universes of discourse \( X \) and \( U \), an **embedded type-1 fuzzy set** \( A_e \) is:

\[
A_e = \int_{x \in X} \theta/x, \quad \theta \in J_x \subseteq U = [0, 1]
\] (2.30)
Figure 2.8. Example of embedded type-2 fuzzy set

The fuzzy set $A_e$ is the union of all the primary memberships of embedded type-2 fuzzy set $\tilde{A}_e$ and there are an uncountable number of embedded type-1 fuzzy sets. Example of embedded type-1 fuzzy set is also shown in Figure 2.7. Both the upper and lower membership function are embedded type-1 fuzzy sets.

**Definition 2.10.** For discrete universes of discourse $X$ and $U$, an embedded type-1 fuzzy set $A_e$ has $N$ elements, where $A_e$ contains exactly one element from $J_{x_1}, \ldots, J_{x_N}$, namely $\theta_1, \ldots, \theta_N$, i.e.,

$$A_e = \sum_{i=1}^{N} \theta_i/x_i, \quad \theta_i \in J_{x_i} \subseteq U = [0, 1]$$

(2.31)

The fuzzy set $A_e$ is the union of all the primary memberships of embedded type-2 fuzzy set $\tilde{A}_e$ and there are a finite number of embedded type-1 fuzzy sets.

Although a general type-2 fuzzy set was shown to be represented as collection of vertical slices in Equation (2.25), a new representation Theorem has been proposed to represent a general type-2 fuzzy set as collection of embedded type-2 fuzzy sets, or so-called wavy slices for discrete universes of discourse $X$ and $U$.\[63\]
Theorem 2.1. (Representation Theorem): For a general type-2 fuzzy set \( \tilde{A} \), whose universes of discourse \( X \) and \( U \) are discrete, the footprint of uncertainty (FOU) is equal to the union of all of its embedded type-1 fuzzy sets. Let \( \tilde{A}_e^j \) denotes the \( j \)th embedded type-2 fuzzy set for \( \tilde{A} \), i.e.,

\[
\tilde{A}_e^j = \sum_{i=1}^{N} [f_{x_i}(\theta_{i1})/\theta_{i1}]/x_i, \quad \theta_{i1} \in J_{x_i}, \quad i = 1, \cdots, N
\]  

(2.32)

where \( J_{x_i} \) is discretized into \( M_i \) elements, namely \( \theta_{i1}, \cdots, \theta_{iM_i} \); and \( X \) is discretized into \( N \) elements, namely \( x_1, \cdots, x_N \).

Then \( \tilde{A} \) can be represented as the union of its embedded type-2 fuzzy sets, i.e.,

\[
\tilde{A} = \sum_{j=1}^{n} \tilde{A}_e^j, \quad n = \prod_{i=1}^{N} M_i
\]

(2.33)

2.4.3 Operations of type-2 fuzzy sets

Representation Theorem provides a new way to understand the innate feature of type-2 fuzzy set, it also plays an important role to simplify the derivation of formulas for union, intersection and complement of type-2 fuzzy sets, without having to use the Extension Principle. [63]

Consider two type-2 fuzzy sets \( \tilde{A} \) and \( \tilde{B} \), i.e.,

\[
\tilde{A} = \int_{X} \mu_{\tilde{A}}(x) / x = \int_{X} \left[ \int_{J_{u}^{x}} f_{x}(u)/u \right] / x, \quad J_{u}^{x} \subseteq [0,1]
\]

(2.34)

\[
\tilde{B} = \int_{X} \mu_{\tilde{B}}(x) / x = \int_{X} \left[ \int_{J_{w}^{x}} g_{x}(w)/w \right] / x, \quad J_{w}^{x} \subseteq [0,1]
\]

(2.35)

Definition 2.11. The union of \( \tilde{A} \) and \( \tilde{B} \) is another type-2 fuzzy set as

\[
\mu_{\tilde{A} \cup \tilde{B}}(x) = \int_{u \in J_{u}^{x}} \int_{w \in J_{w}^{x}} f_{x}(u) \star g_{x}(w)/v \equiv \mu_{\tilde{A}}(x) \sqcup \mu_{\tilde{B}}(x)
\]

(2.36)

where \( v \equiv u \vee w \) and “\( \sqcup \)” denotes the so-called join operation. “\( \star \)” indicates minimum or product t-norms, and “\( \vee \)” represents the max t-conorm.
Definition 2.12. The intersection of $\tilde{A}$ and $\tilde{B}$ is another type-2 fuzzy set as

$$\mu_{\tilde{A} \cap \tilde{B}}(x) = \int_{u \in J_2} \int_{w \in J_2^v} f_x(u) \star g_x(w) / v = \mu_{\tilde{A}}(x) \cap \mu_{\tilde{B}}(x)$$  \hspace{1cm} (2.37)

where $v \equiv u \wedge w$ and “$\cap$” denotes the so-called meet operation. “$\wedge$” also represents product or minimum t-norm.

Definition 2.13. The complement of $\tilde{A}$ is another type-2 fuzzy set as

$$\mu_{\tilde{A}}(x) = \int_{u \in J_2} f_x(u) / (1 - u) \equiv -\mu_{\tilde{A}}(x), \quad x \in X$$  \hspace{1cm} (2.38)

where “$\sim$” denotes the so-called negation operation.

2.4.4 Centroid of type-2 fuzzy sets

As mentioned, comparing with the defuzzification of a type-1 FLS, an important calculation in the output processing part for a type-2 FLS is type reduction, which is an extension of type-1 defuzzification procedure by using the Zadeh’s Extension Principle[98]. In this thesis, the center of sets type reduction method is employed. Since it is based on the concept of generalized centroid, a method to compute the centroid of a type-2 fuzzy set will be described in this subsection.

Definition 2.14. Suppose the type-2 fuzzy set, $\tilde{A}$, whose input domain, $x$, is discretized into $N$ points. Then, the centroid of such a fuzzy set is defined as follows[31],

$$C_{\tilde{A}} = \int_{\theta_1 \in J_{x_1}} \cdots \int_{\theta_N \in J_{x_N}} \left[ f_{x_1}(\theta_1) \star \cdots \star f_{x_N}(\theta_N) \right] / \left( \sum_{i=1}^{N} x_i \theta_i / \sum_{i=1}^{N} \theta_i \right)$$  \hspace{1cm} (2.39)

where $C_{\tilde{A}}$ is a type-1 fuzzy set and $\star$ is product t-norm or minimum t-norm.

Every combination of $\theta_1, \cdots, \theta_N$ and its associated secondary grade $f_{x_1}(\theta_1) \star \cdots \star f_{x_N}(\theta_N)$ forms an embedded type-2 set, $\tilde{A}_e$. From the definition of the centroid, it can be seen that the centroid $C_{\tilde{A}}$ is actually the union of all the centroid of the
embedded type-2 set \( \tilde{A}_e \), while \( \sum_{i=1}^{N} x_i \theta_i / \sum_{i=1}^{N} \theta_i \) is the centroid of the embedded type-1 fuzzy set \( \tilde{A} \).

If \( \tilde{A} \) is an interval type-2 set, Equation (2.39) can then be rewritten as

\[
C_{\tilde{A}} = \frac{1}{\sum_{i=1}^{N} x_i \theta_i / \sum_{i=1}^{N} \theta_i} = [c_l, c_r]
\]  

(2.40)

The centroid is an interval type-1 set which is described by its two ends, \( c_l \) and \( c_r \).

In order to extend the center-of-sets defuzzifier to the center-of-sets type-reducer, both \( x_i \) and \( \theta_i \) should be type-1 sets. This requirement is the motivation for introducing the concept of *generalized centroid*, \( GC \): [31]

\[
GC = \frac{\sum_{i=1}^{N} z_i w_i}{\sum_{i=1}^{N} w_i}
\]

where \( T \) and \( \ast \) are t-norm; \( Z_l \) and \( W_l \) are type-1 fuzzy set with associated membership functions \( \mu_{Z_l}(z_l) \) and \( \mu_{W_l}(w_l) \).

If each \( Z_l \) and \( W_l \) are interval type-1 sets, then the \( GC \) can be rewritten as,

\[
GC = \frac{\sum_{i=1}^{N} z_i w_i}{\sum_{i=1}^{N} w_i} = [y_l, y_r]
\]

It is clear that now the problem to compute \( GC \) becomes to maximize and minimize \( y \) which is treated as a function of \( w_1, \ldots, w_N \), i.e.,

\[
y(w_1, \ldots, w_N) = \frac{\sum_{i=1}^{N} z_i w_i}{\sum_{i=1}^{N} w_i}
\]

For general type-2 fuzzy set, the centroid and general centroid have high computational complexity, since the calculation involves all the embedded type-2 fuzzy set. However, for interval fuzzy set, the calculation becomes more practical, since Karnik and Mendel has developed an iterative procedure to computer the interval type-1 set[31].

**Theorem 2.2. (Karnik-Mendel Method):** Assuming each \( Z_l \) is an interval type-1

...
1 set having center $c_l$ and spread $s_l (s_l > 0)$, and each $W_l$ is also an interval type-1 set having center $h_l$ and spread $\Delta_l (\Delta_l > 0)$.

(a) The maximum of $y(w_1, \cdots, w_N)$ can be obtained by the following iterative procedure. Set $z_l = c_l + s_l (l = 1, \cdots, N)$ and without loss of generality assume that the $z_l$ are arranged in ascending order, i.e., $z_1 \leq \cdots \leq z_N$. Then:

1. Initialize $w_l$ by setting $w_l = h_l$ for $l = 1, \cdots, N$ and then compute $y' = y(h_1, \cdots, h_N)$ using Equation (2.41).

2. Find $k (1 \leq k \leq N - 1)$ such that $z_k \leq y' \leq z_{k+1}$.

3. Set $w_l = h_l - \Delta_l$ for $l \leq k$ and $w_l = h_l + \Delta_l$ for $l \geq k + 1$, and compute $y'' = y(h_1 - \Delta_1, \cdots, h_k - \Delta_k, h_{k+1} + \Delta_{k+1}, \cdots, h_N + \Delta_N)$ using Equation (2.41).

4. Check if $y'' = y'$. If yes, stop and $y''$ is the maximum value of $y(w_1, \cdots, w_N)$.
   If no, go to Step 5.

5. Set $y'$ equal to $y''$. Go to Step 2.

(b) The maximum of $y(w_1, \cdots, w_N)$ can be obtained by using a similar iterative procedure. Only two changes need to be made:

1. Set $z_l = c_l - s_l (l = 1, \cdots, N)$

2. In Step 3, set $w_l = h_l + \Delta_l$ for $l \leq k$ and $w_l = h_l - \Delta_l$ for $l \geq k + 1$, and compute $y'' = y(h_1 + \Delta_1, \cdots, h_k + \Delta_k, h_{k+1} - \Delta_{k+1}, \cdots, h_N - \Delta_N)$ using Equation (2.41).

2.4.5 Properties of the centroid for an interval type-2 set

The centroid for an interval type-2 fuzzy set is a very important concept since it provides a measure of the uncertainty for an interval type-2 set. Using the Representation Theorem, the centroid $C_{\tilde{A}}$ is the collection of all the centroids of its embedded
interval type-2 fuzzy sets. The centroid of an interval type-2 set is an interval type-1 set \([c_l, c_r]\). The two ends \(c_l\) and \(c_r\) can be represented as,

\[
    c_l = \min \left( \sum_{i=1}^{N} x_i \theta_i / \sum_{i=1}^{N} \theta_i \right), \quad \theta_i \in [\mu_{\tilde{A}}(x_i), \mu_{\tilde{A}}(x_i)]
\]

\[\text{(2.41)}\]

\[
    c_r = \max \left( \sum_{i=1}^{N} x_i \theta_i / \sum_{i=1}^{N} \theta_i \right), \quad \theta_i \in [\mu_{\tilde{A}}(x_i), \mu_{\tilde{A}}(x_i)]
\]

\[\text{(2.42)}\]

where \(x_1 \leq \cdots \leq x_N\) and the universe of discourse for the interval type-2 set \(X\) is discretized into \(N\) points, namely \(x_1, \cdots, x_N\).

Let \(A_e(l)\) denote an embedded type-1 fuzzy set for which

\[
    \mu_{A_e}(x) = \begin{cases} 
    \overline{\mu}(x) & \text{if } x \leq l \\
    \underline{\mu}(x) & \text{if } x > l
    \end{cases}
\]

\[\text{(2.43)}\]

where \(\overline{\mu}(x)\) is short for \(\mu_{\tilde{A}}(x)\) and \(\underline{\mu}(x)\) is short for \(\mu_{\tilde{A}}(x)\). \(l\) is the left switch point, i.e. the value of \(x\) at which \(A_e(l)\) switches from \(\overline{\mu}(x)\) to \(\underline{\mu}(x)\). Karnik and Mendel proved that there is only one switch between these functions, and the Karnik-Mendel Method is just used to locate that switch point iteratively. Hence,

\[
    c_l(\tilde{A}) = \min \text{ centroid}(A_e(l)), \quad (l \in X)
\]

\[\text{(2.44)}\]

where

\[
    C_{l}^\prime = \text{ centroid}(A_e(l)) = \frac{\int_{-\infty}^{l} x\overline{\mu}(x)dx + \int_{l}^{\infty} x\underline{\mu}(x)dx}{\int_{-\infty}^{l} \overline{\mu}(x)dx + \int_{l}^{\infty} \underline{\mu}(x)dx}
\]

\[\text{(2.45)}\]

\(C_{l}^\prime\) denotes the centroid of \(A_e(l)\).

Similarly, let \(A_e(r)\) denote an embedded type-1 fuzzy set for which

\[
    \mu_{A_e}(x) = \begin{cases} 
    \underline{\mu}(x) & \text{if } x \leq r \\
    \overline{\mu}(x) & \text{if } x > r
    \end{cases}
\]

\[\text{(2.46)}\]
where $r$ is the right switch point, i.e. the value of $x$ at which $A_e(r)$ switches from $\mu(x)$ to $\overline{\mu}(x)$. Hence,

$$c_r(\tilde{A}) = \max \ \text{centroid}(A_e(r)), \ (r \in X) \tag{2.47}$$

where

$$C'_r = \text{centroid}(A_e(r)) = \frac{\int_{-\infty}^{r} x\mu(x)dx + \int_{r}^{\infty} x\overline{\mu}(x)dx}{\int_{-\infty}^{r} \mu(x)dx + \int_{r}^{\infty} \overline{\mu}(x)dx} \tag{2.48}$$

$C'_r$ denotes the centroid of $A_e(r)$. Figure 2.9 shows how the switch points determine two embedded type-1 fuzzy sets to calculate the left and right end points $c_l$ and $c_r$.

![Figure 2.9. Switch points for calculating the centroid](image)

Since the introduction of the centroid, its properties have been studied [66][87]. Here some of these properties that will be utilized in the following chapter are listed for reference. These are no closed-form formulas for $c_l$ and $c_r$, but these properties provide insights about the centroid and can also greatly simplify the computation of the centroid.[61]

**Property 2.1.** The left and right end-points of the centroid, $c_l(\tilde{A})$ and $c_r(\tilde{A})$, satisfy
the following equations:

\[
\begin{align*}
cl(\tilde{A}) &= \int_{-\infty}^{c_l} \overline{\mu}(x)dx + \int_{c_l}^{\infty} x\mu(x)dx \\
&= \int_{-\infty}^{c_l} \overline{\mu}(x)dx + \int_{c_l}^{\infty} x\mu(x)dx \\
&= \int_{-\infty}^{c_l} \overline{\mu}(x)dx + \int_{c_l}^{\infty} x\mu(x)dx,
\end{align*}
\]

(2.49)

\[
\begin{align*}
cl(\tilde{A}) &= \int_{-\infty}^{c_l} x\mu(x)dx + \int_{c_l}^{\infty} \overline{\mu}(x)dx \\
&= \int_{-\infty}^{c_l} x\mu(x)dx + \int_{c_l}^{\infty} \overline{\mu}(x)dx \\
&= \int_{-\infty}^{c_l} x\mu(x)dx + \int_{c_l}^{\infty} \overline{\mu}(x)dx.
\end{align*}
\]

(2.50)

The results are very interesting as they show that the left end point of the centroid, \( cl(\tilde{A}) \) is simply the switch point \( l \) that minimizes \( \text{centroid}(A_e(l)) \); and the right end point of centroid, \( cr(\tilde{A}) \) is just the switch point \( r \) that minimizes the \( \text{centroid}(A_e(r)) \) [66].

**Property 2.2.** Let \( \tilde{A} \) be an interval type-2 fuzzy set on \( X \), and \( \tilde{A}' \) be \( \tilde{A} \) shifted by \( \Delta m \) along \( X \), i.e. \( \mu_{\tilde{A}}(x) = \mu_{\tilde{A}}(x - \Delta m) \) and \( \overline{\mu}_{\tilde{A}}(x) = \overline{\mu}_{\tilde{A}}(x - \Delta m) \). The centroid of \( \tilde{A}' \), \([cl(\tilde{A}') , cr(\tilde{A}')]\), is the same as the centroid of \( \tilde{A} \), \([cl(\tilde{A}) , cr(\tilde{A})]\), shifted by \( \Delta m \), i.e. \( cl(\tilde{A}') = cl(\tilde{A}) + \Delta m \) and \( cr(\tilde{A}') = cr(\tilde{A}) + \Delta m \). This means the span of the centroid is shift-invariant. Centroid is only associated with shape of FOU, regardless of where along \( X \) the FOU occurs [66].

**Property 2.3.** If the primary variable \( x \) is bounded, i.e. \( x \in [x_L, x_R] \), so that \( x_1 \equiv x_L \) and \( x_N \equiv x_R \), then \( cl(\tilde{A}') \geq x_L \) and \( cr(\tilde{A}') \leq x_R \). This property provides us with a quick way to check an aspect of computed values for those end points [66].

**Property 2.4.** If the interval type-2 fuzzy set \( \tilde{A} \) defined on \( X \) is symmetrical about \( m \in X \), then \( cl(\tilde{A}) \leq m \) and \( cr(\tilde{A}) \geq m \). This property shows that for a symmetrical FOU the centroid’s end points cannot cross over to the other side of the symmetry point of the FOU, i.e. \( cl \) must lie to the left of \( m \) and \( cr \) must lie to the right of \( m \) [66].

**Property 2.5.** Given a FOU for an interval type-2 fuzzy set, one that is symmetrical about primary variable \( y \) at \( y = m \), then the centroid is symmetrical about \( y = m \), and the average value (i.e., the defuzzified value) of all the elements in the centroid equals \( m \). That means, if all that is desired is a crisp number after performing
operations on interval type-2 fuzzy sets, then an interval type-2 fuzzy set that has a non-symmetrical FOU should be applied to make a difference to just using type-1 sets. [66].

### 2.4.6 Type reduction

In a type-1 FLS, the output corresponding to each fired rule is a type-1 fuzzy set in the output space. An example of a type-1 rule is:

\[ R_l^l : \text{IF } x_1 \text{ is } F_l^l \text{ and } \cdots \text{ and } x_p \text{ is } F_p^l, \text{ THEN } y \text{ is } G_l^l, \text{ } l = 1, \cdots, M \]

The defuzzifier combines the output sets corresponding to all the fired rules in some way to obtain a single output set and then finds a crisp number that is representative of this combined output set.

The output corresponding to each fired rule of a type-2 FLS is a type-2 fuzzy set. An example of a type-2 rule is:

\[ R_l^l : \text{IF } x_1 \text{ is } \tilde{F}_1^l \text{ and } \cdots \text{ and } x_p \text{ is } \tilde{F}_p^l, \text{ THEN } y \text{ is } \tilde{G}_l^l, \text{ } l = 1, \cdots, M \]

A type-reducer combines all these output sets in some way and then performs a centroid calculation on this type-2 fuzzy set, which leads to a type-1 fuzzy set which is called type-reduced set. In a word, type-reduction computes the centroid or generalized centroid for type-2 output sets.

In the thesis, center-of-sets type-reduction is used and thus shown here for demonstration, other type-reduction methods are not listed here but can be referred to [57]. Suppose a type-2 FLS has \( M \) rules, \( C_{\tilde{G}_l} \) is the centroid for the consequent set \( \tilde{G}_l \) of the \( l \)th rule. The center-of-set type-reducer finds a weighted average of these centroids. The weight associated with the \( l \)th centroid is the degree of firing corresponding to the \( l \)th rule (this will be discussed in next section), namely \( \prod_{i=1}^{p} \mu_{\tilde{F}_i}(x_i) \equiv E_l \) which is also a type-1 fuzzy set. Then, the expression for the
center-of-sets type-reduced set is in generalized centroid form as: \[57\]

\[
Y_{cos}(X) = \int_{d_1 \in C_{\tilde{G}_1}} \cdots \int_{d_M \in C_{\tilde{G}_M}} \int_{e_1 \in E_1} \cdots \int_{e_M \in E_M} T_{l=1}^M \mu_{C_{\tilde{G}_l}}(d_l) \star T_{l=1}^M \mu_{E_l}(e_l) \left/ \sum_{l=1}^M \frac{d_l e_l}{\sum_{l=1}^M e_l} \right.
\]

(2.51)

where \(T\) and \(\star\) are all t-norm.

### 2.4.7 Interval type-2 fuzzy logic systems

An interval type-2 FLS is very similar to a type-1 FLS. The major structural difference is that the defuzzifier block of a type-1 FLS is replaced by the output processing block in a type-2 FLS. The output processing block consists of type-reduction and followed by defuzzification.

Consider an interval type-2 FLS having \(p\) inputs \(x_1 \in X_1, \cdots, x_p \in X_p\) and one output \(y \in Y\). Suppose that it has \(M\) rules, then the \(l\)th rule has the following form:

\[
R^l: IF \ x_1 \ is \ \tilde{F}_1^l \ and \ \cdots \ and \ x_p \ is \ \tilde{F}_p^l, \ THEN \ y \ is \ \tilde{G}_l^l \quad l = 1, \cdots, M
\]

(2.52)

This rule represents a type-2 fuzzy relation between the input space \(X_1 \times \cdots \times X_p\) and the output space, \(Y\), of the FLS. In an interval type-2 FLS, \(\tilde{F}_i^l, i = 1, \cdots, p\) and \(\tilde{G}_l l = 1, \cdots, M\) are denoted as interval type-2 fuzzy sets by using “\(\tilde{\phantom{a}}\)”. However, this need not necessarily be the case in practice, at least one of its antecedents or consequents is interval type-2 set will make the FLS as an interval type-2 FLS.

The type-2 inference engine combines rules and gives a mapping from input type-2 fuzzy sets to output type-2 fuzzy sets. Operations, such as union, intersection of type-2 sets and compositions of type-2 relations needs to be computed. With reference to (2.52), let \(\tilde{F}_1^l \times \cdots \times \tilde{F}_p^l = \tilde{A}^l\); then, (2.52) can be re-expressed as

\[
R^l: \ \tilde{F}_1^l \times \cdots \times \tilde{F}_p^l \rightarrow \tilde{G}_l = \tilde{A}^l \rightarrow \tilde{G}_l \quad l = 1, \cdots, M
\]

(2.53)
Chapter 2. Theories on Extensional Fuzzy Logic

\( R^l \) is described by the membership function \( \mu_{R^l}(x, y) \), where

\[
\mu_{R^l}(x, y) = \mu_{\overline{A} \rightarrow \overline{G}}(x, y) = \mu_{\overline{F}^l_1}(x_1) \cap \cdots \cap \mu_{\overline{F}^l_p}(x_p) \cap \mu_{\overline{G}^l}(y) = \left[ \bigcap_{i=1}^p \mu_{\overline{F}^l_i(x_i)} \right] \cap \mu_{\overline{G}^l}(y)
\]  

(2.54)

The \( p \)-dimensional input to \( R^l \) is given by the type-2 fuzzy set \( \tilde{A}_x \) whose membership function is defined as

\[
\mu_{\tilde{A}_x}(x) = \mu_{\tilde{X}_1}(x_1) \cap \cdots \cap \mu_{\tilde{X}_p}(x_p) = \bigcap_{i=1}^p \mu_{\tilde{X}_i}(x_i)
\]  

(2.55)

where \( \tilde{X}_i(i = 1, \cdots, p) \) are the labels of the fuzzy sets describing the inputs. Each rule \( R^l \) determines a type-2 fuzzy set \( \tilde{B}^l = \tilde{A}_x \circ R^l \) in \( Y \) such that

\[
\mu_{\tilde{B}^l}(y) = \mu_{\tilde{A}_x \circ R^l}(y) = \bigcup_{x \in X} \left[ \mu_{\tilde{A}_x}(x) \cap \mu_{R^l}(x, y) \right], \ y \in Y
\]  

(2.56)

This equation is the input-output relationship between the type-2 fuzzy set that excites a one-rule inference engine and the type-2 fuzzy set at the output of that engine. Substituting Equation (2.54) and (2.55) into (2.56), then

\[
\mu_{\tilde{B}^l}(y) = \mu_{\tilde{A}_x \circ R^l}(y) = \bigcup_{x \in X} \left[ \mu_{\tilde{A}_x}(x) \cap \mu_{R^l}(x, y) \right] = \bigcup_{x \in X} \left\{ \left[ \bigcap_{i=1}^p \mu_{\tilde{X}_i}(x_i) \right] \cap \left[ \bigcap_{i=1}^p \mu_{\overline{F}^l_i(x_i)} \right] \cap \mu_{\overline{G}^l}(y) \right\} = \mu_{\overline{G}^l}(y) \cap \left\{ \left[ \bigcup_{x_1 \in X_1} \mu_{\tilde{X}_1}(x_1) \cap \mu_{\overline{F}^l_1(x_1)} \right] \cap \cdots \cap \left[ \bigcup_{x_p \in X_p} \mu_{\tilde{X}_p}(x_p) \cap \mu_{\overline{F}^l_p(x_p)} \right] \right\}, \ y \in Y
\]  

(2.57)

The last line follows from the commutativity of the meet using minimum or product and the fact that \( \mu_{\tilde{X}_i}(x_i) \cap \mu_{\overline{F}^l_i(x_i)} \) is only a function of \( x_i \), then each join in Equation (2.57) is just a scalar variable.

By using interval type-2 fuzzy sets, it is possible to obtain a closed-form formula for Equation (2.57). In addition, the type-reduction for an interval type-2 FLS is computationally practical than general type-2 FLSs. Hence, in the latter part of
the thesis, the type-2 FLSs mentioned will be interval type-2 FLSs unless other declaration.

The concepts of upper and lower membership functions play an important role in simplifying the calculations for an interval type-2 FLS. The major results for an interval singleton type-2 FLS in [43] are summarized as the following:

**Theorem 2.3.** In an interval singleton type-2 FLS with meet under product or minimum t-norm:

(a) the result of the input and antecedent operations, which are contained in the firing level set $\cap_{i=1}^{p} \mu_{F_i}(x'_i) \equiv F(x')$, is an interval type-1 set, i.e.,

$$F(x') = [f_l(x'), f_l(x')],$$

where

$$f_l = \mu_{F_1}(x'_1) \star \cdots \star \mu_{F_p}(x'_p); \quad f_l = \mu_{F_1}(x'_1) \star \cdots \star \mu_{F_p}(x'_p)$$

(b) the rule $R_l$ fired output consequent set, $\mu_{\overline{B}_l}(y)$ in Equation (2.56), is the type-1 fuzzy set:

$$\mu_{\overline{B}_l}(y) = \int_{y \in [\overline{f}_l, \overline{f}]}^{\overline{f}_l \star \cdots \star \overline{f}_l} 1/b, \quad y \in Y$$

where $\mu_{\overline{G}_l}(y)$ and $\mu_{\overline{G}_l}(y)$ are the lower and upper membership grades of $\mu_{\overline{G}_l}(y)$.

(c) suppose that $N$ of the $M$ rules in the FLS are fired, where $N \leq M$, and the combined output type-1 fuzzy set is obtained by combining the fired output consequent sets, i.e., $\mu_{\overline{B}}(y) = \cup_{i=1}^{N} \mu_{\overline{B}_i}(y), \quad y \in Y$, then

$$\mu_{\overline{B}}(y) = \int_{y \in [\overline{f}_1 \star \cdots \star \overline{f}_N]}^{\overline{f}_1 \star \cdots \star \overline{f}_N} 1/b, \quad y \in Y$$

Based on the Theorem 2.3, it is known the firing level set is an interval set which is determined by its left-most and right-most points, $f_l$ and $f_l$. The fired output consequent set $\mu_{\overline{B}}(y)$ of rule $R_l$ can be obtained from the firing interval \[ \int_{l=1}^{N} \mu_{\overline{B}_l}(y), \quad y \in Y \]
using Equation (2.60) and it is also an interval set.

The next step after fuzzy inference, is type-reduction. The center-of-sets type-reduced set of an interval type-2 FLS is an interval type-1 set and can be expressed as:

\[
Y_{\text{cos}}(X) = [y_l, y_r] = \left[ \sum_{i=1}^{M} f_i^l y_i^l, \sum_{i=1}^{M} f_i^r y_i^r \right]
\]

To compute \( Y_{\text{cos}}(X) \) for interval type-2 output sets, it is sufficient to compute the upper and lower bounds of the type-reduced set. Let the centroids of the interval type-2 consequent set \( \tilde{G}^i \) be described bounded as \([y_i^l, y_i^r] \) \((i = 1, \cdots, M)\),

\[
C_{\tilde{G}^i} = \left[ \sum_{\theta_1 \in J_{\theta_1}} \cdots \sum_{\theta_N \in J_{\theta_N}} 1 / \sum_{i=1}^{N} y_i \theta_i \sum_{i=1}^{N} \theta_i = [y_i^l, y_i^r] \right]
\]

Let \( f_i^l \) be the firing level of the \( i \)th rule associated with the lower bound \( y_i \) of the type-reduced set; \( f_i^r \) be the firing level of the \( i \)th rule associated with the upper bound \( y_i \) of the type-reduced set. Then, the left and right bounds of the type-reduced set can be expressed as:

\[
y_l = \frac{\sum_{i=1}^{M} f_i^l y_i^l}{\sum_{i=1}^{M} f_i^l} \quad (2.64a)
\]
\[
y_r = \frac{\sum_{i=1}^{M} f_i^r y_i^r}{\sum_{i=1}^{M} f_i^r} \quad (2.64b)
\]

where both \( f_i^l \) and \( f_i^r \) are either one of \( f_i \) and \( \bar{f}_i \). Karnik-Mendel Method can be used to determined the values of \( f_i^l \) and \( f_i^r \) and thus the type-reduced set \([y_l, y_r]\).

Because \( Y_{\text{cos}} \) is an interval type-1 set, the defuzzification is just using the average of \( y_l \) and \( y_r \). Hence, the defuzzified output of a interval singleton type-2 FLS is:

\[
y(x) = \frac{y_l + y_r}{2}
\]

(2.65)
Chapter 3

Non-singleton Type-1 Fuzzy Controller for Noise Rejection

Noise is unavoidable, so there will always be some amount of uncertainties in the measured output, and therefore the feedback signal of a closed-loop system. One strategy that a singleton type-1 fuzzy logic system can use to reduce the impact of noise is to use the moving average filter. The feedback signal is processed by the filter before it is passed to the fuzzifier. For a constant signal that is corrupted by noise with constant mean and variance, the filtered signal will be smoother and closer to the true signal if the window length is sufficiently long. Consequently, the input signal of singleton fuzzy controller will also be smoother and the steady state response of the plant will become less oscillatory. Although the only parameter of the non-weighted moving average filter is its window width, it is not easy to design and implement. Properties of the noise, especially its frequency, needs to be identified via experiments in order to select a suitable window length. The moving average filter also requires additional memory space to store the system state data. These two reasons will increase the cost needed to implement the moving average filter. Furthermore, and perhaps more importantly, the moving average filter will slow down the system dynamics. This is not desirable, especially when fast transient response is needed.
Another approach for coping with an uncertain input signal is to replace the singleton fuzzifier with a non-singleton fuzzifier. This technique has been used successfully in various applications[74], but not in control systems. A non-singleton type-1 fuzzy logic system has been briefly introduced in Section 2.3. Its inputs are generally mapped into type-1 triangular/Gaussian fuzzy sets that have maximum membership grade at the value of the crisp input[57]. Hence, a non-singleton fuzzifier assumes that the input value is most likely to be the correct value. However, due to the presence of noise in the data, the adjacent points may also be the correct value, albeit with smaller possibilities[73]. The non-singleton fuzzifier may then be viewed as a prefilter which is similar to a moving average filter. This chapter aims at developing a non-singleton fuzzy logic controller for handling uncertainties caused by the presence of noise. First, the suitability of existing non-singleton fuzzifier for feedback control is assessed. Then, a new non-singleton fuzzifier is proposed in Section 3.3. The fuzzification strategy is designed to have minimal impact on the system dynamics and to reduce the steady-state fluctuations caused by noise. Then, a pH neutralization process between weak acid and strong base is used as a test bed for studying the feasibility of using a non-singleton fuzzifier to reject sensor noise. The pH process is severely nonlinear and thus good pH control is often difficult to achieve. The presence of noise in the feedback signal would make the problem even more challenging. Finally, experiment results are shown using a thermal chamber with its temperature sensor corrupted with noise to verify the noise rejection ability of proposed non-singleton fuzzy controller.

3.1 Properties of Symmetric Triangular Non-singleton Fuzzifier

Non-singleton type-1 fuzzy logic controller is a fuzzy logic system that employs a non-singleton fuzzifier to map the crisp input value to a type-1 fuzzified input set.
The motivation is to capture the effect of noise on the integrity of the feedback signal. Hence, the convention is to map the input point $x$ onto a fuzzy set $X$ which has a spread of $s$ and unity membership grade at $x$. For input values that are further away from $x$, the degree of membership in $X$ becomes smaller. The membership function for $X$, $\mu_X(x)$, indicates that the sensor reading $x$ is the most likely to be the true value, while the adjacent points are also possible but to a lesser degree [57]. The other components of the fuzzy controller, namely the rule-base, the fuzzy inference engine and defuzzifier, remains unchanged. The non-singleton fuzzifier can be viewed as a pre-filter that estimates the true input value in order to account for effect of input uncertainties. Due to this feature, the non-singleton fuzzifier should have properties that are similar to the moving average filter so that the non-singleton fuzzy logic system can serve as an alternative to the singleton fuzzy logic system plus moving average filter.

The motivation of this chapter is to establish the feasibility of using a non-singleton type-1 fuzzy controller to simultaneously provide control and to minimize the impact of noise. The effects of noise is most noticeable when the feedback system is in steady state. During this stage, the actual error may be zero but measurement noise causes a non-zero error signal, and therefore the control signal to fluctuate. To reduce the influence of noise and to maintain the output at the setpoint, changes in the control signal should be minimized when the output is near the set-point. As a start, the effect of a non-singleton fuzzifier that maps a crisp value, $x$, into a triangular type-1 fuzzy set is studied. The objective is to assess the suitability of the non-singleton fuzzifier for use in a fuzzy PI controller to reject noise. The membership function of the fuzzified input, $X$, is defined as:

$$\mu_X(x_i) = \max(0, 1 - \left| \frac{x - x_i}{s} \right|)$$  \hspace{1cm} (3.1)$$

where $x$ is the apex of the triangular fuzzy set with support $2s$ and $s$ is the spread of the symmetrical triangular fuzzy set. As defined in Equation (3.1), the output of
the fuzzifier is a triangular fuzzy set that is symmetrical about \( x \). A fuzzifier that transforms a crisp input into a symmetric set is commonly employed and chosen for analysis because noise is random so the true value is equally likely to be larger or smaller than the measured value. The spread of the fuzzified input set, \( s \), depends on the noise level. A larger spread should be chosen if there are more uncertainties.

When the fuzzification process is completed, the fuzzified input set \( X \) is passed to the inference engine which produces a mapping from the input sets to output sets. As discussed in Section 2.3, the mapping from input to output set is achieved via the sup-star composition. The difference between employing a type-1 fuzzifier in place of a singleton fuzzifier may, therefore, be analyzed by examining the result of the sup-star composition of \( X \) and an antecedent fuzzy set, \( S_1 \). For a general fuzzy system, the three scenarios listed below may occur:

- Support of \( X \) partially overlaps the support of \( S_1 \)
- Support of \( X \) is a subset of the support of \( S_1 \)
- Support of \( S_1 \) is a subset of \( X \)

### 3.1.1 Case I: Support of \( X \) partially overlaps the support of \( S_1 \)

This subsection presents the results of first case where the support of the type-1 fuzzified input set partially intersects the base of antecedent fuzzy set \( S_1 \). As shown in Table 3.1, the membership function of the antecedent fuzzy set \( S_1 \) is defined as a triangle with vertexes at \((a, 0)\), \((c, 1)\) and \((b, 0)\). Let \( \mu_{x_k}(x_k) \) be the membership function of the fuzzified input set and \( \mu_{S_1}(x_k) \) be the membership function for antecedent set \( S_1 \). Using product t-norm, the sup-star composition is the supremum of \( \mu_Q \equiv \mu_{x_k}(x_k) \times \mu_{S_1}(x_k) \) (refer to Equation (2.20)). It is obvious that \( x_{k,\text{max}} \) in Equation (2.20) is inside \([x, c]\) or \([c, x]\). Besides the two possible situations where \( x < c \) or \( x \geq c \), there are also two other possible situations where the left vertex
Table 3.1. Partial overlap between the input and antecedent $S1$ sets

<table>
<thead>
<tr>
<th>Diagram</th>
<th>Expression</th>
<th>Condition</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>$x_{k,\text{max}} = \frac{x + s + a}{2}$, (if $\frac{x + s + a}{2} &lt; c$)</td>
<td>$\mu Q(x_{k,\text{max}}) = \frac{2}{s(c-a)} - \frac{(x+s)a}{s}$</td>
<td></td>
</tr>
<tr>
<td>(b)</td>
<td>$x_{k,\text{max}} = \frac{x - s + b}{2}$, (if $\frac{x - s + b}{2} &gt; c$)</td>
<td>$\mu Q(x_{k,\text{max}}) = \frac{4}{s(b-c)} - \frac{(x-s)b}{s}$</td>
<td></td>
</tr>
<tr>
<td>(c)</td>
<td>$x_{k,\text{max}} = c$, (if $\frac{x - s + b}{2} \leq c$)</td>
<td>$\mu Q(x_{k,\text{max}}) = \frac{c - x + s}{s}$</td>
<td></td>
</tr>
<tr>
<td>(d)</td>
<td>$x_{k,\text{max}} = x$, $\mu Q(x_{k,\text{max}}) = \frac{b - x}{b - c}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$x_{k,\text{max}} = x$, $\mu Q(x_{k,\text{max}}) = \frac{x - a}{c - a}$
a or right vertex \( b \) of \( S_1 \) set is inside the base of input set. Hence, Case I can be divided into four situations as shown in Table 3.1.

Consider Case I(a) in Table 3.1, where \( x < c, x - s < a < x + s < b \) and \( x_{k,max} \) is inside \([x, c]\). The membership functions for \( \mu_{x_k}(x_k) \) and \( \mu_{S_1}(x_k) \) in the range of \([x, c]\) are:

\[
\mu_{x_k}(x_k) = -\frac{x_k - x - s}{s}, \quad x_k \in [x, c] \tag{3.2}
\]

\[
\mu_{S_1}(x_k) = \frac{x_k - a}{c - a}, \quad x_k \in [x, c] \tag{3.3}
\]

Using the product t-norm, the sup-star composition needs \( \mu_{x_k}(x_k) \times \mu_{S_1}(x_k) \):

\[
\mu_Q(x_k) = \mu_{x_k}(x_k) \times \mu_{S_1}(x_k) = -\frac{x_k^2 - (x + s + a)x_k + (x + s)a}{s(c - a)} \quad x_k \in [x, c]
\]

To calculate the maximum value of \( \mu_Q(x_k) \), its derivative is needed to search for the tuning point:

\[
\frac{\partial \mu_Q(x_k)}{\partial x_k} = -\frac{2x_k - (x + s + a)}{s(c - a)}, \quad \frac{\partial^2 \mu_Q(x_k)}{\partial x_k^2} = -2
\]

Let \( \frac{\partial \mu_Q(x_k)}{\partial x_k} = 0 \), and rearranging

\[
x_{k,max} = \frac{x + s + a}{2}, \tag{3.4}
\]

\[
\mu_Q(x_{k,max}) = \frac{(x + s + a)^2}{4} - \frac{(x + s)a}{s(c - a)}
\]

In case I(a), \( x - s < a \) and \( a < x + s < b \). Hence, it is always true that:

\[
x - s < a \Rightarrow x < \frac{x + s + a}{2}
\]

Since the \( x_{k,max} \) must be within \([x, c]\), there are two situations for \( x_{k,max} \) depending
on the right bound \( c \) and the actual firing level should be:

\[
\begin{align*}
\text{(1)} \quad x_{k,\text{max}} &= \frac{x + s + a}{2}, \quad (\text{if } \frac{x + s + a}{2} < c) \\
\mu_Q(x_{k,\text{max}}) &= \frac{4}{s(c - a)} - \frac{(x + s)a}{(x + s + a)^2} \\
\text{(2)} \quad x_{k,\text{max}} &= c, \quad (\text{if } \frac{x + s + a}{2} \geq c) \\
\mu_Q(x_{k,\text{max}}) &= -\frac{c - x + s}{s}
\end{align*}
\]  

(3.5)

Since the derivation procedure is similar, the results for Case I(b), I(c) and I(d) are summarized in Table 3.1. It shows all the possible situations of firing level and corresponding \( x_{k,\text{max}} \) when only partial base of the type-1 fuzzified input set overlaps the base of fuzzy set \( S_1 \). Note that Case I(c) and I(d) show that the effect of the non-singleton fuzzifier is equivalent to a singleton fuzzifier under such situations. The point \( x_{k,\text{max}} = \frac{x + s + a}{2} \) for Case I(a) is actually midpoint between right vertex of type 1 fuzzified input set \( x + s \) and left vertex of fuzzy set \( S_1, a \); while \( x_{k,\text{max}} = \frac{x - s + b}{2} \) for Case I(b) is actually midpoint between left vertex of type-1 fuzzified input set \( x - s \) and right vertex of fuzzy set \( S_1, b \).

3.1.2 Case II: Support of \( X \) is a subset of the support of \( S_1 \)

In this case, \( a < x - s < x + s < b \), which means the spread \( s \) is no larger than \( \frac{b - a}{2} \). It is also obvious that \( x_{k,\text{max}} \) is inside \([x, c]\) or \([c, x]\). Using similar analysis, the equations in Table 3.2 may be obtained. There are only two possible situations \((x < c \text{ or } x \geq c)\) of firing level and corresponding \( x_{k,\text{max}} \) when the base of the type-1 fuzzified input set is strictly within the base of fuzzy set \( S_1 \). The results demonstrate that the effect of non-singleton fuzzifier is also just no more than a singleton fuzzifier under such situations.
Table 3.2. Fuzzified input base is a subset of $S_1$ base

<table>
<thead>
<tr>
<th>Case</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>$x_{k,\text{max}} = x$, $\mu_Q(x_{k,\text{max}}) = \frac{x - a}{c - a}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(b)</td>
<td>$x_{k,\text{max}} = x$, $\mu_Q(x_{k,\text{max}}) = \frac{b - x}{b - c}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 3.1.3 Case III: Support of $S_1$ is a subset of $X$

Table 3.3 shows the third case when the base of antecedent fuzzy set $S_1$ is a subset of the base of fuzzified input set, i.e. $x - s \leq a < b \leq x + s$ and the spread $s$ is no less than $\frac{b - a}{2}$. It is also obvious that $x_{k,\text{max}}$ is inside $[x, c]$ or $[c, x]$. Case III(a) shows the two possible situations and the corresponding $x_{k,\text{max}}$ when the base of fuzzy set $S_1$ is a subset of the base of fuzzified input set. Case III(b) has the other two possible situations. The two equations when $x_{k,\text{max}} = c$ imply that when spread $s$ is large enough, then $x_{k,\text{max}} = c$ results and $\lim_{s \to \infty} \mu_Q(c) = -\frac{c - c - s}{s} = 1$ or $\lim_{s \to \infty} \mu_Q(c) = \frac{c - c + s}{s} = 1$.

The derivations in this subsection show that the non-singleton fuzzifier effectively reduces to a singleton fuzzifier when the spread is chosen as a small value, such that the base of fuzzified input set may become a subset of bases of antecedents. In order for the effect of the non-singleton fuzzifier to differ from that of the singleton fuzzifier when firing the antecedent fuzzy sets, a relatively large value of spread is needed.
### 3.2 Non-singleton Type-1 PI Fuzzy Controller

#### 3.2.1 Structure of non-singleton PI controller

Section 2.2 has shown that an interesting property of a type-1 fuzzy logic controller is that it may be used to realize a PID controller [68]. Fuzzy and PID controllers are equivalent when the fuzzy inference engine employs the product t-norm and height defuzzification is used to defuzzify the output set. For ease of design and to provide a base for comparing against a PID control system that employs moving average filter, the structure of the singleton controller and non-singleton controller studied herein are designed such that both are equivalent to the same PI controller. As shown in Figure 3.1, the input signals for the fuzzy controllers are the error signal, $e$ and the derivative of error, $\dot{e}$, while the output of such PD-like FLSs is the increment of the actual controller output($\Delta u$). An integrator is cascaded to the PD FLS to obtain a PI-like controller.
3.2.2 Structure of inference engine

Figure 3.2 shows the antecedent fuzzy sets of the fuzzy PD-like FLS. They are designed to be symmetrical about the zero point. Five fuzzy sets are used to partition the error domain. The three vertexes for the NS, O and PS fuzzy sets are \([-ep, -es, 0]\), \([-es, 0, es]\) and \([0, es, ep]\) respectively. As shown in [68], a singleton type-1 fuzzy controller is equivalent to a particular PID controller regardless of the number of fuzzy sets that are used to characterize the input universes of discourse. There is, therefore, no need for a singleton type-1 fuzzy controller to have more than two fuzzy sets for each input domain. This property does not carry over to a non-singleton type-1 fuzzy controller. As the non-singleton fuzzifier converts the crisp input into a type-1 fuzzy set, there is a chance that the fuzzified input may activate more than two antecedent fuzzy sets in one input domain. The number of fired antecedent sets depends on the relative support of the fuzzy sets, and consequently on the number of fuzzy sets used to partition the input domain. This chapter aims at investigating the noise rejection ability of non-singleton type-1 fuzzy controllers, particularly at steady-state. Hence, it is useful if the fuzzy rules fired by the output of the non-singleton fuzzifier during transient and steady-state are different. This is why five fuzzy sets are used to partition the error domain.

Three antecedent sets are used to describe the input domain for \(\dot{e}\). The consequent singletons are set up based on the equivalent proportional gain \(K_p\) and derivative gain \(K_D\) using the method for setting up the equivalent fuzzy PID controller in Section 2.2. Table 3.4 shows the consequent singletons of the fuzzy PD FLS. In the following section, the traditional non-singleton type-1 fuzzy controller
will be formally investigated.

![Figure 3.2. The antecedents of PD-like FLSs](image)

Table 3.4. Consequent singletons of the fuzzy PD FLS

<table>
<thead>
<tr>
<th>NB</th>
<th>NS</th>
<th>O</th>
<th>PS</th>
<th>PB</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>$-K_P \cdot e_p - K_D \cdot d_e$</td>
<td>$-K_P \cdot e_s - K_D \cdot d_e$</td>
<td>$-K_D \cdot d_e$</td>
<td>$K_P \cdot e_s - K_D \cdot d_e$</td>
</tr>
<tr>
<td>$O_{de}$</td>
<td>$-K_P \cdot e_p$</td>
<td>$-K_P \cdot e_s$</td>
<td>$0$</td>
<td>$K_P \cdot e_s$</td>
</tr>
<tr>
<td>$P$</td>
<td>$-K_P \cdot e_p + K_D \cdot d_e$</td>
<td>$-K_P \cdot e_s + K_D \cdot d_e$</td>
<td>$K_D \cdot d_e$</td>
<td>$K_P \cdot e_s + K_D \cdot d_e$</td>
</tr>
</tbody>
</table>

### 3.2.3 Characteristics of fuzzy PI controller using symmetric non-singleton fuzzifier

By leveraging on the analysis in Section 3.1, the implication of employing a non-singleton fuzzifier on the output or control surface of a fuzzy PI controller will be discussed here. The fuzzy sets PB and NB describe the transient performance while the fuzzy sets PS and NS mainly describe the steady state performance. The value of the error domain parameter $e_s$ (see Figure 3.2) is chosen so that at steady state, only three antecedent fuzzy sets (NS, O and PS) are fired. The output of the non-singleton fuzzifier is a type-1 triangular fuzzified set defined in Equation (3.1).
the non-singleton fuzzifier aims at modelling the possibility that the points adjacent to the input error variable $x_e$ may also be the correct value, the spread $s$ should be selected to reflect the variance of the noise signal. Since the variance of the noise signal at steady state is normally small compared to the base of the fuzzy sets, the value of $x_e$ and spread $s$ are very small compared to $es$ and $ep$.

Figure 3.3 shows the symmetric triangular non-singleton fuzzifier defined in Equation (3.1) and the three antecedent fuzzy sets that are likely to be activated during the steady state period. During steady state, the base of the input set is a subset of antecedent fuzzy set O. The input error $x_e$ is mapped into $x^{max}_O$ and the firing level is actually case II(a) in Table 3.2 as:

$$f^{max}_O = \frac{x_e + es}{es}, \quad x^{max}_O = x_e$$

The non-singleton fuzzifier has the same effect as the singleton fuzzifier on firing the antecedent fuzzy set O.

![Figure 3.3. Triangular non-singleton fuzzifier with small spread for $e$](image)

Most of the time, the base of input set partially intersects the antecedent fuzzy sets NS and PS during steady state. Hence, if the inputs fires NS and PS according to Case I(c) and I(d), the non-singleton fuzzification for the input error domain will always be equivalent to singleton fuzzification. It is, therefore, sufficient to investigate the other situations where antecedent sets NS and PS are fired as in Case I(a) or I(b). Suppose the feedback error, $x_e$, is negative as in Figure 3.3. For
the antecedent fuzzy set NS, the input error $x_e$ is mapped into $x_{NS}^{\text{max}}$ and the firing level is case I(b) in Table 3.1 as:

$$f_{NS}^{\text{max}} = \frac{(x_e - s)^2}{4 \times e_s \times s}, \quad x_{NS}^{\text{max}} = \frac{x_e - s}{2}$$

For the antecedent fuzzy set PS, the non-singleton fuzzifier will map $x_e$ into $x_{PS}^{\text{max}}$ and the firing level is actually case I(a) in Table 3.1 as:

$$f_{PS}^{\text{max}} = \frac{(x_e + s)^2}{4 \times e_s \times s}, \quad x_{PS}^{\text{max}} = \frac{x_e + s}{2}$$

The input for $\dot{e}$ is denoted as $x_{de}$ and the spread of its fuzzifier based on noise level is much smaller than the parameter $de$ of antecedent as in Figure 3.4. The property of $\dot{e}$ is similar to the signal of $e$ at the steady state. If there is no noise, all the desired values for both $\dot{e}$ and $e$ should be zero. However, when the inputs are corrupted with noise, the input $\dot{e}$ may fire all the three sets at steady state. Hence, using similar derivation process, the membership grades of $x_{de}$ in the fuzzy sets for the $\dot{e}$ domain are:

$$f_{P}^{\text{max}} = \frac{(x_{de} + s)^2}{4 \times de \times s}, \quad f_{O_{de}}^{\text{max}} = \frac{x_{de} + de}{de}, \quad f_{N}^{\text{max}} = \frac{(x_{de} - s)^2}{4 \times de \times s} \quad (3.6)$$

As the output approaches steady state, 9 rules are generally activated and thus the
output of the controller can be calculated as:

\[
Y(x, x_{de}) = K_P \times e s \{(f_{PS}^{max} - f_{NS}^{max})(f_P^{max} + f_{O,de}^{max} + f_N^{max})\} + \\
K_D \times de \{(f_P^{max} - f_N^{max})(f_{PS}^{max} + f_{O,de}^{max} + f_N^{max})\} \\
\frac{(f_P^{max} + f_{O,de}^{max} + f_N^{max})}{(f_{PS}^{max} + f_{O,de}^{max} + f_N^{max})} (3.7)
\]

The sum of all the firing levels for input \(e\) using non-singleton fuzzification is:

\[
f_{PS}^{max} + f_{O}^{max} + f_{NS}^{max} = \frac{2(x_e + s)^2 + 4 \times es \times s}{4 \times es \times s}
\]

Generally, the steady state values of the error signal \(x_e\) and the spread \(s\) that is chosen based on the noise level are much smaller than \(es\). Hence, the term of \(2(x_e + s)^2\) may be negligible compared to the term of \((4 \times es \times s)\). Hence, the sum of all firing levels at the steady state may be approximated as:

\[
f_{PS}^{max} + f_{O}^{max} + f_{NS}^{max} \approx \frac{4 \times es \times s}{4 \times es \times s} = 1 \quad (3.8)
\]

The sum of the firing levels for input \(\dot{e}\) may also be approximated as:

\[
f_P^{max} + f_{O,de}^{max} + f_N^{max} = \frac{2(x_{de} + s)^2 + 4 \times de \times s}{4 \times de \times s} \approx \frac{4 \times de \times s}{4 \times de \times s} = 1 \quad (3.9)
\]

Hence, the output may be approximated as:

\[
Y(x, x_{de}) = \frac{K_P \times e s \{(f_{PS}^{max} - f_{NS}^{max})(f_P^{max} + f_{O,de}^{max} + f_N^{max})\} + \\
K_D \times de \{(f_P^{max} - f_N^{max})(f_{PS}^{max} + f_{O,de}^{max} + f_N^{max})\} \\
\frac{(f_P^{max} + f_{O,de}^{max} + f_N^{max})}{(f_{PS}^{max} + f_{O,de}^{max} + f_N^{max})}} (3.10)
\]

\[
\approx K_P \times e s \times \frac{x}{es} \times 1 + K_D \times de \times \frac{x_{de}}{de} \times 1 \\
= K_P \times x + K_D \times x_{de}
\]

Equation (3.10) means the output of this non-singleton fuzzy PD-like FLS during the steady state period is very similar to the output of singleton fuzzy PD-like FLS. That means when the spread chosen based on the noise level is small, then it will give rise to results that are very similar to singleton fuzzy controllers.
Since the normal noise level in system is usually much less than the initial error, the value of \( ep \) is much bigger than that of \( es \) which can be chosen based on noise level. When only three antecedent sets are fired during steady state period as shown in Figure 3.3, it holds that \(-es \leq x_e - s \leq 0 \leq x_e + s \leq es\). From the discussion in Section 3.1, the largest difference between the firing level of the singleton and non-singleton type-1 FLSs occurs when the spread \( s \) is large such that \( x_e - s \leq -es < x < es \leq x_e + s \). This is Case III in Section 3.1 where the supports of the antecedent fuzzy sets are all smaller than the base of fuzzified input set as shown in Figure 3.5.

To examine the behavior of such a FLS, consider once again a crisp input \( x_e \) that is negative. For the fuzzy set NS, the input error \( x_e \) is mapped into \( x_{NS}^{max} \). The mathematical expression for \( x_{NS}^{max} \) and the resulting firing level is Case III(b) in Table 3.3:

\[
 f_{NS}^{max} = \frac{-es - x_e + s}{s} = 1 - \frac{es}{s} - \frac{x_e}{s}, \quad x_{NS}^{max} = -es
\]

Similarly, the mapping point and the firing level for the fuzzy set O are as the Case III(a) in Table 3.3:

\[
 f_{O}^{max} = \frac{x_e + s}{s} = 1 + \frac{x_e}{s}, \quad x_{O}^{max} = 0
\]

The mapping point and the firing level for the fuzzy set PS are as the Case III(a) in Table 3.3:

\[
 f_{PS}^{max} = \frac{-es + x_e + s}{s} = 1 - \frac{es}{s} + \frac{x_e}{s}, \quad x_{PS}^{max} = es
\]

Since the absolute value of the noisy error signal is much smaller than the spread \( s \) (\(|x_e| << s\)), this means \( f_{NS}^{max} \approx f_{PS}^{max} \approx 1 - \frac{es}{s} \). For similar reasons, the firing levels for antecedents NB and PB are nearly the same (\( f_{NB}^{max} \approx f_{PB}^{max} \)). This property coupled with the fact that the consequent sets are symmetric about zero means that the contributions of the rule containing NS will be approximately cancelled by the rule associated with PS. Hence, the contribution of the rule with the fuzzy set O in the antecedent will dominate and the change in the control signal is minimized.
Chapter 3. Non-singleton Type-1 Fuzzy Controller for Noise Rejection

Similarly, the spread for $\dot{e}$ is also chosen large in order that the base of fuzzified input set covers all other bases of antecedents of $\dot{e}$ as in the lower part of Figure 3.4. Hence, at the steady state when the absolute value of the noisy $\dot{e}$ signal is normally a small value around zero, the firing levels for antecedents N and P are also nearly the same ($f_{N}^{max} \approx f_{P}^{max}$) and the contribution of the fuzzy set O$_{de}$ will dominate.

As a result, from the Equation 3.10, when large spread is used, the output of such fuzzy PD-like FLS will approximately be zero:

$$Y(x_e, x_{de}) \approx 0$$

(3.11)

This means that the control output of a non-singleton PI FLS that employs a larger spread may be less affected by the noise resulting in a control action increment $\Delta u$ that is approximating 0. After the subsequent integrator, the output of the fuzzy PI controller will remain approximately constant and the response of the whole system thus may greatly reduce the impact from the noisy signals. Although a large spread $s$ will be better able to reject the undesirable impact of noise, the disadvantage is that there will no longer be a link between spread and noise level. Consequently, it
would be difficult to use information about noise variance as a basis for selecting a suitable spread $s$.

Insights from the analysis in this subsection indicate that changes in control signal when the output is near the set-point can be minimized by firing two antecedents, NS and PS (or N and P), at similar level to obtain a balanced output. This reduces the fluctuations in the control action which can be also regarded as neglecting the information of noisy error. In the next section, a new non-singleton fuzzifier based on this idea is proposed.

### 3.3 Non-symmetric non-singleton Fuzzifier

The proposed non-singleton fuzzifier, referred to as the nonsymmetric non-singleton fuzzifier is defined in this Section. The objective is to design a non-singleton fuzzy controller that reduces the influence of noisy feedback signal on performance and, at the same time, is easy to design. The proposed fuzzifier can be regarded as an interval type-1 set as defined in Equation (3.12). Denoting the measured input error as $x_e$, the lower bound of the proposed non-singleton fuzzifier is defined as $V_l = (x_e - v1)$ and the upper bound of the interval type-1 set is $V_r = (x_e - v2)$. The values of the spread $v1$ and $v2$ are chosen as in Equation (3.12). Figure 3.6 shows how the interval type-1 non-singleton fuzzifier fires the antecedents fuzzy sets that determine the steady state behavior.

$$
\begin{cases}
V_l = x_e - v1, & V_r = x_e + v2, & \max\{v1, v2\} \leq B_o \\
v1 = v, & v2 = v - \alpha_f \times x_e & (x_e \leq 0) \\
v1 = v + \alpha_f \times x_e, & v2 = v & (x_e > 0)
\end{cases}
$$

(3.12)

The proposed non-singleton fuzzifier is “non-symmetric” because the values of $v1$ and $v2$ are generally different. The value of $v$ can be chosen based on the noise level of the system. The reason for why different values are defined for $v1$ and $v2$ based on sign of $x_e$ is to make sure the new fuzzifier always fires the two antecedents
Figure 3.6. Rectangular nonsymmetric non-singleton fuzzifier

NS and PS at the same time, that is, \((x_e - v1) < 0\) and \((x_e - v2) > 0\) are always true. When this condition is satisfied, the firing level of the antecedent fuzzy set \(O\) will be close or equal to unity grade, signifying that there is full confidence that the error is zero. The non-singleton fuzzifier will also activate NS and PS at the similar level when \(\alpha_f\) is close to 2. \(B_v\) is the bound for \(v1\) and \(v2\) and its value is also set up based on noise level. Since it is not desirable to reduce the impact of input signal error during the transient period, the bound \(B_v\) is chosen to roughly distinguish the period of transience and steady state and thus the antecedent sets may not be activated at similar level during the transience.

In summary, the nonsymmetric non-singleton fuzzifier has two key properties:

(i) the crisp input is mapped onto an interval type-1 fuzzy set, and (ii) the crisp input value is not placed at the center of the interval type-1 fuzzy set. The decision to map the crisp input to an interval type-1 set may be attributed to the assumption that noise is a uniform random number within certain range. All values within the noise variance are equally likely to be the true value. Hence, it is reasonable for the values adjacent to measured output value to have the same possibility of being real uncorrupted output value. The second characteristic of the non-singleton fuzzifier is designed to activate equally the antecedent fuzzy sets which are symmetric about the zero point in order to minimize fluctuations in the control action.

Since the consequent fuzzy sets are symmetrical about the zero point, the con-
tribution of the NS rule will be negated by the PS rule so the output of such a fuzzy proportional controller will depend only on the O rule. The aim of noise rejection is achieved by neglecting information about the noisy input signal when the output is near the setpoint. However, choosing a value exactly equal to 2 for $\alpha_f$ may cause steady state error since the control output stabilizes regardless of the noisy error signal. Hence, choosing a slightly value less than 2 for $\alpha_f$ will be a good compromise between performance and noise cancellation.

3.4 Simulation Results

In this section, the performance of non-singleton controller with nonsymmetric fuzzifier is shown and analyzed using the pH neutralization process between weak acid and strong base. The non-singleton controller is used for noise rejection and to control the pH value.

3.4.1 pH process in CSTR

The reagents in the pH neutralization reaction in this chapter are two monoprotic reagents: a weak acid acetic acid $CH_3COOH$ and a strong base $NaOH$. The reason is that the pH neutralization process may be modelled by a Wiener structure if a strong base and a weak acid are used. The strength of an reagent is classified according to the fraction of molecules that dissociate, or the degree of dissociation. An strong reagent is one which dissociate completely in water, while a weak one only partially dissociates. Consider the weak acetic acid, an equilibrium is set up between undissociated molecules $CH_3COOH$ and the ions $CH_3COO^-$ and $H^+$ in water:

$$CH_3COOH \rightleftharpoons CH_3COO^- + H^+$$

At equilibrium, the acid dissociation constant $K_a$ denotes the strength of the acetic acid as:
\[ K_a = \frac{[CH_3COO^-][H^+]}{[CH_3COOH]} \]

In the special case for water at 25°C, the product of the hydrogen ions and hydroxide ions is equal to \(10^{-14} \text{mol}^2 \text{dm}^{-6}\). This product is known as the ionic product of water \(K_w\).

\[ K_w = [H^+][OH^-] = 10^{-14}, \quad pK_w = 14 \]

The weak acid acetic acid \(CH_3COOH\) reacts with the strong base \(NaOH\) to produce the following equilibrium:

\[ CH_3COOH + NaOH \rightleftharpoons CH_3COONa + H_2O \]

As the ionic charges must balance, the following equation should hold:

\[ [Na^+] + [H^+] = [CH_3COO^-] + [OH^-] \]

By expressing \(x_b = [Na^+]\) as the ionic base concentration, \(x_a = [CH_3COO^-] + [CH_3COOH]\) as the ionic acid concentration, the following titration relation can be derived:

\[ x_b + [H^+] = \frac{K_ax_a}{[H^+] + K_a} + [OH^-] \quad (3.13) \]

\[ x_b = \frac{x_a}{1 + 10^{4.75-pH} - 10^{-pH} + 10^{pH-14}} \quad (3.14) \]

Figure 3.7 shows graphically the titration relation for the reaction between \(CH_3COOH\) and \(NaOH\). As seen from the figure, the titration curve is an 'S' shaped curve. Most of the pH curve is relatively flat while the portion around the equivalence point is extremely steep where the concentrations of acidic solution \(x_a\) and basic solution \(x_b\) are equal. This extremely steep gradient causes the pH process to be extremely sensitive to variations. This characteristic is the reason why pH control is very difficult. Hence, the control valves need to have great rangeability to effectively bring about the required change in pH value. The extreme variation in the gain of static pH process also raises difficult stability and performance issues.
The pH neutralization process considered in this chapter is assumed to take place in a Continuously Stirred Tank Reactor (CSTR) [49], as shown in Figure 3.8. It consists of an influent acetic acid $CH_3COOH$ as the process stream, an influent base $NaOH$ as the titrating stream to maintain the solution volume in the tank as a constant. The dynamics for the mixing process may be described by the following set of bilinear differential equations:

\[
\begin{align*}
V \frac{dx_a}{dt} &= F_a C_a - (F_a + F_b)x_a \\
V \frac{dx_b}{dt} &= F_b C_b - (F_a + F_b)x_b
\end{align*}
\]  

(3.15) 

(3.16)

where $F_a$ (liter/min) is the flow rate of the influent stream, $F_b$ (liter/min) is the flow rate of the titrating stream, $C_a$ (gm-mol/liter) is the concentration of the influent stream, $C_b$ (gm-mol/liter) is the concentration of the titrating stream, $x_a$ (gm-mol/liter) is the concentration of the acidic solution, $x_b$ (gm-mol/liter) is the concentration of the basic solution and $V$ is the volume of the mixture in the CSTR.
The dynamic equations describe how the concentration of the acidic and basic components \( x_a \) and \( x_b \) change subject to the input streams \( F_a \) and \( F_b \).

\[
F_a \cdot C_a + F_b \cdot C_b = F_a + F_b, \quad x_a, x_b
\]

Figure 3.8. The CSTR configuration with two influent streams

The mathematical model for the pH process in a CSTR is a combination of the CSTR dynamics and the static titration equation. From Equation (3.15) and (3.16), it can be seen that the CSTR dynamics is a set of bilinear equations. The dynamics is essentially linear if the assumption \( F_a \gg F_b \) holds. This assumption is reasonable because in most situations the reagents used in the titrating stream can be adjusted for concentration in such a way that the relation holds. Hence, the CSTR is essentially linear and governed by Equation (3.16). Hence, the pH neutralization process in a CSTR can be separated into two parts: one is an almost linear dynamics and the other is a static nonlinear titration relation. Thus the system approximates a Wiener model.

### 3.4.2 Performance of proposed controller

As the pH process may be modelled by a Wiener structure, the control problem can be simplified by eliminating the static non-linearity. The linearized process can then be placed under the control of linear controllers, whose behavior are well understood. Figure 3.9 shows the architecture that may be used to control plants
which are described by Wiener-type non-linear models. The inverse static block in
the figure is the inverse of the neutralization equation defined in Equation (3.14).
Once the output pH value has been transformed into an estimate of the basic ion
concentration ($\hat{x}_b$), pH control can be achieved by employing the non-singleton type-
1 fuzzy controller to regulate the state, $x_b$, and to reject disturbances caused by
changes in the acid flowrate and concentration. The input error of the controller
is the difference between the desired concentration of the basic solution $x_b$ and
estimated concentration of the basic solution $\hat{x}_b$ ($e = x_b - \hat{x}_b$). The output of the
controller is the flow rate of the titrating stream $F_b$. A uniform random number is
added to the pH value to simulate the effect of measurement noise on the integrity
of the feedback system.

![Figure 3.9. The control scheme for CSTR](image)

In the simulations, the fuzzy controller is used to control the pH process at the
most sensitive equivalent point (pH=8.5). It is very critical to reduce the effect of
noise at the equivalent point since the gain of static pH process at this point is ex-
tremely large. The initial pH value of the CSTR is at the pH=5. The initial parame-
ters of CSTR are set as: $x_a = 0.0216 (gm - mol/liter)$, $x_b = 0.0138 (gm - mol/liter)$,
$F_b = 0.1279 (liter/min)$, $F_a = 0.8 (liter/min)$, $C_a = 0.025 (gm - mol/liter)$, $C_b =
0.1 (gm - mol/liter)$ and $V = 2 (liter)$.

Table 3.5 shows the parameters of the non-singleton fuzzy PD plus integrator
controller. The inputs are the error $e = x_b - \hat{x}_b$ and its derivative. The sampling
time of the controller is 0.1 second. The equivalent proportional gain is $K_P = 0.76$
and derivative gain is $K_D = 2$. The universe of discourse domain of input $e$ are
chosen as $ep = 0.1$ and $es = 0.05$. The universe of discourse domain of input $\dot{e}$ is chosen as $de = 0.1$.

Table 3.5. Parameters of the nonsymmetric non-singleton fuzzy PD plus integrator controller for pH setpoint at 8.5

<table>
<thead>
<tr>
<th></th>
<th>$K_P$</th>
<th>$K_D$</th>
<th>$\alpha_f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ep$</td>
<td>0.1</td>
<td>$es$</td>
<td>0.05</td>
</tr>
<tr>
<td>$de$</td>
<td>0.1</td>
<td>$v$</td>
<td>$0.00006$</td>
</tr>
<tr>
<td>$B_v$</td>
<td>$0.0002$</td>
<td>$\alpha_f$</td>
<td>1.9</td>
</tr>
</tbody>
</table>

Figure 3.10. The details of $e$ and $\dot{e}$ of the proposed nonsymmetric non-singleton fuzzy PD plus integrator fuzzy controller at the steady state pH=8.5

The measurement noise is assumed to be a uniform distributed number in $[-1, 1]$ at the point of pH measurement. Hence, the noise level after the inverse static equation converts the error signal to $\hat{\eta}_b$ will be mostly within $[-0.00006, 0.00006]$ during the simulation when the setpoint is pH=8.5 as shown in Figure 3.10. A suitable value for $v$ would be $0.00006$. To distinguish the steady state and transient period, the bound $B_v$ for the spread $v_1$ or $v_2$ is chosen as $0.0002$ to balance the performance of the steady state response and transient response. The value of
bound $B_v$ is chosen as slightly bigger than $v$ such that it provides some margin and the proposed non-singleton fuzzifier may be fully utilized to handle the noise. However, the titration relationship or its inverse static dynamic is very nonlinear and the system gain varies at different pH setpoints. This will cause the inverse static equation to convert the noise signal with different gains. As a result, values of $v$ and $B_v$ should be different for different tasks of different desired pH setpoints. For reference, Table 3.6 shows the noise level of error signal at different pH setpoints and recommended values for $v$ and $B_v$. Finally, the nonsymmetric non-singleton fuzzifiers are used both for input $e$ and input $\dot{e}$. The parameter $\alpha_f$ is chosen as 1.9, since it is not desired to neglect all information at the steady state.

Table 3.6. Noise level at different pH setpoint and recommended control parameters

<table>
<thead>
<tr>
<th>pH setpoint</th>
<th>Noise level</th>
<th>$v$</th>
<th>$B_v$</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>[-0.001, 0.001]</td>
<td>0.001</td>
<td>0.003</td>
</tr>
<tr>
<td>8.5</td>
<td>[0.00006, 0.00006]</td>
<td>0.0006</td>
<td>0.0002</td>
</tr>
<tr>
<td>10</td>
<td>[-0.005, 0.005]</td>
<td>0.005</td>
<td>0.015</td>
</tr>
</tbody>
</table>

Figure 3.11. Comparison of singleton type-1 PI controllers with moving average filters and non-singleton fuzzy PD plus integrator controller
Table 3.7. Mean-squared errors and standard deviations of singleton and non-singleton fuzzy PI controllers for noise rejection during steady state period

<table>
<thead>
<tr>
<th>Steady state performance</th>
<th>Means</th>
<th>Mean-squared errors</th>
<th>Standard deviations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Singleton PI with filter</td>
<td>8.4759</td>
<td>0.0727</td>
<td>0.2691</td>
</tr>
<tr>
<td>(window length is 5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Singleton PI with filter</td>
<td>8.4776</td>
<td>0.0559</td>
<td>0.2356</td>
</tr>
<tr>
<td>(window length is 20)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-singleton type-1 PD plus integrator</td>
<td>8.4935</td>
<td>0.0099</td>
<td>0.0919</td>
</tr>
</tbody>
</table>

Figure 3.12. The pH responses of singleton PI controller and proposed non-singleton fuzzy controllers at different setpoints
Performance of the conventional singleton fuzzy PI controllers with moving average filters and proposed non-singleton fuzzy PD plus integrator controller are shown in Figure 3.11. The response of non-singleton fuzzy PD plus integrator controller is the best one which is the least affected by the noise and can reach the desired setpoint 8.5 with no overshoot. On the other hand, when the window width of the moving average filter increases, the filter can neglect some information about the noise and thus the steady-state response will become a bit less oscillatory (scaled partition in Figure 3.11). However, the use of large window width will generate overshoot resulting in undesirable transient response. The large overshoot is likely caused by the filtering effect which slows down the control loop and caused the overshoot. Table 3.7 shows the numerical investigation of noise rejection ability for both singleton fuzzy PI and non-singleton fuzzy controllers. It shows that with the help of the proposed non-singleton fuzzifer, the fuzzy PI controller can achieve better noise rejection performance than using moving average filter, since both the meansquared errors and standard deviations reduce very much. Hence, the proposed non-singleton control scheme will be a good alternative to using moving average filter to reject noise and avoid the undesirable effects of filtering on the transient response.

Figure 3.12 shows the pH responses of proposed non-singleton fuzzy controllers and conventional singleton fuzzy PI controller with moving average filter at different setpoints. Based on the results in Figure 3.11, the window length of moving average filter is set to 10 to compromise between transient and steady state responses. The solid line is the response of nonsymmetric non-singleton fuzzy PD plus integrator controller. The dashed line is the response of singleton fuzzy PI controller with moving average filter. Figure 3.12 shows the fuzzy PD plus integrator controller performs very well no matter what the setpoint is. The response is much less oscillatory than the response of singleton fuzzy PI controller with moving average filter. In addition, the responses of singleton fuzzy PI controller at setpoint 6 and 10 seem to have slight steady state error which may be due to sluggish effect of the filter on
the response. Since the pH process requires the control valves to have great range to effectively bring about the required change in pH value, the smooth response is much preferred.

The design of proposed non-singleton fuzzifier depends on the three parameters, namely $v$, $\alpha_f$ and $B_v$. These three parameters may have different effect on the steady state and transient responses. Hence, it is worthy to investigate their individual effect on the performance. The following study compares different cases of responses when varying certain parameter and at the same time fixing the other two parameters.

![Responses of proposed non-singleton controllers with different $v$ values](image)

Figure 3.13. The responses of proposed non-singleton fuzzy controllers with different $v$

Figure 3.13 shows the responses of the proposed non-singleton fuzzy controllers with three values for $v$, namely $3 \times 10^{-5}$, $6 \times 10^{-5}$ and $1 \times 10^{-4}$, while the value of $\alpha_f$ is fixed at 1.9 and $B_v$ is at 0.0002. The objective is to show how the value of $v$ chosen based on the noise can affect the whole response. It can be seen that changing $v$ does not affect the noise rejection ability much if each value of $v$ can still cover large portion of noisy signal. However, changes in $v$ affect the transient
Figure 3.14. The responses of proposed non-singleton fuzzy controllers with different $\alpha_f$ values.

Figure 3.15. The responses of proposed non-singleton fuzzy controllers with different $B_v$ values.
performance. When the value of $v$ increases, the rise time of the response will increase slightly. The accompanied overshoot when the $v$ increases may be because that the system will have larger margin to consider the response as entering steady state period and thus may not provide enough control effort during the transition between the transience and steady state.

Figure 3.14 shows the responses of the proposed non-singleton fuzzy controllers for different values of $\alpha_f$. It is shown that when $\alpha_f$ increases, the balancing effect on the antecedents to both sides of zero becomes better and thus it can have better noise rejection ability. As shown in the Figure 3.14, the response when $\alpha_f = 1.9$ is the least oscillatory one.

Figure 3.15 shows the responses when the values of $B_v$ vary at 0.0001, 0.0002 and 0.0006. The $B_v$ is supposed to provide a balance between the transient period and steady state period. However, if the $B_v$ is small, it may limit the capability of proposed non-singleton fuzzifier. As in the case of $B_v$ at 0.0001, the possible range of bound $V_l$ or $V_r$ is limited by the small $B_v$ for current noise level and thus the noise rejection performance is not good enough. On the other hand, if $B_v$ is chosen a larger value at 0.0006, it means that it enlarges the margin for steady state period, then the proposed non-singleton fuzzifier may consider the response to enter the steady state earlier. Hence, the response may become a bit sluggish as the fuzzy PD controller is supposed to provide output effort as small as possible during steady state period. The value of $B_v$ at 0.0002 is a good choice between the transient period and steady state period.

### 3.5 Case Study: Thermal chamber

In this section, experimental results that assesses the performance of the proposed non-singleton fuzzy controller as a temperature controller for a thermal chamber corrupted with noise is presented. The test bed, a thermal chamber, is shown in Figure 3.16. The actuator is the lamp (white box on the top) which heats the
inner chamber. A thermocouple, which serves as a measurement sensor, is inside the chamber. The output of the controller is the duty cycle to drive a lamp. The temperature of the thermal chamber is collected through a data acquisition card to the PC. The noisy signal is simulated by adding a uniform random number generated from the PC to the acquired temperature signal. The overall noise level in the noisy temperature signal is about $[-0.1^\circ C, 0.1^\circ C]$. The objective of this study is to examine the noise rejection performance of the proposed non-singleton fuzzy controller on controlling the thermal chamber with noise comparing with the performance of conventional singleton fuzzy controller. The initial temperature of the thermal chamber is 29$^\circ C$, and the desired temperature is set at 31$^\circ C$. After performance system identification on the thermal chamber using step response method, the transfer function of the thermal chamber can be described as:

$$ G(s) = \frac{28e^{-0.3}}{(8s + 1)} $$

Figure 3.16. Diagram of a thermal chamber
Many empirical methods may be used to tune the parameters of a conventional PID controller for a first order system with deadtime. The ITAE setpoint tuning method [79] as defined below is used to choose suitable gains for PI controller:

\[
K_p = \frac{0.586}{K} \left( \frac{T_d}{\tau} \right)^{-0.916} \\
K_i = K_p \times \left[ 1.03 - 0.165 \left( \frac{T_d}{\tau} \right) \right]
\]

Hence, the gains of the fuzzy PD FLS can be calculated as:

\[
K_D = \frac{0.586}{28} \left( \frac{0.3}{8} \right)^{-0.916} = 0.4236 \\
K_P = 0.4236 \times \left[ 1.03 - 0.165 \left( \frac{0.3}{8} \right) \right] = 0.0542
\]

Table 3.8 shows the parameters of the proposed non-singleton fuzzy PD plus integrator controller for the thermal chamber. Based on the noise level during temperature acquisition, the value of \( v \) is chosen to be 0.1. The bound \( B_v \) is then chosen as 0.3 to balance the performance of the steady state response and transient response.

Figure 3.17 shows the responses of proposed non-symmetric non-singleton fuzzy PD plus integrator controller and conventional singleton fuzzy controller. The response of the proposed non-singleton controller is much less oscillatory than the response of singleton controller at the steady state period. The proposed non-singleton controller shows a good noise rejection ability which is caused by its relative less
Chapter 3. Non-singleton Type-1 Fuzzy Controller for Noise Rejection

Figure 3.17. The responses of proposed non-singleton controller and conventional singleton controller

Figure 3.18. Control signals of proposed non-singleton controller and conventional singleton controller
oscillatory control signal as shown in Figure 3.18 at the steady state. It proves that the proposed non-singleton controller may “neglect” noisy error information at the steady state, then relatively reduce the controller gains and produce smoother control signal, and finally provide nice noise rejection ability.

3.6 Conclusions

This chapter presented a new type of non-singleton type-1 fuzzy controller. It provides an alternative to the moving average filter for minimizing the undesired output fluctuations due to noise. Analysis showed that the non-singleton fuzzifier should map the crisp input into a nonsymmetric interval type-1 fuzzy set, rather than a symmetric triangular set when the response is near steady state. The fuzzy PD plus integrator type is shown to be able to handle uncertainty information by neglecting mostly noisy information. The simulation results demonstrated that the dual objectives of rejecting noise and reducing the deterioration in transient response have been achieved. The performance of the proposed non-singleton fuzzifier is mainly determined by three parameters, namely $v$, $\alpha_f$ and $B_v$. Simulations results have shown different controller responses with different values for each parameter. It indicates that parameter $v$ may affect the transient response and increase the rise time when the value of $v$ increase since the fuzzifier will start earlier to “reject” noisy error signal when near the steady state period. The parameter $\alpha_f$ determines the balancing effect on the antecedents to both sides of zero, larger the value of $\alpha_f$, better noise rejection ability of the proposed controller. The parameter $B_v$ is usually supposed to provide a balance between the transient period and steady state period. Finally, experiment results are shown where the proposed non-singleton fuzzy controller is used to control a thermal chamber with noisy feedback temperature sensor signals. It shows the proposed non-singleton fuzzifier can perform a much better noise rejection job than singleton controller.
Chapter 4

Type-2 Fuzzy PI Controller with Adjustable Type-reduced Output

Uncertainty is ubiquitous in real-world applications. Consequently, people are constantly seeking strategies to cope with a plethora of uncertainties in the hope of minimizing the deleterious effects of those uncertainties. Non-singleton fuzzy logic system is suitable for handling uncertain inputs in the system[57]. In order to better accommodate other kinds of uncertainties in the systems, type-2 fuzzy set was introduced by Zadeh[98] in 1975. A type-2 fuzzy set is defined as one that has a fuzzy primary membership function [71] i.e. the membership grade is a fuzzy set in the unit interval [0,1], rather than a point in [0,1]. Interval type-2 fuzzy sets have secondary membership grade that is always equal to unity. An interval type-2 fuzzy sets is generally characterized by the footprint of uncertainty (FOU), which is described by its boundaries – the upper and lower membership functions. FOU is very useful because it provides information about uncertainties in the shape and position of the membership function. Interval type-2 sets play an important role in modeling uncertainties since they are simple and are usually the only practical sets for constructing type-2 fuzzy logic systems [57]. The centroid of an interval type-2 fuzzy set, which is an interval type-1 fuzzy set, is used as a measure of uncertainty in the interval type-2 fuzzy set. Recent research showed that the centroid of an interval
Chapter 4. Type-2 Fuzzy PI Controller with Adjustable Type-reduced Output

Fuzzy set plays a central role in an interval type-2 fuzzy logic system [31][65][95][66]. Basic theories about type-2 fuzzy sets and type-2 fuzzy logic systems are presented in Section 2.4.

As control is one of the most common applications of type-1 fuzzy set theory, there is growing interest in using type-2 fuzzy logic systems for control engineering problems [56][16][83]. Uncertainties in system properties may manifest as process parameters that assume any value within a certain percentage of the nominal values. A controller needs to provide consistent performance in the face of such parametric uncertainties. To handle such uncertainties, this chapter describes a framework for constructing a type-2 controller that is equivalent to a Proportional plus Integral (PI) controller with bounded gains. Due to this relationship, empirical PI tuning rules, such as the Ziegler-Nichols and ITAE setpoint tuning method [79], can be used to set up the type-2 fuzzy controller. Another motivation is to use the structure of the type-2 fuzzy system as constraints to limit the range of control action that is applied to the process, thereby minimizing the possibility of encountering stability problems. However, existing center-of-sets type-reduction (Section 2.4.6) and defuzzification methods impose restrictions on the flexibility of type-2 fuzzy controllers as they give rise to a fixed input-output map. A methodology for adjusting the centroids of the type-2 consequent sets used to calculate the type-reduced output set based on the uncertainties is proposed. The algorithm enables the type-2 fuzzy controller to vary its output and minimize the influence of uncertain process parameters on the control performance.

The rest of the chapter is organized as follows: Leveraging on the knowledge that a PID controller can be realized by a type-1 FLS [68], a type-2 PI controller whose type-reduced output set is associated with proportional and integral gains within pre-defined bounds is described in Section 4.1. Next, a technique that equips the type-2 fuzzy controller with a variable control surface is proposed. The algorithm is based on the centroid of one embedded type-1 set of each consequent type-2 set and uses it to perform type-reduction. Section 4.5 analyzes the performance of
Chapter 4. Type-2 Fuzzy PI Controller with Adjustable Type-reduced Output

the proposed type-2 fuzzy PI controller by using it to control a first order system with dead time (FOSDT) whose parameters may assume random values from the prescribed ranges. Section 4.6 compares the performances of the proposed type-2 fuzzy PI controller and traditionally fuzzy PI gain-schedule controller on the non-linear pH neutralization plant. Section 4.7 shows experiment results and the test bed is also the thermal chamber whose parameters are considered as uncertain because its fan speed can be uncertain. Finally, conclusions are drawn in Section 4.8.

4.1 Realization of Type-2 Fuzzy PI Controller

PID controllers have been realized by fuzzy logic systems since 1995 [68]. A type-2 fuzzy proportional controller [87] that is able to maintain system performance even when the system parameters deviate from their nominal values has been proposed. However, the type-2 fuzzy proportional controller is useful only when the process contains an integrator. The framework will now be extended to a fuzzy PI controller [38] for controlling a larger class of systems. The key idea is to design the type-2 fuzzy system such that its type-reduced output set is related to a range of pre-determined proportional gain as $[K_{pl}, K_{pr}]$ and integral gain as $[K_{il}, K_{ir}]$.

The type-2 fuzzy PI controller is similar to the type-1 fuzzy PI controller shown in Section 2.2. The input signals of the proposed type-2 fuzzy PI controller is the feedback error ($e$) and the derivative of error ($\dot{e}$). To emulate a PI controller, the output signal is the derivative of the control action ($\dot{u}$). The antecedents are all type-1 fuzzy sets while the consequents are interval type-2 fuzzy sets, since the uncertainties are assumed to be only due to the process and not the feedback signal, thus singleton fuzzifier is used. It is also assumed that the sum of membership grades when an input fires the antecedents are always equal to unity and there are always two antecedent fuzzy sets are fired for each input.
The rule-base of the proposed type-2 fuzzy PI controller is as follows:

\[
\text{Rule } k : \quad \text{If } e \text{ is } e_{k1} \text{ and } \dot{e} \text{ is } \dot{e}_{k2}, \text{ then } \dot{u} \text{ is } u_k, \quad k = [1, 2, 3, 4], \quad (4.1)
\]

\[k1 \in \{i, i + 1\}, \quad k2 \in \{j, j + 1\}\]

Since the parameters of the process under control is uncertain, different control action is needed to achieve the desired output state when the input pair is \((e, \dot{e})\). To cope with such uncertainties and to maintain consistent control performance, the type-2 fuzzy PI controller should, therefore, have the ability to produce the control signals needed to drive the different possible processes along the desired trajectory. Hence, the centroid of the type-2 consequent sets, \(u_k\), of the type-2 fuzzy PI controller is designed to be the interval type-1 set \(C^{u_k} = [C^{u_k}_l, C^{u_k}_r]\):
\[ \begin{align*}
\mathcal{C}_{il}^{uk} &= \min \left\{ \text{sign}(e_k)K_{il}, \text{sign}(e_k)K_{ir} \right\} \times |e_k| \\
&\quad + \min \left\{ \text{sign}(\dot{e}_k)K_{pl}, \text{sign}(\dot{e}_k)K_{pr} \right\} \times |\dot{e}_k| \\
\mathcal{C}_{ir}^{uk} &= \max \left\{ \text{sign}(e_k)K_{il}, \text{sign}(e_k)K_{ir} \right\} \times |e_k| \\
&\quad + \max \left\{ \text{sign}(\dot{e}_k)K_{pl}, \text{sign}(\dot{e}_k)K_{pr} \right\} \times |\dot{e}_k| \\
\end{align*} \] 

\[(4.2a)\]

\[(4.2b)\]

\([K_{pl}, K_{pr}]\) and \([K_{il}, K_{ir}]\) are the range of proportional and integral gains which should be determined by the uncertainty of system to maintain certain performance index. Equation (4.2) is defined to provide the maximum possible range of control output values based on the range of proportional and integral gains.

Having defined the structure of the type-2 fuzzy PI controller, the analysis to show that the proposed controller is equivalent to a PI controller with bounded gains is shown as follows. It is sufficient to use only the activated fuzzy partition in the study for either input to describe the fuzzy system, even though each input domain is normally characterized by several type-1 fuzzy sets. Figure 4.1 shows the fuzzy partition \([e_i, e_{i+1}] \times [\dot{e}_j, \dot{e}_{j+1}]\) that appears in the antecedent part of the fuzzy rules. The partition domain of the feedback error \(e\) is \([e_i, e_{i+1}]\), and the partition domain of the derivative of error \(\dot{e}\) is \([\dot{e}_j, \dot{e}_{j+1}]\). The type-1 fuzzy sets, denoted by the labels \(e_i\) and \(e_{i+1}\), satisfy the partition of unity and have unity membership grade at \(e_i\) and \(e_{i+1}\) respectively. The consequent of the \(k\)th rule, denoted as \(u_k\) in Figure 4.1, is an interval type-2 fuzzy sets whose centroid \([\mathcal{C}_{il}^{uk}, \mathcal{C}_{ir}^{uk}]\) is related to pre-determined proportional gain \([K_{pl}, K_{pr}]\) and integral gain \([K_{il}, K_{ir}]\).

Suppose the input signals are \(e\) and \(\dot{e}\) as shown in Figure 4.1. There will be two sets fired each for \(e\) and \(\dot{e}\), thus four rules are fired. The firing grades after
fuzzification for $e$ and $\dot{e}$ are:

\[
\begin{align*}
    f_{e_{i+1}} &= a = \frac{e - e_i}{e_{i+1} - e_i}, \quad f_{e_i} = 1 - a = \frac{e_{i+1} - e}{e_{i+1} - e_i} \\
    f_{\dot{e}_{j+1}} &= b = \frac{\dot{e} - \dot{e}_j}{\dot{e}_{j+1} - \dot{e}_j}, \quad f_{\dot{e}_j} = 1 - b = \frac{\dot{e}_{j+1} - \dot{e}}{\dot{e}_{j+1} - \dot{e}_j}
\end{align*}
\] (4.3)

Using product t-norm, the firing levels for the four rules associated with the fuzzy partition will be:

\[
\begin{align*}
    f_1 &= ab \\
    f_2 &= a(1 - b) \\
    f_3 &= (1 - a)b \\
    f_4 &= (1 - a)(1 - b)
\end{align*}
\] (4.5)

Using the Equation (2.64) to perform center-of-sets type reduction, the upper and lower bounds of the type-reduced output set will be $Y_\dot{u} = [y_l, y_r]$ where $y_l$ and $y_r$ can be obtained using Equation (4.6).

\[
\begin{align*}
    y_l &= \sum_{i=1}^{4} f_i C_{li} \
    &= \sum_{i=1}^{4} \left[ ab \left( \min \{ \text{sign}(e_{i+1})K_{il}, \text{sign}(e_{i+1})K_{ir} \} \right) |e_{i+1}| + \min \{ \text{sign} (\dot{e}_{j+1})K_{pl}, \text{sign} (\dot{e}_{j+1})K_{pr} \} |\dot{e}_{j+1}| \right] \\
    &\quad + \cdots + (1 - a)(1 - b) \left[ \min \{ \text{sign}(e_i)K_{il}, \text{sign}(e_i)K_{ir} \} |e_i| + \min \{ \text{sign}(\dot{e}_j)K_{pl}, \text{sign}(\dot{e}_j)K_{pr} \} |\dot{e}_j| \right] \\
    &\quad + \cdots + (1 - a)(1 - b) \left[ \min \{ \text{sign}(e_i)K_{il}, \text{sign}(e_i)K_{ir} \} |e_i| + \min \{ \text{sign}(\dot{e}_j)K_{pl}, \text{sign}(\dot{e}_j)K_{pr} \} |\dot{e}_j| \right]
\end{align*}
\] (4.6a)

\[
\begin{align*}
    y_r &= \sum_{i=1}^{4} f_i C_{ri} \\
    &= \sum_{i=1}^{4} \left[ ab \left( \max \{ \text{sign}(e_{i+1})K_{il}, \text{sign}(e_{i+1})K_{ir} \} \right) |e_{i+1}| + \max \{ \text{sign}(\dot{e}_{j+1})K_{pl}, \text{sign}(\dot{e}_{j+1})K_{pr} \} |\dot{e}_{j+1}| \right] \\
    &\quad + \cdots + (1 - a)(1 - b) \left[ \max \{ \text{sign}(e_i)K_{il}, \text{sign}(e_i)K_{ir} \} |e_i| + \max \{ \text{sign}(\dot{e}_j)K_{pl}, \text{sign}(\dot{e}_j)K_{pr} \} |\dot{e}_j| \right] \\
    &\quad + \cdots + (1 - a)(1 - b) \left[ \max \{ \text{sign}(e_i)K_{il}, \text{sign}(e_i)K_{ir} \} |e_i| + \max \{ \text{sign}(\dot{e}_j)K_{pl}, \text{sign}(\dot{e}_j)K_{pr} \} |\dot{e}_j| \right]
\end{align*}
\] (4.6b)

Figure 4.2 shows an example about how the bounds of the interval type-1 type-reduced set varies with the input signals. Since the bounds are a set of linear planes,
it may be concluded that a PI controller with prescribed gains can be realized by
the proposed type-2 fuzzy PI controller. Assuming that the plant parameters vary
within known bounds, then $K_{pl}$, $K_{pr}$, $K_{il}$ and $K_{ir}$ can be selected by substituting the
parameter bounds into empirical PI tuning rules. After the range of proportional
and integral gains are obtained, the closed-loop system stability using conventional
PI controller with such gains can be verified for any possible uncertain system whose
parameters are within known bounds. Since the type-2 fuzzy controller is equivalent
to a set of PI controllers which have usually been guaranteed as stable closed-loop
systems beforehand, the type-2 fuzzy controller structure can thus minimize the
possibility of encountering stability problems.

4.2 Analysis of Type-2 Fuzzy PI Controller

The main difference between type-2 FLS and type-1 FLS is the inclusion of a type-
reducer to map the type-2 output sets produced by the inference engine into a
type-1 fuzzy set [57]. As shown in Section 2.4.5, an interval type-2 fuzzy set that is symmetrical about the point \( m \) has the following properties [66]:

- The centroid of the type-2 fuzzy set is symmetrical about \( y = m \).
- The mean of all elements that make up the centroid is \( m \).

If interval type-2 fuzzy numbers characterized by symmetrical FOU are used to perform operations (e.g. arithmetic, set-theoretic and nonlinear function on them), the results will also be interval type-2 fuzzy numbers with symmetrical FOU. Hence, the result of combined centroid plus defuzzification procedures could be obtained as well by treating the T2 fuzzy numbers as crisp numbers and performing crisp operations on them [66].

For the current type-2 FLS framework of type-2 fuzzy PI controller, the antecedents are type-1 sets and height defuzzification is performed after type-reduction. On the other hand, the only type-2 sets are consequent sets, if symmetric interval type-2 sets are used, the final crisp output or the control surface would be the same as the result if symmetric type-1 sets replace the symmetric type-2 sets. This indicates that there is a need to change the manner in which the crisp output is calculated. In order to obtain a variable input-output mapping that differs from its type-1 counterpart, the type-2 fuzzy PI controller needs a revised type reduction method algorithm that adjusts the type-reduced output set according to the performance of the type-2 fuzzy controller.

### 4.3 Theorems on Properties of Centroids

The centroid calculation is very important in the type-reduction method, so this section will show the derivation of some important properties of centroids of interval type-2 fuzzy sets. The theoretical basis of the proposed type-reduction method is based on Theorem 4.1 and 4.2 which can be proved using the Property 2.1, 2.2 and 2.4 of centroids in Section 2.4.5 [38]:

...
Theorem 4.1. Given an interval type-2 fuzzy set which has a symmetrical FOU about the primary variable $y$ at $y = m$, the centroids of embedded type-1 fuzzy sets $A_e(l), C'_l$, must be contained within $[c_l(\tilde{A}), m]$; and the centroids of embedded type-1 fuzzy sets $A_e(r), C'_r$, must be contained within $[m, c_r(\tilde{A})]$, that is:

\begin{align*}
    C'_l &\in [c_l(\tilde{A}), m], \quad l \in X \quad (4.7) \\
    C'_r &\in [m, c_r(\tilde{A})], \quad r \in X \quad (4.8)
\end{align*}

Proof: Based on the Representation Theorem, the concept and definition of centroids (Equation (2.44) and (2.47)) already indicate that for any value of $l$ ($l \in X$) other than $c_l(\tilde{A})$, the centroid of embedded type-1 set $A_e(l)$ is larger than $c_l(\tilde{A})$. For any value of $r$ ($r \in X$) other than $c_r(\tilde{A})$, the centroid of embedded type-1 set $A_e(r)$ is no smaller than $c_r(\tilde{A})$. That is:

\begin{align*}
    c_l(\tilde{A}) &\leq C'_l \\
    C'_r &\leq c_r(\tilde{A})
\end{align*}

Hence, it is now only necessary to prove $C'_l \leq m$ and $m \leq C'_r$. Since the centroid is shift-invariant (see the Property 2.2 of centroids in Section 2.4.5), it is sufficient to analyze the set $\tilde{A}'$ that is obtained by shifting $\tilde{A}$ to the origin, so that $\Delta m = -m$. The theorem can be then established by proving that $\text{centroid}(A'_e(l)) \leq 0$ and $0 \leq \text{centroid}(A'_e(r))$, where $\tilde{A}'$ is symmetrical about the origin.

Consider the special embedded type-1 fuzzy set that is the upper membership function, $A'_{e,\text{UMF}}$. Since it is assumed that the FOU is symmetric, the centroid of $A'_{e,\text{UMF}}$ will be zero, which means:

\[ \int_{-\infty}^{\infty} x\bar{\mu}(x)dx = 0 \quad (4.9) \]
Hence, using Equation (2.45), the centroid of any embedded set $A'_e(l)$ is:

\[
\text{centroid}(A'_e(l)) = \frac{\int_{-\infty}^{l} x\bar{\mu}(x)dx + \int_{l}^{\infty} x\mu(x)dx}{\int_{-\infty}^{l} \bar{\mu}(x)dx + \int_{l}^{\infty} \mu(x)dx} = \frac{\int_{-\infty}^{l} x\bar{\mu}(x)dx - \int_{l}^{\infty} x(\bar{\mu}(x) - \mu(x))dx}{\int_{-\infty}^{l} \bar{\mu}(x)dx + \int_{l}^{\infty} \mu(x)dx}
\]

\[
= -\frac{\int_{l}^{\infty} x(\bar{\mu}(x) - \mu(x))dx}{\int_{-\infty}^{l} \bar{\mu}(x)dx + \int_{l}^{\infty} \mu(x)dx}
\]

(4.10)

It is always true that $(\bar{\mu}(x) - \mu(x)) \geq 0$ and $\int_{-\infty}^{l} \bar{\mu}(x)dx + \int_{l}^{\infty} \mu(x)dx > 0$. It follows that:

Case I: If $l \geq 0$, \(-\int_{l}^{\infty} x(\bar{\mu}(x) - \mu(x))dx \leq 0\)

\[
\Rightarrow \text{centroid}(A'_e(l)) = \frac{-\int_{l}^{\infty} x(\bar{\mu}(x) - \mu(x))dx}{\int_{-\infty}^{l} \bar{\mu}(x)dx + \int_{l}^{\infty} \mu(x)dx} \leq 0
\]

Case II: If $l < 0$, then

\[
\text{centroid}(A'_e(l)) = -\left[\frac{\int_{l}^{0} x(\bar{\mu}(x) - \mu(x))dx}{\int_{-\infty}^{l} \bar{\mu}(x)dx + \int_{l}^{\infty} \mu(x)dx} + \int_{-\infty}^{l} x(\bar{\mu}(x) - \mu(x))dx \right]
\]

\[
= -\frac{\int_{l}^{\infty} x(\bar{\mu}(x) - \mu(x))dx}{\int_{-\infty}^{l} \bar{\mu}(x)dx + \int_{l}^{\infty} \mu(x)dx} \leq 0
\]

In conclusion, it is always true $\text{centroid}(A'_e(l)) \leq 0$.

Similarly, using Equation (2.48), the centroid of any embedded set $A'_e(r)$ is:

\[
\text{centroid}(A'_e(r)) = \frac{\int_{-\infty}^{r} x\mu(x)dx + \int_{r}^{\infty} x\bar{\mu}(x)dx}{\int_{-\infty}^{r} \mu(x)dx + \int_{r}^{\infty} \bar{\mu}(x)dx} = \frac{\int_{-\infty}^{r} x\mu(x)dx - \int_{r}^{\infty} x(\mu(x) - \bar{\mu}(x))dx}{\int_{-\infty}^{r} \mu(x)dx + \int_{r}^{\infty} \bar{\mu}(x)dx}
\]

\[
= \frac{-\int_{r}^{\infty} x(\mu(x) - \bar{\mu}(x))dx}{\int_{-\infty}^{r} \mu(x)dx + \int_{r}^{\infty} \bar{\mu}(x)dx}
\]

(4.11)
It is always true that \((\mu(x) - \bar{\mu}(x)) \leq 0\) and \(\int_{-\infty}^{0} \mu(x) dx + \int_{0}^{\infty} \bar{\mu}(x) dx > 0\). It follows that:

Case I: If \(r \geq 0\), \(-\int_{r}^{\infty} x(\mu(x) - \bar{\mu}(x)) dx \geq 0\)

\[
\Rightarrow \text{centroid}(A'_e(r)) = \frac{-\int_{r}^{\infty} x(\mu(x) - \bar{\mu}(x)) dx}{\int_{-\infty}^{\infty} \mu(x) dx + \int_{r}^{\infty} \bar{\mu}(x) dx} \leq 0
\]

Case II: If \(r < 0\), then

\[
\text{centroid}(A'_e(r)) = -\left[\frac{\int_{0}^{r} x(\mu(x) - \bar{\mu}(x)) dx}{\int_{-\infty}^{\infty} \mu(x) dx + \int_{r}^{\infty} \bar{\mu}(x) dx} + \frac{\int_{0}^{r} x(\mu(x) - \bar{\mu}(x)) dx + \int_{-\infty}^{r} x(\mu(x) - \bar{\mu}(x)) dx}{\int_{-\infty}^{\infty} \mu(x) dx + \int_{r}^{\infty} \bar{\mu}(x) dx}\right]
\]

\[
= -\frac{\int_{-\infty}^{r} x(\mu(x) - \bar{\mu}(x)) dx}{\int_{-\infty}^{\infty} \mu(x) dx + \int_{r}^{\infty} \bar{\mu}(x) dx} \geq 0
\]

In conclusion, it is always true \(\text{centroid}(A'_e(r)) \geq 0\).

From the shift – invariant property of centroids, it will be true that:

\[
C'_l \in [c_l(\tilde{A}) , m] , \quad l \in X
\]

\[
C'_r \in [m , c_r(\tilde{A})] , \quad r \in X
\]

This is a new theorem that can be regarded as generalization of Theorem 7 in [66]. It demonstrates that for a symmetrical FOU, the centroid of special embedded set, \(A_e(l)\) or \(A_e(r)\) cannot cross over to other side of the symmetry point of the FOU.

**Theorem 4.2.** Let \([x_1, x_N]\) be the primary domain of an interval type-2 fuzzy set \(\tilde{A}\). Hence, the centroids of embedded type-1 fuzzy sets \(A_e(l)\), \(C'_l\), will decrease monotonically when \(l \in [x_1 , c_l(\tilde{A})]\) and increase monotonically when \(l \in [c_l(\tilde{A}) , x_N]\). Similarly, the centroids of embedded type-1 fuzzy sets \(A_e(r)\), \(C'_r\), will be increase monotonically when \(r \in [x_1 , c_r(\tilde{A})]\) and decrease monotonically when \(r \in [c_r(\tilde{A}) , x_N]\).
Hence, the sign of the derivative of $C$ of Chapter 4. Type-2 Fuzzy PI Controller with Adjustable Type-reduced Output 87

Proof: Consider an arbitrary embedded type-1 fuzzy set, $A_c(l)$. The derivative of $C'_i$ with respect to $l$ is, $\frac{d(C'_i)}{dl}$:

$$\frac{d}{dl} \left[ \int_{x_1}^{l} x\overline{\mu}(x)dx + \int_{l}^{x_N} x\mu(x)dx \right] = \frac{[\overline{\mu}(l) - \mu(l)]\left[\int_{x_1}^{l} x\overline{\mu}(x)dx + \int_{l}^{x_N} \mu(x)dx\right]}{\left[\int_{x_1}^{l} \overline{\mu}(x)dx + \int_{l}^{x_N} \mu(x)dx\right]^2}$$

$$- \frac{[\overline{\mu}(l) - \mu(l)]\left[\int_{x_1}^{l} x\overline{\mu}(x)dx + \int_{l}^{x_N} \mu(x)dx\right]}{\left[\int_{x_1}^{l} \overline{\mu}(x)dx + \int_{l}^{x_N} \mu(x)dx\right]^2}$$

(4.12)

It is always true that $(\overline{\mu}(l) - \mu(l)) \geq 0$ and $\int_{x_1}^{l} \overline{\mu}(x)dx + \int_{l}^{x_N} \mu(x)dx > 0$. Hence, the sign of the derivative of $C'_i$ thus only depends on the sign of the term, $l \left[ \int_{x_1}^{l} \overline{\mu}(x)dx + \int_{l}^{x_N} \mu(x)dx \right] - \left[ \int_{x_1}^{l} x\overline{\mu}(x)dx + \int_{l}^{x_N} x\mu(x)dx \right]$. There are two cases:

Case I: If $l < c_l(\tilde{A})$, then let $l = c_l - d$ ($d > 0$),

$$\Rightarrow l \left[ \int_{x_1}^{l} \overline{\mu}(x)dx + \int_{l}^{x_N} \mu(x)dx \right] - \left[ \int_{x_1}^{l} x\overline{\mu}(x)dx + \int_{l}^{x_N} x\mu(x)dx \right]$$

$$= (c_l - d) \left[ \int_{x_1}^{c_l} \overline{\mu}(x)dx + \int_{c_l}^{x_N} \mu(x)dx - \int_{c_l-d}^{c_l} (\overline{\mu}(x) - \mu(x))dx \right]$$

$$- \left[ \int_{x_1}^{c_l} x\overline{\mu}(x)dx + \int_{c_l}^{x_N} x\mu(x)dx \right] + \left[ \int_{c_l-d}^{c_l} x(\overline{\mu}(x) - \mu(x))dx \right]$$

$$= \int_{c_l-d}^{c_l} (x - c_l)(\overline{\mu}(x) - \mu(x))dx - d \left[ \int_{x_1}^{l} \overline{\mu}(x)dx + \int_{l}^{x_N} \mu(x)dx \right] \leq 0$$

Hence, it is easy to conclude that $\frac{d(C'_i)}{dl} \leq 0$. The centroids of embedded type-1 fuzzy sets $A_c(l)$, $C'_i$, will monotonically decrease when $l \in [x_1, c_l(\tilde{A})]$.

Similarly,

Case II: If $l > c_l(\tilde{A})$, then let $l = c_l + d$ ($d > 0$),

$$\Rightarrow l \left[ \int_{x_1}^{l} \overline{\mu}(x)dx + \int_{l}^{x_N} \mu(x)dx \right] - \left[ \int_{x_1}^{l} x\overline{\mu}(x)dx + \int_{l}^{x_N} x\mu(x)dx \right]$$

$$= (c_l + d) \left[ \int_{x_1}^{c_l} \overline{\mu}(x)dx + \int_{c_l}^{x_N} \mu(x)dx + \int_{c_l}^{c_l+d} (\overline{\mu}(x) - \mu(x))dx \right]$$

$$- \left[ \int_{x_1}^{c_l} x\overline{\mu}(x)dx + \int_{c_l}^{x_N} x\mu(x)dx \right] - \left[ \int_{c_l}^{c_l+d} x(\overline{\mu}(x) - \mu(x))dx \right]$$

$$= \int_{c_l}^{c_l+d} (c_l - x)(\overline{\mu}(x) - \mu(x))dx + d \left[ \int_{x_1}^{l} \overline{\mu}(x)dx + \int_{l}^{x_N} \mu(x)dx \right] \geq 0$$
Likewise, \( \frac{d(C'_l)}{dl} \geq 0 \). The centroids of embedded type-1 fuzzy sets \( A_e(l), C'_1 \), will monotonically increase when \( l \in [c_l(\tilde{A}), x_N] \).

The proof for embedded type-1 fuzzy sets \( \tilde{A}_e(r) \) is similar so it is omitted.

Theorem 4.1 shows that the centroids of the embedded type-1 fuzzy sets are bounded regardless of the positions of the switch points, once the shape of the type-2 fuzzy set is fixed. This property is useful because the equivalent proportional and integral gain that results when the centroids of the consequent sets are varied would still be bounded. Hence, the system structure serves as a constraint that minimizes the possibility of encountering stability problems.

Theorem 4.2 shows that when the left and right switch points are shifted in either directions away from \( c_l(\tilde{A}) \) and \( c_r(\tilde{A}) \), the left and right end points of the centroid will both move towards \( m \), the center of the set. Hence, it would be sufficient to search the range \([x_1, c_l(\tilde{A})]\), instead of the entire primary domain \([x_1, x_N]\), for an alternative left switch point \( l \) in order to alter the output of the type-2 controller. Similarly, the range of possible right switch point \( r \) lies in \([c_r(\tilde{A}), x_N]\). The search for \( l \) and \( r \) can be directed by the derivatives of the centroids with respect to the switch points. Example in Figure 4.4 shows the values of centroids of both the embedded sets along the primary domain for the type-2 fuzzy set in Figure 4.3.

### 4.4 Adaptive Algorithm for Type-reduction

#### 4.4.1 Switch point adjustment algorithm

Equation (4.2) shows that centroids of consequent sets are bounded and the two end points of each centroid are actually related to the pre-determined proportional gains or integral gains. This framework will result in a type-reduced set whose bounds are equivalent to a PI controller with gains within the specified range. Once the system parameters deviate from the nominal values, the type-2 PI controller
should still be able to provide a suitable output within the control surface range to maintain the performance in spite of the uncertainties. This requires the equivalent gains of type-2 PI controller to vary based on the uncertainty and at the same time stay within the pre-determined range to provide a suitable output surface. As the defuzzification method is to choose the mid-point of \( y_l \) and \( y_r \) in Equation (4.6), one way to provide a variable control surface is to derive an algorithm to vary \( y_l \) or \( y_r \) based on the uncertainty information. To vary the value of \( y_l \) or \( y_r \), the end points of centroids used in Equation (4.6) need to change. Hence, the main idea of the algorithm for type-reduction is to update the switch points of embedded sets and use their centroids to calculate Equation (4.6). By varying the switch points according to the uncertainties, the centroids and the value of \( y_l \) or \( y_r \) will change to generate a variable control surface.

Suppose the current values of \( C_{1_{\text{old}}}^{u_k} \) and \( C_{r_{\text{old}}}^{u_k} \) (denoted as \( C_{1_{\text{old}}}^{\text{old}} \) and \( C_{r_{\text{old}}}^{\text{old}} \)) should change to the new values \( C_{1_{\text{new}}}^{u_k} \) and \( C_{r_{\text{new}}}^{u_k} \) such that:

\[
\Delta C_{1_{\text{old}}}^{u_k} = C_{1_{\text{new}}}^{u_k} - C_{1_{\text{old}}}^{u_k} = L\varepsilon C_{1_{\text{old}}}^{u_k}, \quad \Delta C_{r_{\text{old}}}^{u_k} = C_{r_{\text{new}}}^{u_k} - C_{r_{\text{old}}}^{u_k} = L\varepsilon C_{r_{\text{old}}}^{u_k}
\]

(4.13)
Figure 4.4. An example of two Theorems for the particular type-2 fuzzy set

\[ \varepsilon = Y_{ref} - Y_{actual} \] is the error between the uncertain system response and the nominal system response. The learning rate \( L \) determines how fast the system adjusts its control surface and it is a parameter to balance between performance and robustness of the type-2 PI controller. It is defined as:

\[
L = \frac{(1 + \alpha |\varepsilon|) L_{ss}}{T} \quad (4.14)
\]

where \( L_{ss} \) is a base learning rate. \( T \) is the time relative to the time when there is a change in the setpoint and \( \alpha \) is a weighting factor. The learning rate \( L \) is designed such that more attention is paid to the initial period following a setpoint change and less to the steady state where the error between the actual and reference responses may be very small.

Theorem 4.2 shows that the derivatives of centroids with respect to switch points are monotonic on either side of fuzzy set (it is assumed that the domain of left switch point, \( l \), is the left side support of fuzzy set while the domain of \( r \) is the right side).
This will guarantee a unique value for adjustment of switch point to obtain the desired new centroid. Hence, the algorithm for adjusting the switch points ($l$ and $r$) can be obtained as follows:

\[ \delta_{lk} = \left. \frac{\Delta C_{uk}^{lk}}{dC_{uk}^{lk}} \right|_{l=l_{old}} + \Delta \delta_{lk} = \frac{L\varepsilon C_{l_{old}}^{uk}}{dC_{uk}^{lk}} + \beta L\varepsilon C_{l_{old}}^{ui} \]  

\[ \delta_{rk} = \left. \frac{\Delta C_{uk}^{rk}}{dC_{uk}^{rk}} \right|_{l=l_{old}} + \Delta \delta_{rk} = \frac{L\varepsilon C_{r_{old}}^{uk}}{dC_{uk}^{rk}} + \beta L\varepsilon C_{r_{old}}^{ui} \]  

$C_{l_{old}}^{uk}$ and $C_{r_{old}}^{uk}$ are end points of centroid in $k$th consequent at previous adaptation iteration. The derivatives of centroids with respect to switch points, $\frac{dC_{uk}^{lk}}{dl}$ and $\frac{dC_{uk}^{rk}}{dr}$, are needed to estimate the adjustment in switch points to achieve desired adjustment in centroids. The first term in the right hand side of Equation (4.15) or (4.16) has the effect of varying the equivalent gains while roughly maintaining a fixed relationship between the proportional and integral gains. Hence, the second term, $\Delta \delta_{lk} = \beta L\varepsilon C_{l_{old}}^{ui}$ or $\Delta \delta_{rk} = \beta L\varepsilon C_{r_{old}}^{ui}$, is used to give the centroids extra freedom to vary the gain relationship and thus a more flexible control surface may be generated. $\beta$ is a scaling factor for the learning rate of the gain-proportion adjustment. $C_{l_{old}}^{ui}$ is the left end point of the centroid of $i$th consequent in previous adaptation iteration where $e_{k2} = 0$ as in Equation (4.2); $C_{r_{old}}^{ui}$ is the right end point of old centroid of $i$th consequent in previous adaptation iteration where $e_{k2} = 0$. Since $C_{l_{old}}^{ui}$ and $C_{r_{old}}^{ui}$ is associated to the proportional gain $K_{pl}$ or $K_{pr}$, the second term has the effect to fine tune the equivalent proportional gain $K_{pl}$ or $K_{pr}$ related to all the centroids of consequents and thus provide a flexible way to vary the proportion between the equivalent proportion gains and integral gains. The second term may hopefully provide extra freedom in varying the control surface and thus improve the performance and accelerate the tuning process.
4.4.2 Derivatives of centroid with respect to switch points

The adjustment of switch points in the adaptive algorithm needs the derivatives of centroids of embedded sets with respect to the switch points. In order to provide the formulae to calculate the derivatives for use in type-reduction algorithm, the derivatives of $C'_l$ and $C'_r$ of the type-2 fuzzy set used in the study are derived in this subsection. Figure 4.5 shows the symmetric interval type-2 fuzzy set used in the derivations, it has uncertain mean and most parts of lower MF and upper MF are parallel. Thus, the primary domain or support of the fuzzy type-2 set may be describe using a scaling factor $K_v$.

![Figure 4.5. The standard fuzzy set used in this chapter](image)

The derivative of $C'_l$ with respect to $l$ is:

$$
\frac{dC'_l}{dl} = \frac{d}{dl} \left[ \frac{\int_{-\infty}^{l} x\bar{\mu}(x)dx + \int_{l}^{\infty} x\mu(x)dx}{\int_{-\infty}^{l} \bar{\mu}(x)dx + \int_{l}^{\infty} \mu(x)dx} \right]
$$

$$
= \frac{[\bar{\mu}(l) - \mu(l)]}{[\int_{-\infty}^{l} \bar{\mu}(x)dx + \int_{l}^{\infty} \mu(x)dx]} - \frac{[\bar{\mu}(l) - \mu(l)][\int_{-\infty}^{l} x\bar{\mu}(x)dx + \int_{l}^{\infty} x\mu(x)dx]}{[\int_{-\infty}^{l} \bar{\mu}(x)dx + \int_{l}^{\infty} \mu(x)dx]^2}
$$

$$
= \frac{[\bar{\mu}(l) - \mu(l)][l - C'_l]}{[\int_{-\infty}^{l} \bar{\mu}(x)dx + \int_{l}^{\infty} \mu(x)dx]}
$$

(4.17)

Consider two cases:

**Case I:** $m - (K_v - 2)a < l < m - a$. 
According to the upper membership function:

\[
\mu(l) = \frac{l - m + K_v \times a}{(K_v - 1)a}, \quad (4.18)
\]

\[
\int_{-\infty}^{l} \mu(x)dx = \frac{1}{2} \frac{(l - m + K_v \times a)^2}{(K_v - 1)a} \quad (4.19)
\]

Similarly, according to the lower membership function,

\[
\mu(l) = \frac{l - m + (K_v - 2)a}{(K_v - 1)a} \quad (4.20)
\]

\[
\int_{l}^{\infty} \mu(x)dx = \frac{(K_v - 2)^2 a}{K_v - 1} - \frac{1}{2} \frac{(l - m + (K_v - 2)a)^2}{(K_v - 1)a} \quad (4.21)
\]

Substituting into Equation (4.17)

\[
\frac{dC'_l}{dl} = \frac{2(l - C'_l)}{K_v^2 a + 2(l - m) - 2(K_v - 1)a} \quad (4.22)
\]

**Case II: \( l \leq m - (K_v - 2)a \)**

In this case, according to the upper membership function,

\[
\mu(l) = \frac{l - m + K_v \times a}{(K_v - 1)a} \quad (4.23)
\]

\[
\int_{-\infty}^{l} \mu(x)dx = \frac{1}{2} \frac{(l - m + K_v \times a)^2}{(K_v - 1)a} \quad (4.24)
\]

According to the lower membership function,

\[
\mu(l) = 0 \quad (4.25)
\]

\[
\int_{l}^{\infty} \mu(x)dx = \frac{(K_v - 2)^2 a}{K_v - 1} \quad (4.26)
\]

Finally,

\[
\frac{dC'_l}{dl} = \frac{2(l - m + K_v \times a)(l - C'_l)}{(l - m + K_v \times a)^2 + 2(K_v - 2)^2 a^2} \quad (4.27)
\]
The derivative of the centroid $\text{centroid}(A_r(r))$, denoted by $C_r'$, can be derived using a similar procedure.

\[
\frac{dC_r'}{dr} = \frac{d}{dr} \left[ \int_{-\infty}^{r} x\mu(x)dx + \int_{r}^{\infty} x\overline{\mu}(x)dx \right] = \frac{\mu(r) - \overline{\mu}(r)}{\int_{-\infty}^{r} \mu(x)dx + \int_{r}^{\infty} \overline{\mu}(x)dx} \left[ \int_{-\infty}^{r} x\mu(x)dx + \int_{r}^{\infty} x\overline{\mu}(x)dx \right]
\]

\[
= \frac{\mu(r) - \overline{\mu}(r)}{\int_{-\infty}^{r} \mu(x)dx + \int_{r}^{\infty} \overline{\mu}(x)dx} \left[ \mu(r) - \overline{\mu}(r) \right] \left[ \int_{-\infty}^{r} \mu(x)dx + \int_{r}^{\infty} \overline{\mu}(x)dx \right]^2
\]

Derivation process is similar and results are summarized here,

**Case I:** $m + (Kv - 2)a > r > m + a$

\[
\frac{dC_r'}{dr} = \frac{2(C_r' - r)}{Kv^2a + 2(m - r) - 2(Kv - 1)a}
\]

**Case II:** $r \geq m + (Kv - 2)a$

\[
\frac{dC_r'}{dr} = \frac{2(m + K v \times a - r)(C_r' - r)}{(m + K v \times a - r)^2 + 2(Kv - 2)^2a^2}
\]

### 4.4.3 Algorithm initialization

The manner in which the algorithm for adjusting the switch points is initialized is described in this subsection. The switch points, $l$ and $r$, are set to the mid-point of the search domain: $[x_1, c_l(\tilde{A})]$ and $[c_r(\tilde{A}), x_N]$. As the initial left and right switch points are symmetrical about the center of the respective type-2 sets, the output of the type-2 PI controller is equal to that of the fuzzy PI controller when type-2 consequents is replaced by type-1 sets. Hence, the algorithm may be regarded as starting from a baseline type-1 PI controller and gradually searches for a more suitable control action.

Based on Equation (4.2), the mid point of the symmetric consequent type-2 set,
Chapter 4. Type-2 Fuzzy PI Controller with Adjustable Type-reduced Output 95

$m^k$ will always be:

$$m^k = \frac{K_{pl} + K_{pr}}{2} \dot{e}_{k2} + \frac{K_{il} + K_{ir}}{2} e_{k1} = K_p^m \dot{e}_{k2} + K_i^m e_{k1} \quad (4.31)$$

where $K_p^m = \frac{K_{pl} + K_{pr}}{2}$, $K_i^m = \frac{K_{il} + K_{ir}}{2}$.

After height defuzzification, the type-2 PI controller that is initialized as described has an equivalent proportional gain $K_p^m$ and an integral gain $K_i^m$. The control surface with equivalent gains $K_p^m$ and $K_i^m$ is shown as the planar surface in Figure 4.6. Theorem 4.1 shows that the range for updated endpoints of centroid are always bounded and divided by the middle point. For a symmetric type-2 set, the possible range for updating the left switch point is from the left end point of support to the middle point of the support; while the right switch point is from the middle point to the right end. If the switch point is updated to the middle point, its centroid of embedded set will relate to gains $K_p^m$ and $K_i^m$. If the switch point is updated to end point of support, its centroid of embedded set will relate to gains $K_{pl}$, $K_{pr}$, $K_{il}$ and $K_{ir}$. Hence, the control surface obtained after type-reduction with adaptive algorithm must be between the upper surface and the planar surface or between the planar surface and the lower surface shown in Figure 4.6. To illustrate this result, the possible range of equivalent gain of the type-2 fuzzy PI controller during the update process is derived. Suppose after adjustment of the switch points, the
updated centroid becomes \( \hat{C}_{uk} = [\hat{C}_{uk}^l, \hat{C}_{uk}^r] \), where:

\[
\hat{C}_{uk}^l = \min\{\text{sign}(e_k)\hat{K}_d, \text{sign}(e_k)\hat{K}_ir\} \times |e_k| \\
+ \min\{\text{sign}(\dot{e}_k)\hat{K}_{pl}, \text{sign}(\dot{e}_k)\hat{K}_{pr}\} \times |\dot{e}_k|
\]

(4.32a)

\[
\hat{C}_{uk}^r = \max\{\text{sign}(e_k)\hat{K}_d, \text{sign}(e_k)\hat{K}_ir\} \times |e_k| \\
+ \max\{\text{sign}(\dot{e}_k)\hat{K}_{pl}, \text{sign}(\dot{e}_k)\hat{K}_{pr}\} \times |\dot{e}_k|
\]

(4.32b)

where \( \hat{K}_{pl}, \hat{K}_{pr}, \hat{K}_d \) and \( \hat{K}_d \) may be regarded as gains related to centroids. These updated parameters are bounded as:

\[
K_{pl} \leq \hat{K}_{pl} \leq K^m_{pl}, \quad K^m_{pl} \leq \hat{K}_{pr} \leq K_{pr}
\]

(4.33)

\[
K_{il} \leq \hat{K}_{il} \leq K^m_{il}, \quad K^m_{il} \leq \hat{K}_{ir} \leq K_{ir}
\]

(4.34)

Figure 4.6. Illustration of equivalent gains for type-2 PI using the algorithm

Since height defuzzification transforms the type-reduced set into a crisp value
equal to the mean of the upper and lower bounds of the interval set, the type-2 PI controller cannot provide an output surface with equivalent gains $K_{pl}$, $K_{pr}$, $K_{il}$ and $K_{ir}$ even though its type-reduced set may have equivalent gains related to such values. The range of equivalent proportional $[K_{p,min}, K_{p,max}]$ and integral gains $[K_{i,min}, K_{i,max}]$ of the type-2 PI controller can be derived using the updated centroids in Equation (4.32),

$$K_{p,min} = \min \left\{ \frac{\hat{K}_{pl} + \hat{K}_{pr}}{2} \right\} = \frac{K_{pl} + K_{pr}}{2} = \frac{3K_{pl} + K_{pr}}{4},$$  

(4.35)

$$K_{p,max} = \max \left\{ \frac{\hat{K}_{pl} + \hat{K}_{pr}}{2} \right\} = \frac{K_{p}^m + K_{pr}}{2} = \frac{K_{pl} + 3K_{pr}}{4}.$$  

(4.36)

$$K_{i,min} = \min \left\{ \frac{\bar{K}_{il} + \bar{K}_{ir}}{2} \right\} = \frac{K_{il} + K_{ir}}{2} = \frac{3K_{il} + K_{ir}}{4},$$  

(4.37)

$$K_{i,max} = \max \left\{ \frac{\bar{K}_{il} + \bar{K}_{ir}}{2} \right\} = \frac{K_{i}^m + K_{ir}}{2} = \frac{K_{il} + 3K_{ir}}{4}.$$  

(4.38)

Since $K_{p,min}$ and $K_{p,max}$ are the range of actual output equivalent proportional gain of type-2 PI controller, they should also be the smallest and largest proportional gains needed to control the set of uncertain plants that the controller may encounter. Likewise, $K_{i,min}$ and $K_{i,max}$ are determined based on the pre-known knowledge of uncertain systems.

Solving Equations (4.35) to (4.38) to calculate $[K_{pl}, K_{pr}]$ and $[K_{il}, K_{ir}]$, the parameters which that initialize the centroids of the proposed type-2 PI controller (Equation (4.2)) can be obtained as:

$$K_{pl} = \frac{3K_{p,min} - K_{p,max}}{2}$$  

(4.39)

$$K_{pr} = \frac{3K_{p,max} - K_{p,min}}{2}$$  

(4.40)

$$K_{il} = \frac{3K_{i,min} - K_{i,max}}{2}$$  

(4.41)

$$K_{ir} = \frac{3K_{i,max} - K_{i,min}}{2}.$$  

(4.42)
4.5 Simulation Results

This section investigates the performance of the type-2 fuzzy PI controller with variable centroids by using it to control an uncertain first order plus dead time (FOPDT) system:

\[ G(s) = \frac{Ke^{-Ts}}{(\tau s + 1)} \]  

(4.43)

It is assumed that the nominal model is

\[ \frac{e^{-s}}{(5s + 1)} \]  

(4.44)

and the static gain and time constant may vary from the nominal values by 10% i.e. \( K \in [0.9, 1.1] \) and \( \tau \in [4.5, 5.5] \).

The ITAE setpoint tuning method [79] as defined below is used to set up the type-2 fuzzy PI controller:

\[
K_p = \frac{0.586}{K} \left( \frac{T_d}{\tau} \right)^{-0.916}
\]  

(4.45)

\[
K_i = K_p \times \left[ 1.03 - 0.165 \left( \frac{T_d}{\tau} \right) \right]
\]  

(4.46)

By substituting \( K \in [0.9, 1.1] \) and \( \tau \in [4.5, 5.5] \) into the ITAE setpoint tuning method, the range of proportional and integral gains for providing a performance that is close to the desired performance is calculated as \( [K_{p,\text{min}}, K_{p,\text{max}}] = [2.11, 3.10] \) and \( [K_{i,\text{min}}, K_{i,\text{max}}] = [0.46, 0.57] \). Using Equation (4.39) to (4.42), the parameters for initializing the centroids of the proposed type-2 PI controller are:

\[
K_{pl} = \frac{3K_{p,\text{min}} - K_{p,\text{max}}}{2} = 1.6175, \quad K_{pr} = \frac{3K_{p,\text{max}} - K_{p,\text{min}}}{2} = 3.5986
\]

\[
K_{il} = \frac{3K_{i,\text{min}} - K_{i,\text{max}}}{2} = 0.4075, \quad K_{ir} = \frac{3K_{i,\text{max}} - K_{i,\text{min}}}{2} = 0.6242
\]

The other parameters of adaptation algorithm and parameters of the type-2 fuzzy
PI controller are chosen as:

\[ L_{ss} = 0.06 \, , \, \alpha = 10 \, , \, \beta = 0.4 \, , \, K_v = 11 \, , \, e_1 = -5 \, , \]
\[ e_2 = 0 \, , \, e_3 = 5 \, , \, \dot{e}_1 = -10 \, , \, \dot{e}_2 = 0 \, , \, \dot{e}_3 = 10 \]

The test signal is a pulse signal with unit amplitude, a period of 40 seconds and 50% duty cycle. Sampling time controller is set as 0.1 second. The reference signal is the response of nominal plant under the control of a PI controller tuned by the ITAE setpoint tuning method. This study examines whether the proposed controller is able to maintain consistent control performance regardless of the actual process parameters.

![Figure 4.7. ITAEs of type-2 fuzzy PI controller in Monte Carlo uncertainty analysis](image)

500 Monte Carlo simulations are carried out for a robust uncertainty and performance analysis for the uncertain FOPDT plants. 500 uncertain FOPDT systems for the analysis are randomized to perform the Monte Carlo analysis and each simulation time is set as 300 seconds. The ITAEs of type-2 fuzzy PI controller in the
Chapter 4. Type-2 Fuzzy PI Controller with Adjustable Type-reduced Output

Figure 4.8. Histogram of ITAEs of type-2 fuzzy PI controller

Figure 4.9. ITAEs of type-1 fuzzy PI controller in Monte Carlo uncertainty analysis
Chapter 4. Type-2 Fuzzy PI Controller with Adjustable Type-reduced Output

500 Monte Carlo simulations congregate and are always close to the nominal ITAE value, which is shown in Figure 4.7. The histogram of ITAEs of type-2 fuzzy PI controller in Figure 4.8 shows clearly that nearly 80% of the ITAEs are range between 3.12 and 3.16. That means the proposed type-2 fuzzy PI controller handle the uncertainty very well and can always maintain the desired performance regardless of the uncertainty whose range is pre-known inside the FOPDT systems.

Figure 4.9 shows the ITAEs of type-1 fuzzy PI controller in the 500 Monte Carlo simulations. It shows that except for the cases that the uncertain parameters are near to the nominal values, the ITAEs may vary from the nominal ITAE value of 3.1406 to more than 4. Not surprisingly, the histogram of the ITAEs in Figure 4.10 shows that the type-1 fuzzy PI controller cannot handle the uncertainty of the system well. The ITAEs mainly uniformly range from 3.05 to 3.8 which is far away from the nominal ITAE value.

The most challenging task that will be faced by the type-2 fuzzy controller is when the deviation between the actual and nominal process parameters is largest.
Simulation is performed to investigate the step responses obtained under such uncertainties. The resulting step responses are shown in Figure 4.11 when $K = 0.9$, $\tau = 4.5$ and in Figure 4.12 when $K = 1.1$, $\tau = 5.5$. The ITAE values for this two cases is also tabulated in Table 4.1. Both the step responses and the ITAE value demonstrate that the type-2 fuzzy PI controller performs well.

<table>
<thead>
<tr>
<th>Controller</th>
<th>$K = 0.9$</th>
<th>$K = 1.1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau = 4.5$</td>
<td>3.8763</td>
<td>3.7261</td>
</tr>
<tr>
<td>$\tau = 5.5$</td>
<td>3.1783</td>
<td>3.1366</td>
</tr>
<tr>
<td>Reference</td>
<td>3.1406</td>
<td>3.1406</td>
</tr>
</tbody>
</table>

### 4.6 Comparison with Fuzzy PI Gain-scheduling Control

The proposed adaptive algorithm for type-reduction changes the switch points and thus varies the centroids of consequents. This may vary the relevant proportional and integral gains of the type-2 PI controller. After performing the type-reduction and defuzzification, the relevant gains of the type-2 fuzzy PI controller may change to handle the uncertainty and maintain the desired performance. The control scheme of the proposed type-2 fuzzy PI controller is similar to a fuzzy PI gain-scheduling controller. This section compares the performances between the proposed type-2 fuzzy PI controller and traditional type-1 fuzzy gain-scheduling controller on controlling the nonlinear pH neutralization process in Chapter 3.

The well-known fuzzy gain scheduling PID control scheme proposed in [101] was used in the comparative study. In order to create a same basis for comparison, the design scheme in [101] is used to design a type-1 fuzzy PI gain-scheduling controller based on the uncertain system parameters. The inputs of the type-1 fuzzy PI gain-scheduling controller are also $e$ and $\dot{e}$. The input is described using seven triangular
Figure 4.11. Responses of three control systems when $K = 0.9$, $\tau = 4.5$. (a) The first step response; (b) Step response after adaptation.
Figure 4.12. Responses of three control systems when $K = 1.1$, $\tau = 5.5$. (a) The first step response; (b) Step response after adaptation.
type-1 fuzzy sets as in Figure 4.13. The consequent sets for the normalized gains $K'_p$ and $K'_i$ are the two exponential functions shown in Figure 4.14. The actual PI gains may be obtained from $K'_p$ and $K'_i$ as:

$$K_p = (K_{p,max} - K_{p,min})K'_p + K_{p,min} \quad (4.47)$$

$$K_i = (K_{i,max} - K_{i,min})K'_i + K_{i,min} \quad (4.48)$$

Hence, the output of the fuzzy PI gain-scheduling controller is the derivative of the control action as well:

$$\dot{u} = K_p \dot{e} + K_i e, \quad (4.49)$$

The fuzzy rule-base is determined with heuristic based on the step response of the process, i.e. fast rise time and small overshoot. Thus, the sets of rule-base are shown in Table 4.2 for $K'_p$ and Table 4.3 for $K'_i$ (S stands for Small and B stands for Big).

### 4.6.1 Uncertain parameters for pH neutralization process

For the simulations, one of the most important things is to determine the range of gains, $[K_{p,min}, K_{p,max}]$ and $[K_{i,min}, K_{i,max}]$ based on the uncertain parameters of the
Figure 4.14. Consequents for $K'_p$ and $K'_i$ of fuzzy PI gain-scheduling controller.
pH neutralization process. For reference, the dynamics of the CSTR mixing process shown in Equation (3.15) and (3.16) is restated again:

\[
\begin{align*}
V \frac{dx_a}{dt} &= F_a C_a - (F_a + F_b) x_a \\
V \frac{dx_b}{dt} &= F_b C_b - (F_a + F_b) x_b
\end{align*}
\]

The nominal pH plant is same as the one in Chapter 3, where the nominal value of \(F_a\) is 0.8(liter/min) and \(F_b\) is 0.1279(liter/min), and the nonlinear mixing dynamic might be regarded as relatively linear as:

\[
V \frac{dx_b}{dt} \approx F_b C_b - F_a x_b \quad (4.50)
\]

After performing Laplace Transform, the transform function of the pH neutralization process might be linearized as:

\[
x_b(s) V s \approx F_b(s) C_b - F_a x_b(s) \quad \Rightarrow \quad G(s) = \frac{x_b(s)}{F_b(s)} = \frac{C_b}{V s + F_a} \quad (4.51)
\]

Usually the pH measurement at the effluent stream may cause some delay for the whole control system. For the simulation, it is assumed that the measurement delay is 0.5 second. The sampling time of the simulations is set as 0.1 second. Hence, the delay time(\(T_d\)) is five samples and the pH neutralization process is finally roughly modelled as:

\[
G(s) = \frac{K e^{-T_d s}}{\tau s + 1} = \frac{C_b}{F_a} e^{-T_d s} \quad (4.52)
\]

Sometimes, the flow rate of the process stream from the factory may vary due to the working condition. Hence, \(F_a\) is actually an uncertain parameter, which is now assumed to be within the range of \([0.7, 0.9]\)(liter/min). The concentration of the influent stream, \(C_b\) might also be an uncertain factor. The value of \(C_b\) is
assumed to be within \([0.09, 0.11] (gm - mol/liter)\) and its nominal value is still 0.1\((gm - mol/liter)\). Under these conditions, the uncertain range of \(K\) and \(\tau\) for the pH neutralization process will be:

\[
K = [0.1, 0.1571], \quad \tau = [2.2222, 2.8571]
\]

(4.53)

Using the ITAE tuning method in Equation (4.53), the possible range of desired proportional and integral gains will be:

\[
[K_{p,\text{min}}, K_{p,\text{max}}] = [1.7742, 3.5097], \quad [K_{i,\text{min}}, K_{i,\text{max}}] = [0.5259, 0.9106]
\]

(4.54)

### 4.6.2 Simulation results for pH neutralization process with uncertain parameters

The test signal for the pH neutralization process is a step signal with setpoints ranging from 5 to 10 and a cycle period of 400 seconds. The reference signal is the response of fuzzy gain-scheduling PI control on the nominal plant whose parameters are the same as in Chapter 3. For both of the gain-scheduling PI controller and the proposed type-2 PI controller, the domain of \(e\) and \(\dot{e}\) are set as \(e \in [-0.02, 0.02]\) and \(\dot{e} \in [-0.5, 0.5]\).

This study compares the performances between the proposed type-2 PI controller and the conventional gain-scheduling PI controller on controlling the pH neutralization process with uncertain parameters. Suppose that uncertainties cause the parameters of the pH neutralization process to change to \(F_a = 0.8593\) (liter/min) and \(C_b = 0.0932\) (gm - mol/liter). The dashed line in Figure 4.15 is the response of the proposed type-2 PI controller after six learning iterations for pH neutralization at different setpoints. The solid line in 4.15 is the reference signal which is the response of conventional gain-scheduling PI controller on controlling the nominal pH plant when \(F_a = 0.8\) (liter/min) and \(C_b = 0.1\) (gm - mol/liter). Although the pH neutralization process is severely nonlinear, the response of the proposed type-2
PI controller can gradually adjust the switch points of consequent and thus adjust the output according to the uncertainty to track the desired reference signal more closely. The ISE at the sixth iteration between the response of type-2 PI controller and the reference is 2.863.

![Performance comparison after six learning iterations](image)

Figure 4.15. Performances of proposed type-2 PI and conventional gain-scheduling controllers

The dotted line in Figure 4.15 shows the responses of the conventional gain-scheduling PI controller on the uncertain pH neutralization process. When the setpoint is around critical pH 8 to 8.5, the response obtained using gain-scheduling PI controller deviates from the reference. The ISE at the sixth iteration between the response of type-2 PI controller and the reference is 9.215 much bigger than the type-2 fuzzy PI controller. The result shows that the proposed type-2 PI controller is better than conventional gain-scheduling PI controller for uncertain nonlinear systems.
Chapter 4. Type-2 Fuzzy PI Controller with Adjustable Type-reduced Output

As a further analysis, 100 pH neutralization plant with uncertain $F_a$ and $C_a$ are randomly generated for testing the performance of proposed type-2 PI controller and gain-scheduling controller. The performance index is the ISE between the reference and the responses along the setpoint change after six learning iterations for each uncertain plant. Figure 4.16 shows the ISEs of responses of type-2 PI controller are mostly within 3. On the other hand, the ISEs of gain-scheduling controller in Figure 4.17 are within 9. This indicates that the performance of the type-2 PI controller is better than conventional gain-scheduling controller in [101] for pH neutralization.

![Histogram of ISE of type-2 PI for pH neutralization](image)

Figure 4.16. Histogram of the ISEs between the reference and responses of type-2 PI controller

4.7 Case Study: Thermal chamber

In this section, the thermal chamber shown in Figure 3.16 is again used as a test bed to investigate the performance of type-2 fuzzy PI controller with adjustable type-reduced output on handling system with uncertain parameters. There is a fan attached to one side of the thermal chamber. The speed of the fan is adjustable and
thus it is used to provide the thermal chamber with uncertain system parameters for the experiment. Suppose the fan can be adjusted from 20% of full speed to 90% of full speed, then the thermal chamber may be considered as an uncertain system and the first step is to identify such an uncertain system with its range of possible system parameters. Using step response system identification method and adjusting the fan speed at different levels within the range, the possible range of system parameters may be roughly identified. The transfer function of the thermal chamber can be described as a first-order system as:

\[ G(s) = \frac{K e^{-L}}{(\tau s + 1)} \]  

(4.55)

The nominal system is the same as Equation (3.17) in Chapter 3. When the fan speed is uncertain and adjustable within the pre-known range, the possible range of
static gain and time constant can be estimated as:

\[ K \in [22, 26] \text{ and } \tau \in [7.5, 7.8] \quad (4.56) \]

The reference tracking trajectory is the closed loop response of nominal system with a singleton controller as in Chapter 3. The ITAE setpoint tuning method [79] in Equation (4.45) and (4.46) is used to set up the type-2 fuzzy PI controller. By substituting \( K \in [22, 26] \) and \( \tau \in [7.5, 7.8] \) into the ITAE setpoint tuning method, the range of proportional and integral gains is calculated as \([K_{p,\min}, K_{p,\max}] = [0.43, 0.5267]\) and \([K_{i,\min}, K_{i,\max}] = [0.0585, 0.0693]\). Using Equation (4.39) to (4.42), the parameters for initializing the centroids of the proposed type-2 PI controller are:

\[
K_{pl} = \frac{3K_{p,\min} - K_{p,\max}}{2} = 0.3817, \quad K_{pr} = \frac{3K_{p,\max} - K_{p,\min}}{2} = 0.5751 \\
K_{il} = \frac{3K_{i,\min} - K_{i,\max}}{2} = 0.0531, \quad K_{ir} = \frac{3K_{i,\max} - K_{i,\min}}{2} = 0.0747
\]

The other parameters of adaptation algorithm and parameters of the type-2 fuzzy PI controller are chosen as:

\[ L_{ss} = 0.002, \quad \alpha = 10, \quad \beta = 0.4, \quad Kv = 11, \quad e_1 = -5, \]  
\[ e_2 = 0, \quad e_3 = 5, \quad \dot{e}_1 = -10, \quad \dot{e}_2 = 0, \quad \dot{e}_3 = 20 \]

The test signal is a repeated step signal with 2°C amplitude, which is from 29°C to 31°C for iterative learning. Sampling time controller is set as 0.1 second. By changing the speed of fan, Figure 4.18 and 4.19 show the responses of proposed adaptive type-2 PI controller after iterative learning and type-1 PI controller for thermal chamber temperature tracking control when the system is uncertain. It shows that no matter the fan is at low speed or high speed, the proposed adaptive type-2 PI controller can always learn to track the nominal response (mainly the
transient response) regardless of the uncertain condition of fan. On the contrary, the faster the fan spins, the whole closed loop system using type-1 PI controller become more sluggish and thus the transient response deviates more from the desirable response.

![Responses of thermal chamber temperature control](image)

Figure 4.18. Responses of different control systems when the fan speed is 30% of full speed

### 4.8 Conclusions

A framework to realize a type-2 fuzzy PI controller is introduced in this chapter. The merit of the type-2 fuzzy controller is that the equivalent PI parameters are in pre-determined intervals. The controller parameters may be chosen to maintain the performance based on the information about the process uncertainties. To fully harness the potential of the proposed type-2 PI controller, an algorithm for adjusting the centroid used in the type reduction algorithm based on the level of uncertainty is proposed. Simulation results on FOPDT plants show the type-2 fuzzy PI controller can maintain the system performance without any deterioration than traditional PI controller. The proposed algorithm for updating the switch points or centroid makes
Chapter 4. Type-2 Fuzzy PI Controller with Adjustable Type-reduced Output

![Graph showing temperature control](image)

**Figure 4.19.** Responses of different control systems when the fan speed is 80% of full speed

The type-2 PI control scheme is similar to conventional gain-scheduling PI controller since the centroids of the type-2 consequents relate to the controller’s equivalent PI gains. A comparison study is then carried out on controlling a nonlinear uncertain system, pH neutralization process with uncertain parameters. The simulation results show that the proposed type-2 PI controller performs better than conventional gain-scheduling PI controller to track the reference when the parameters change for the pH neutralization process. It shows the potential of the proposed type-2 PI controller to control some nonlinear uncertain systems. Experiment results also show that the type-2 PI controller can maintain the desirable temperature of a thermal chamber whose system parameters are uncertain comparing with its type-1 counterpart.
Chapter 5

On-line Learning Algorithm for Type-2 Fuzzy-Neural Controller

The Representation Theorem in [63] shows that a type-2 fuzzy set actually comprises many embedded type-2 fuzzy set. For an interval type-2 FLS, the output set is composed of the outputs of its embedded type-1 FLSs. The interval type-2 FLS has the potential to handle uncertainties inside the systems by choosing a suitable control surface within the range of possible control output surfaces generated by the innate features of type-2 fuzzy sets and FLSs. Chapter 4 proposed a novel way to change the switch points used in the calculation of center-of-sets type-reduction, thereby producing a variable control surface to minimize the effect of uncertainty and maintain the performance of the system.

On the other hand, the type-2 fuzzy set is a three-dimensional fuzzy set as defined in Chapter 2. It is this extra freedom in the fuzzy set that provides the type-2 FLS with more freedom to model complex input-output relationship. Fuzzy basis function has been proposed as a mathematical formula to describe the mapping relationship from a crisp input $x$ into a crisp output $y = f(x)$ for a type-1 FLS[93]. Through the fuzzy basis function, it is proved that for a singleton type-1 FLS that uses product composition, product implication and height defuzzification, the FLS can uniformly approximate any real continuous non-linear function to an arbitrary
degree of accuracy[93]. The degree of freedom in a FLS, namely the number of inputs, the number of rules and the number of fuzzy sets for each input variable, will control the accuracy of the approximation modelling for a FLS. Hence, an interval type-2 FLS may achieve similar approximation accuracy with less number of fuzzy sets and rules compared with its type-1 counterpart, since the interval type-2 fuzzy sets inside itself offer more degrees of freedom to model complex relationship.

The motivation of this Chapter is to examine the modelling ability or approximation accuracy of an interval type-2 FLS using less rules and fuzzy sets, and then comparing the performance with a type-1 FLS with more rules. Since the use of multi-layered feedforward neural networks are very popular now to synthesize and automate the design of a general fuzzy logic control system when there are many parameters to be determined, the work in this Chapter is based on an higher level fuzzy-neural system, the type-2 fuzzy-neural controller (T2FNC). A brief review on type-1 and type-2 fuzzy-neural system will be provided in next section. Section 5.2 then shows the architecture of T2FNC used in this Chapter. After that, Section 5.3 shows the control scheme using the T2FNC which can on-line update the system parameters. Section 5.4 derives the update rules for weights and parameters of type-2 fuzzy sets of T2FNC with general rule base. Simulation results are shown in Section 5.5 using the T2FNC to control a severe nonlinear plant, pH neutralization process. Experiment results are also shown in Section 5.6 and the test bed is a thermal chamber with uncertain disturbance from a fan. Finally, conclusion is draw in Section 5.7.

5.1 Type-1 and Type-2 Fuzzy-Neural Systems—

General Background

The publication of Professor J.J.Hopfield’s seminal work on neural networks with symmetric weights in the early 1980’s[22] has started the modern era of neural
networks. A great amount of literature has been published that deals with the ability of neural networks to classify, store, recall and associate information or patterns. The performance of neural networks depends on the computational function of the neurons in the network, the structure and topology of the network, and the learning rule or the update rule for the connecting weights. The publication of the “Back-propagation Algorithm” by Rumelhart, Hinton, and William[82] has further extended the learning capability and improved the learning ability of neural networks. This concept of trainable neural networks provides the motivation for utilizing the learning ability of neural networks to learn the fuzzy control rules and the membership functions of a fuzzy logic control system[47].

The idea of fuzzy-neural system transforms the burden of the tedious design problem of fuzzy logic control systems to the training/learning of neural networks. It brings the low-level computational power and learning of neural networks into fuzzy logic systems, and provides the high level IF-THEN rule thinking and reasoning of fuzzy logic systems into neural networks. This integration approach combines both the benefits of neural networks and fuzzy logic systems. That is, the neural networks provide the connectionist structure (fault tolerance and distributed representation properties) and the learning abilities to the fuzzy logic systems, and the fuzzy logic systems provide a structural framework with high-level fuzzy IF-THEN rules humanlike thinking and reasoning to the neural network.

There are many research works which are related to the fuzzy-neural systems using type-1 fuzzy logic since the introduction of the integration approach for fuzzy logic and neural networks. However, there are not many applications that integrate type-2 fuzzy logic and neural networks. As reviewed in Chapter 1, Melin and Castillo designed an adaptive controller of non-linear plants using Type-2 fuzzy logic and neural networks[56]. Lee and Lin applied type-2 fuzzy neural systems with adaptive filter to nonlinear uncertain systems[39]. Singh and et al also proposed a type-2 fuzzy neural model based controller for a nonlinear system[85].

Wang, Chen and Lee developed a type-2 fuzzy neural network for the truck
backing-up control that yielded better performance than those using type-1 fuzzy neural network [91]. The authors developed a dynamical learning rate to update the weights and maximize the error reduction of the BP method. However, due to variations in the initial membership function parameters, the performance of back propagation training process is sensitive to the manner in which the parameters are initialized. Furthermore, a genetic algorithm (GA) is used to search for the optimal parameters of the membership functions (MFs) or FOU shapes to achieve better total performance. This means the learning of the membership function is an off-line one-pass method and based on the initial condition of the system. The use of uncertainty bounds in the design of embedded real-time type-2 neuro-fuzzy speed controller for marine diesel engines has also been recently proposed [50]. The main contribution of the study is to use an approximation algorithm for type-reduction and thus this simpler method made it possible to derive a BP update algorithm for parameters of membership functions. Hence, the work only utilized a simplified update algorithm to design the type-2 neural-fuzzy controller, and does not update its parameters on-line. Hence, both of these works of type-2 fuzzy-neural systems are only for off-line learning. The type-2 fuzzy neural network with the FOU determined by running the GA or simplified parameter update algorithm one-time may not handle the task well when the working condition of the system changes. Hence, an on-line learning algorithm for updating the parameters of membership functions (MF variables) or reshaping the FOUs is required to continuously develop suitable FOU shapes to minimize the effects of shifts in the working condition of the uncertain systems. This is the motivation for introducing an on-line update algorithm for MF variables when using type-2 fuzzy-neuro systems with original center-of-sets type-reduction.

5.2 Architecture of type-2 FNC

A fuzzy-neural network is an implementation of a fuzzy inference procedure in the framework of a neural network. Briefly, a neural network consists of several succes-
sive layers of interconnected nodes where a node in any particular layer performs a specific function in relation to the overall network. Figure 5.1 shows the structure of a typical type-2 FNC which mimics the operation of the type-2 fuzzy controller. Due to the complexity of general type-2 FLSs, an interval type-2 fuzzy-neural network is used in this Chapter.

Unlike ordinary neural networks, a fuzzy-neural network has components of a fuzzy set present in it. The layers of the network, when properly interconnected, implement the fuzzification-inference-defuzzification stages of a fuzzy controller. The fuzzy-neural system in this Chapter is a $n$-input-1-output system. Layer I performs the singleton fuzzification step on the input vector $X = [X_1, X_2, \cdots, X_n]^T$. For a particular input $X_i$, its type-2 antecedents using the type-2 triangles with uncertain base are shown in Figure 5.2 (Appendix A shows the reason why type-2 sets with
uncertain base are used in this Chapter). It is assumed that \( N_j \) fuzzy sets are used to describe the domain for the input \( X_i \). The membership function nodes in Layer II are type-2 fuzzy sets may be constructed by shifting a type-1 fuzzy set to the left or right. The amount shifted is governed by “MF variables” which is defined as the variables, e.g. \( \bar{U}_{L}^{X_j,2} \) and \( \bar{U}_{L}^{X_j,2} \) shown in Figure 5.2. The dashed triangle is termed as type-1 principle fuzzy set. The triangular type-2 fuzzy set reduces to the type-1 principle set when MF variables/FOU reduce to zero. The type-2 fuzzy set can be constructed by shifting the two end points at the base of the type-1 principle fuzzy set to the left or right and the distance is determined by the value of related MF variable. The left and right side of type-2 membership functions are given independent freedom for update. That means the MF variables at the left side and right side can be independently updated. Hence, as shown in Figure 5.2, the left side MF variables are denoted as \( \bar{U}_{L}^{X_j,p} \) and \( \bar{U}_{L}^{X_j,p} \), while the right side as \( \bar{U}_{R}^{X_j,p} \) and \( \bar{U}_{R}^{X_j,p} \) \((p = 1 \cdots N_j)\). \((\bar{U} \) means the MF variable to determine the upper membership function, and \( U \) means the MF variable to determine the lower membership function. On the other hand, the subscript \( L \) added to \( U \) or \( U \) means the MF variable is used to construct the left side of the triangular type-2 fuzzy set and the right side for subscript \( R \).)

![Figure 5.2. Creation of Footprint of Uncertainty (FOU) from type-1 fuzzy sets](image)

The nodes in Layers III corresponds to the inference or rule evaluation layer. A Layer III node accepts as inputs the preconditions (antecedents) for each rule and combines them according to some pre-specified operation (usually minimum or
product). This essentially corresponds to evaluating the antecedent portion of the rule consisting of the AND connectives. The outputs from these nodes are referred to as the firing levels. In the case of interval type-2 inference, each node will output an interval firing set whose bounds are the individual firing levels for the upper and lower membership functions forming the boundary of the FOU on the input sets.

The rule base of a general type-2 FNC can be expressed as ($i = 1 \cdots M$):

$$Rule\ i:\ IF\ X_1\ is\ \tilde{F}_{1,i}^o\ and\ X_2\ is\ \tilde{F}_{2,i}^o\ \cdots\ and\ X_n\ is\ \tilde{F}_{n,i}^o\ THEN\ Y\ is\ z_i.$$

(5.1)

Each input in a fuzzy rule base is associated with a unique label $\tilde{F}_{j,i}^o$. It is because a fuzzy set $\tilde{X}_{j,p}$ ($p = 1 \cdots N_j$) describing the input $X_j$ may appear in the antecedent part of different rules of a common fuzzy control system. Thus, it is better to distinguish them using different notations in the rule base before the relationship is clearly known. In other words, the labels $\tilde{F}_{j,1}^o, \cdots, \tilde{F}_{j,M}^o$ for the the $j$th input in the $M$ rules individually denote one of the fuzzy sets ($\tilde{X}_{j,p}$ ($p = 1 \cdots N_j$)) for $j$th input while $M > N_j$:

$$\tilde{F}_{j,i}^o \in \{\tilde{X}_{j,1}, \tilde{X}_{j,2}, \cdots, \tilde{X}_{j,N_j}\}, \ j \in [1, \cdots, n], \ i \in [1, \cdots, M], \ M = N_1 \times N_2 \times \cdots \times N_j$$

(5.2)

Using the product t-norm, the firing set associated with the $i$th rule is:

$$\tilde{f}_i^o = \mu_{\tilde{F}_{1,i}^o} (X_1) \times \cdots \times \mu_{\tilde{F}_{n,i}^o} (X_n)$$

(5.3)

$$\tilde{f}_i^o = \mu_{\tilde{F}_{1,i}^o} (X_1) \times \cdots \times \mu_{\tilde{F}_{n,i}^o} (X_n)$$

(5.4)

Layer IV performs type-reduction and its output is a type-reduced set. The links between the third and fourth layer are the consequent fuzzy sets $z_i$, ($i = 1 \cdots M$). To lower the computation complexity of the type-2 fuzzy-neural system, singletons are used to characterize the output domain. The weighting step produces an interval type-2 output set for each rule. The combined output set resulting from the union
of all rules will finally be an interval type-reduced set \([u_l, u_r]\) according to the center-of-sets type-reduction algorithm\([57]\).

\[ u_l = \frac{\sum_{i=1}^{L} f_i^o z_i + \sum_{i=L+1}^{M} f_i^o z_i}{\sum_{i=1}^{L} T_i^o + \sum_{i=L+1}^{M} f_i^o} \]  

\[ u_r = \frac{\sum_{i=1}^{R} f_i^o z_i + \sum_{i=R+1}^{M} T_i^o z_i}{\sum_{i=1}^{R} f_i^o + \sum_{i=R+1}^{M} T_i^o} \]  

(5.5)  

(5.6)

where \(L\) and \(R\) are the switch points that partition the consequent weight space in a manner which allows the weights on either side to be scaled by the appropriate upper or lower firing strength.

Finally, Layer V is the defuzzification step which takes the average of the lower and upper bounds to produce the crisp control output \(u_f\).

\[ u_f = \frac{1}{2}(u_l + u_r) \]  

(5.7)

5.3 Control Scheme of Type-2 Fuzzy-Neural Control System

Before deriving the self-learning algorithm, the control scheme of a type-2 fuzzy-neural controller (FNC) which uses the BP method to update the weights and MF variables will be briefly introduced first. The self-learning control scheme, as shown in Figure 5.3, is adopted from [88]. It consists of three principle components: (i) a reference model or filter, (ii) a proportional controller in a feedback configuration, (iii) a type-2 FNC in a feedforward configuration. The function of the reference filter is to specify a desired trajectory for the system output, but more importantly to ensure that unreasonable or unattainable requests are not made of the adaptation
algorithm for the type-2 fuzzy controller, in pursuit of trajectories incompatible with the controller dynamics and control action range.

![Diagram: Feed-forward feedback FNC](image)

Figure 5.3. Feed-forward feedback FNC

The key module of the self-learning controller is a type-2 fuzzy-neural system that serves as a feedforward controller. The objective is to train the type-2 fuzzy-neuro system in the feedforward path such that it eventually represents an inverse mapping between the output of the reference model and the plant output. On accepting the desired filtered input from the reference model, the type-2 FNC will learn to predict the control signal required to drive the plant to the desired state along the trajectory.

To train the feedforward controller, the control signal that should be applied to the plant for its output to reach a desired state is required. Since this information is usually unknown a priori, the feedback error learning strategy [34] is employed to estimate the required control action, $u_d(t)$:

$$u_d(t) - u_f(t - t_d) = \gamma e(t), \quad \gamma \text{ is constant}$$

(5.8)

where $u_f(t - t_d)$ is the output of the feedforward controller $t_d$ sampling instants earlier and $\gamma$ is the learning rate. This learning rule is motivated by the observation that the feedback error at sampling instant $t$ is due to the erroneous control action
administered to the plant with dead-time \( t_d \) sampling instants earlier. Consequently, the feedback error is a reflection of the control error and may be used to estimate the control action that will drive the process output to the desired value. Since a perfect inverse mapping is rarely achieved in practice for such a system, the proportional controller is introduced in the feedback path to reduce the difference between the reference and plant output that arises from model mismatch. As the self learning mechanism improves the model following capability, the set point is tracked more accurately, progressively reducing the feedback error. This has the effect of eventually shutting off the control action contribution from the proportional controller when the type-2 FNC has learned its parameters well enough to entirely provide the correct feedforward control action. The proportional controller is thus an aid in making the learning procedure robust.

### 5.4 On-line Self-learning Algorithm for MF Variables and Weights

Generally speaking, when there is enough information or expert knowledge for a control system, the desired control surface can be derived. For a system which is severely nonlinear, more fuzzy sets may be needed to generate the complex control surface. The manual tuning of MF variables is not an easy job. The back propagation (BP) learning method is commonly adopted to tune the parameters of neural networks. The BP method minimizes the difference between the desired and actual outputs through iterative learning. The most important and challenging part of the BP method is computing the derivatives that are needed. Although derivative calculations that are applicable to any kind of type-2 membership functions have been developed [60], the derivations assume that the parameters to be tuned were different for each rule and for each antecedent and consequent, which is not a common case in fuzzy controllers. Hence, this section shows the case of calculation of deriva-
tives and simplifies them when parameters are shared across rules. Subsequently, the on-line self-learning algorithm is proposed to update both consequent weights and parameters of antecedents or their FOU.

Another difficulty of the derivative calculation is that the type-reduction always requires the weights of the rules to be arranged ascending order. However, during the update of weights of the rule-base, the values of weights may be always changing, so it is necessary to reorder the weights after each update. Since rule reordering is required each time when the consequent weights are updated, the relationship of rule number $i$ ($i = 1, \ldots, M$) and the membership function of antecedents and its associated FOU variables are changed due to the index change. Hence, the reordered weights are denoted using $w_i$ and the reordered type-2 FNC can be represented using another sets of notations where $w_1 \leq \cdots \leq w_M$:

Rule $i$ : IF $X_1$ is $\tilde{F}_{1,i}$ and $X_2$ is $\tilde{F}_{2,i}$ and ... and $X_n$ is $\tilde{F}_{n,i}$ THEN $Y$ is $w_i$. \hfill (5.9)

Notice by comparing (5.1) and (5.9) that the antecedents after reordering, $\tilde{F}_{j,i}$, may not be the same fuzzy set as the one in the original rule-base, $\tilde{F}_{j,i}^o$.

To track the reordering procedure, an ordering indicator vector $I$ is introduced which contain index of $z_i$ arranged in ascending order. Hence, a permutation matrix $Q$ may be used to track the reordering:

$$w = Qz, \quad Q(i, I(i)) = 1, \quad and \quad Q(i, j) = 0, \quad (j \neq I(i)) \quad (i \in [1, \cdots, M])$$

For instance, after reordering a FLS with four rules, if the ordered weights are
$z_3 \leq z_2 \leq z_1 \leq z_4$, then the procedure is tracked as:

\[
I = \begin{bmatrix} 3 \\ 2 \\ 1 \\ 4 \end{bmatrix}, \quad Q = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = QZ = \begin{bmatrix} z_3 \\ z_2 \\ z_1 \\ z_4 \end{bmatrix}
\]

The relationship between the original and reordered firing level and membership grades are:

\[
f = Qf^o, \quad \tilde{f} = Q\tilde{f}^o, \quad \Psi_{\tilde{X}_j} = Q\Psi_{\tilde{X}_j}^o, \quad \overline{\Psi}_{\tilde{X}_j} = Q\overline{\Psi}_{\tilde{X}_j}^o
\]

where the firing level vectors and membership grade vectors are defined as:

\[
f = \begin{bmatrix} f_1 \\ \vdots \\ f_M \end{bmatrix}, \quad \tilde{f} = \begin{bmatrix} \tilde{f}_1 \\ \vdots \\ \tilde{f}_M \end{bmatrix}, \quad \Psi_{\tilde{X}_j} = \begin{bmatrix} \mu_{\tilde{F}_{j,1}} \\ \vdots \\ \mu_{\tilde{F}_{j,M}} \end{bmatrix}, \quad \overline{\Psi}_{\tilde{X}_j} = \begin{bmatrix} \overline{\mu}_{\tilde{F}_{j,1}} \\ \vdots \\ \overline{\mu}_{\tilde{F}_{j,M}} \end{bmatrix}
\]

\[
f^o = \begin{bmatrix} f_1^o \\ \vdots \\ f_M^o \end{bmatrix}, \quad \tilde{f}^o = \begin{bmatrix} \tilde{f}_1^o \\ \vdots \\ \tilde{f}_M^o \end{bmatrix}, \quad \Psi_{\tilde{X}_j}^o = \begin{bmatrix} \mu_{\tilde{F}_{j,1}}^o \\ \vdots \\ \mu_{\tilde{F}_{j,M}}^o \end{bmatrix}, \quad \overline{\Psi}_{\tilde{X}_j}^o = \begin{bmatrix} \overline{\mu}_{\tilde{F}_{j,1}}^o \\ \vdots \\ \overline{\mu}_{\tilde{F}_{j,M}}^o \end{bmatrix}
\]

### 5.4.1 Weight update rules

The back-propagation learning algorithm, also called gradient descent update rule, has already been widely applied to update the weights of a general fuzzy-neural system. In this subsection, the derivation of an on-line back-propagation learning algorithm for updating the consequent weights is shown first and followed by the algorithm for MF variables in next subsection.
First of all, the cost function is defined as:

\[ J(t) = \frac{1}{2} e^2(t) \]  \hspace{1cm} (5.10)

where the error \( e \) is defined as the difference between the reference signal \( y_{\text{ref}} \) and the actual plant output \( y \):

\[ e(t) = y_{\text{ref}}(t) - y(t) \]  \hspace{1cm} (5.11)

The weight update rule aims to update only the consequent weights so that the cost function is minimized. In terms of the reordered consequent weight vector \( w \), the weight update rule is described as:

\[ \frac{\partial J}{\partial w} = e \cdot \frac{\partial e}{\partial w} = e \cdot \frac{\partial [y_{\text{ref}}(t) - y(t)]}{\partial w} \]  \hspace{1cm} (5.12)

\[ \frac{\partial J}{\partial w} = -e \cdot \frac{\partial y(t)}{\partial w} \]  \hspace{1cm} (5.13)

Equation (5.12) indicates that minimization of cost function \( J(t) \) depends on information about the current plant output \( y(t) \). However, since the zero-order hold sampling method is used and the dead-time of the system introduces additional delay, the current output \( y(t) \) is not available. In order to solve the problem, the error \( e \) is regarded as from the difference between the correct control action \( u_d \) and the actual control action \( u_f \) of \( t_d \) samplings earlier. It is reasonable thus to define the difference as function of the error:

\[ u_d - u_f(t - t_d) = F(e(t)) \]  \hspace{1cm} (5.14)

Using the Equation (5.8), it is assumed that the control error is proportional to the model error \( e \):

\[ u_d - u_f(t - t_d) = \gamma(e(t)), \]  \hspace{1cm} (5.15)

where \( \gamma \) is a proportional gain. Thus, the weight update rule as (5.12) can be
rewritten as:

\[
\frac{\partial J}{\partial w} = e \cdot \frac{\partial e}{\partial w} = \frac{1}{\gamma} \cdot \frac{\partial [u_d - u_f(w, t - t_d)]}{\partial w} \quad (5.16)
\]

\[
\frac{\partial J}{\partial w} = -\frac{1}{\gamma} e \cdot \frac{\partial u_f(w, t - t_d)}{\partial w} \quad (5.17)
\]

Since in interval type-2 FLS, the output \( u_f \) is the mid-point of interval output set \([u_l, u_r]\), then:

\[
\frac{\partial J}{\partial w} = -\alpha \cdot e(t) \cdot \frac{\partial (u_l + u_r)}{\partial w}, \quad \alpha = \frac{1}{2\gamma} \quad (5.18)
\]

\[
\frac{\partial J}{\partial w} = -\alpha \cdot e(t) \cdot \left( \frac{\partial u_l}{\partial w} + \frac{\partial u_r}{\partial w} \right), \quad \alpha = \frac{1}{2\gamma} \quad (5.19)
\]

The above equations indicate that the amount by which the weights need to be updated depends on \( \left( \frac{\partial u_l}{\partial w} + \frac{\partial u_r}{\partial w} \right) \).

The type-2 FNC uses fuzzy singletons for the weights, thus using center-of-sets type-reduction algorithm with Karnik-Mendel method to identify the switch points, the left bound and right bound of the output set can be expressed using Equation (2.64):

\[
u_l = \frac{\sum_{i=1}^{L} f_i w_i + \sum_{i=L+1}^{M} f_i w_i}{\sum_{i=1}^{L} f_i + \sum_{i=L+1}^{M} f_i} = \frac{\sum_{i=1}^{L} f_i w_i + \sum_{i=L+1}^{M} f_i w_i}{N_l} \quad (5.20)
\]

\[
u_r = \frac{\sum_{i=1}^{R} f_i w_i + \sum_{i=R+1}^{M} f_i w_i}{\sum_{i=1}^{R} f_i + \sum_{i=R+1}^{M} f_i} = \frac{\sum_{i=1}^{R} f_i w_i + \sum_{i=R+1}^{M} f_i w_i}{N_r} \quad (5.21)
\]
where

\[ N_l = \sum_{i=1}^{L} f_i + \sum_{i=L+1}^{M} \bar{J}_i \quad (5.22) \]
\[ N_r = \sum_{i=1}^{R} f_i + \sum_{i=R+1}^{M} \bar{J}_i \quad (5.23) \]

are the normalization factors.

Define the factors as

\[ q_{l_i} = \frac{f_i}{N_l}, \quad q_{l_i} = \frac{\bar{J}_i}{N_l}, \quad q_{r_i} = \frac{f_i}{N_r}, \quad q_{r_i} = \frac{\bar{J}_i}{N_r} \quad (5.24) \]

Thus, the left bound and right bound of the output set can be rewritten as:

\[ u_l = \sum_{i=1}^{L} q_{l_i} w_i + \sum_{i=L+1}^{M} q_{l_i} w_i \quad (5.25) \]
\[ u_r = \sum_{i=1}^{R} q_{r_i} w_i + \sum_{i=R+1}^{M} q_{r_i} w_i \quad (5.26) \]

Expressing the factors in vector form:

\[ q_l = [q_{l_1}, \cdots, q_{l_L}, q_{l_{L+1}}, \cdots, q_{l_M}] \quad (5.27) \]
\[ q_r = [q_{r_1}, \cdots, q_{r_R}, q_{r_{R+1}}, \cdots, q_{r_M}] \quad (5.28) \]

the partial derivatives of \( u_l \) and \( u_r \) with respect to the weight vector is just

\[ \left( \frac{\partial u_l}{\partial \mathbf{w}} + \frac{\partial u_r}{\partial \mathbf{w}} \right) = (q_l + q_r) \quad (5.29) \]

Hence, the gradient descent update rule for weight \( w_i \) is:

\[ \mathbf{w}(t) = \mathbf{w}(t-1) - \eta \cdot \frac{\partial J}{\partial \mathbf{w}} \quad (5.30) \]
\[ \mathbf{w}(t) = \mathbf{w}(t-1) + \frac{\eta}{2\gamma} \cdot e(t) \cdot \left( \frac{\partial u_l}{\partial \mathbf{w}} + \frac{\partial u_r}{\partial \mathbf{w}} \right) \quad (5.31) \]
Finally, the gradient descent weight update rule for the consequent weight vector \( w \) in the type-2 adaptive fuzzy controller will be:

\[
    \mathbf{w}(t) = \mathbf{w}(t-1) + \delta \cdot e(t) \cdot (\mathbf{q}_l + \mathbf{q}_r), \quad \delta = \frac{\eta}{2\gamma}
\]

In order to obtain the updated weights of original rule base to calculate the output, the inverse permutation matrix \( Q^{-1} \) can be used to order the updated weights back to the sequence as in the original rule-base index.

\[
    \mathbf{z}(t) = Q^{-1} \mathbf{w}(t)
\]

### 5.4.2 MF variables update rules

A full update rule includes the weight update rule presented in the previous section, which means the update rule for consequent weights remains the same. As an extension to the weight update rule, the strategy is to back propagate the model error \( e \) through the fuzzy-neural model to update the MF variables which define the FOU of type-2 fuzzy sets. As discussed previously, one problem that hinders the derivation of the update rule is that the partial derivatives are associated with the MF variables, and after reordering the rules, the fuzzy sets used to describe the antecedents in the \( i \)th rule may be changed, so as the MF variables. This difficulty is overcome by introducing an antecedent indicator to provide the information about how the \( N_j \) fuzzy sets are shared among the antecedents in the rules and thus can simplify the calculation of derivatives.

The antecedent indicator vector used to describe the rule-organization is defined in Equation 5.34. For the input \( X_j \), \( N_j \) fuzzy sets are shared as antecedents across the original rule-base. The \( i \)th element of the vector \( I_{X_j,o}^{X_j,n} \), which is a certain value \( p' \) where \( p' \in [1, \cdots, N_j] \), indicates that the fuzzy set \( X_{j,p'} \) is used as the antecedent for input \( X_j \) in the \( i \)th rule. The number of the elements in the antecedent indicator vector equals to the number of rules \( M \). For a particular input \( X_j \), its antecedent
indicator vector $I_{X,j,o}^U$ may be divided into one block ($j = 1$) or $\prod_{i=j+1}^{n} N_i$ ($j \neq 1$) blocks. Each block has the same elements and all the elements are individually repeated $N_i$ ($j \neq n$) times or just one time ($j = n$).

$$I_{X,j,o}^U = \begin{bmatrix} \vdots \\ \vdots \\ \vdots \\ 1 \\ \vdots \\ \vdots \\ 1 \\ \vdots \\ \vdots \\ \vdots \end{bmatrix} \prod_{i=j+1}^{n} N_i \text{ blocks } (j \neq 1) \quad \begin{bmatrix} 1 \\ \vdots \\ 1 \\ \vdots \\ \vdots \end{bmatrix} \prod_{i=j+1}^{n} N_i \text{ blocks } (j = 1) \quad \begin{bmatrix} \vdots \\ \vdots \\ \vdots \\ N_j \\ \vdots \\ N_j \end{bmatrix} \prod_{i=j+1}^{n} N_i \text{ blocks } (j \neq n) \quad \begin{bmatrix} 1 \\ \vdots \\ \vdots \end{bmatrix} \prod_{i=j+1}^{n} N_i \text{ blocks } (j = n)$$

(5.34)

For example, the relationship for a simple two-input FLS with four rules in which each input domain is characterized by two fuzzy sets can be described as:

$$\begin{align*}
\tilde{F}_{1,1}^o &= \tilde{X}_{1,1}, & \tilde{F}_{1,2}^o &= \tilde{X}_{1,1}, & \tilde{F}_{1,3}^o &= \tilde{X}_{1,2}, & \tilde{F}_{1,4}^o &= \tilde{X}_{1,2} \\
\tilde{F}_{2,1}^o &= \tilde{X}_{2,1}, & \tilde{F}_{2,2}^o &= \tilde{X}_{2,2}, & \tilde{F}_{2,3}^o &= \tilde{X}_{2,1}, & \tilde{F}_{2,4}^o &= \tilde{X}_{2,2}
\end{align*}$$

The indicator vectors will be

$$I_{U}^{X_{1,o}} = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 2 \end{bmatrix}, \quad I_{U}^{X_{2,o}} = \begin{bmatrix} 1 \\ 2 \\ \vdots \\ 2 \end{bmatrix}$$

As another example, the relationship for a simple two-input FLS with nine rules
in which each input domain has three fuzzy sets is shown below:

\[
\begin{align*}
\bar{F}_{o,1}^1 &= \bar{X}_{1,1}, & \bar{F}_{o,2}^1 &= \bar{X}_{1,2}, & \bar{F}_{o,3}^1 &= \bar{X}_{1,3}, & \bar{F}_{o,4}^1 &= \bar{X}_{1,4}, & \bar{F}_{o,5}^1 &= \bar{X}_{1,5}, \\
\bar{F}_{o,6}^1 &= \bar{X}_{1,6}, & \bar{F}_{o,7}^1 &= \bar{X}_{1,7}, & \bar{F}_{o,8}^1 &= \bar{X}_{1,8}, & \bar{F}_{o,9}^1 &= \bar{X}_{1,9}, \\
\bar{F}_{o,1}^2 &= \bar{X}_{2,1}, & \bar{F}_{o,2}^2 &= \bar{X}_{2,2}, & \bar{F}_{o,3}^2 &= \bar{X}_{2,3}, & \bar{F}_{o,4}^2 &= \bar{X}_{2,4}, & \bar{F}_{o,5}^2 &= \bar{X}_{2,5}, & \bar{F}_{o,6}^2 &= \bar{X}_{2,6}, & \bar{F}_{o,7}^2 &= \bar{X}_{2,7}, & \bar{F}_{o,8}^2 &= \bar{X}_{2,8}, & \bar{F}_{o,9}^2 &= \bar{X}_{2,9}, \\
\end{align*}
\]

The indicator vectors will be

\[
\begin{align*}
I_{U}^{X_1,o} &= \begin{bmatrix} 1 \\ 1 \\ 1 \\ 2 \\ 2 \\ 2 \\ 3 \\ 3 \end{bmatrix}, & I_{U}^{X_2,o} &= \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \\ 2 \\ 3 \\ 1 \\ 2 \\ 3 \end{bmatrix}
\end{align*}
\]

After reordering of the rule base, the sharing relationship between the antecedents \((X_{j,p}^o \text{ and } \bar{F}_{j,i}^o)\) is also changed. Using the permutation matrix, the reordered antecedent indicator vector can be obtained as:

\[
I_{U}^{X_j} = QI_{U}^{X_j,o}
\]  

(5.35)

**Update rules for upper MF variables \(U\)**

The update rules for the upper MF variables will be presented now. The original MF variable vector for input \(X_j\) is defined in Equation (5.36). Each element in the
vector is associated with a particular antecedent set in the rule base:

\[
\mathbf{U}^o_{X_j} = \begin{bmatrix} \mathbf{U}^{F_{j,1}}_o \\ \vdots \\ \mathbf{U}^{F_{j,M}}_o \end{bmatrix}, \quad \mathbf{U}^o_{X_j} = \begin{bmatrix} \mathbf{U}^{F_{j,1}}_o \\ \vdots \\ \mathbf{U}^{F_{j,M}}_o \end{bmatrix}
\] (5.36)

After rule reordering, the reordered MF variable vector will become:

\[
\bar{\mathbf{U}}_{X_j} = \begin{bmatrix} \mathbf{U}^{\bar{F}_{j,1}} \\ \vdots \\ \mathbf{U}^{\bar{F}_{j,M}} \end{bmatrix}, \quad \bar{\mathbf{U}}_{X_j} = \begin{bmatrix} \mathbf{U}^{\bar{F}_{j,1}} \\ \vdots \\ \mathbf{U}^{\bar{F}_{j,M}} \end{bmatrix}
\] (5.37)

The elements, \( \mathbf{U}^{\bar{F}_{j,i}}_o \) and \( \mathbf{U}^{\bar{F}_{j,i}}_o \), denotes either the left side variables \( \mathbf{U}^{\bar{F}_{j,i}}_L \) and \( \mathbf{U}^{\bar{F}_{j,i}}_R \) or the right side variables \( \mathbf{U}^{\bar{F}_{j,i}}_L \) and \( \mathbf{U}^{\bar{F}_{j,i}}_R \). Once the inputs are fed into the system, it can be determined that which side of the membership function is fired for every antecedent in the rules. However, the derivation of update rules for MF variables only requires the values of membership grades and its derivatives and the procedure is actually not dependent on which side is fired. Hence, the following derivation is valid for both left side and right side cases regardless of the side subscriptions “\( L \)” and “\( R \)”, but during implementation the information has to be processed at the beginning and thus the side-related MF variables can be chosen to be updated.

The derivation of update rule for MF variables is similar to that of weight update rule. The partial derivative of \( J \) with respect to \( \mathbf{U}^{\bar{F}_{j,i}}_o \) in the \( i \)th rule for MF variable of input \( X_j \) is:

\[
\frac{\partial J}{\partial \mathbf{U}^{\bar{F}_{j,i}}_o} = -\alpha \cdot e(t) \cdot \left( \frac{\partial u_l}{\partial \mathbf{U}^{\bar{F}_{j,i}}_o} + \frac{\partial u_r}{\partial \mathbf{U}^{\bar{F}_{j,i}}_o} \right), \quad \alpha = \frac{1}{2\gamma}
\] (5.38)
The partial derivative term can be expanded by using Equation (5.25) and (5.26):

\[
\left( \frac{\partial u_l}{\partial\bar{U}^{F_{j,i}}} + \frac{\partial u_r}{\partial\bar{U}^{F_{j,i}}} \right) = \sum_{a=1}^{L} w_a \frac{\partial \bar{q}^l_a}{\partial\bar{U}^{F_{j,i}}} + \sum_{b=L+1}^{M} w_b \frac{\partial \bar{q}^l_b}{\partial\bar{U}^{F_{j,i}}} + \sum_{c=1}^{R} w_c \frac{\partial \bar{q}^r_c}{\partial\bar{U}^{F_{j,i}}} + \sum_{d=R+1}^{M} w_d \frac{\partial \bar{q}^r_d}{\partial\bar{U}^{F_{j,i}}} \tag{5.39}
\]

The partial derivative of each element \( \bar{q}^l, \bar{q}^r \) with respect to \( \bar{U}^{F_{j,i}} \) are required.

\[
\frac{\partial \bar{q}^l}{\partial\bar{U}^{F_{j,i}}} = \left( \frac{1}{N^2_i} \right) \left( \frac{\partial \bar{f}^l_i}{\partial\bar{U}^{F_{j,i}}} N_i - \bar{f}^l_i \frac{\partial N_i}{\partial\bar{U}^{F_{j,i}}} \right), \quad h \in [1, \cdots, L] \tag{5.40}
\]

\[
\frac{\partial \bar{q}^r}{\partial\bar{U}^{F_{j,i}}} = \left( \frac{1}{N^2_i} \right) \left( \frac{\partial \bar{f}^r_i}{\partial\bar{U}^{F_{j,i}}} N_i - \bar{f}^r_i \frac{\partial N_i}{\partial\bar{U}^{F_{j,i}}} \right), \quad h \in [L + 1, \cdots, M] \tag{5.41}
\]

\[
\frac{\partial \bar{q}_{l,i}}{\partial\bar{U}^{F_{j,i}}} = \left( \frac{1}{N^2_i} \right) \left( \frac{\partial \bar{f}_{l,i}}{\partial\bar{U}^{F_{j,i}}} N_{r} - \bar{f}_{l,i} \frac{\partial N_{r}}{\partial\bar{U}^{F_{j,i}}} \right), \quad h \in [R + 1, \cdots, M] \tag{5.42}
\]

\[
\frac{\partial \bar{q}_{r,i}}{\partial\bar{U}^{F_{j,i}}} = \left( \frac{1}{N^2_i} \right) \left( \frac{\partial \bar{f}_{r,i}}{\partial\bar{U}^{F_{j,i}}} N_{r} - \bar{f}_{r,i} \frac{\partial N_{r}}{\partial\bar{U}^{F_{j,i}}} \right), \quad h \in [1, \cdots, R] \tag{5.43}
\]

As \( \bar{U}^{F_{j,i}} \) is associated only with the upper firing level \( \bar{f} \), the partial derivative can be simplified as:

\[
\frac{\partial \bar{q}^l}{\partial\bar{U}^{F_{j,i}}} = \left( \frac{1}{N^2_i} \right) \left( \frac{\partial \bar{f}^l_i}{\partial\bar{U}^{F_{j,i}}} N_i - \bar{f}^l_i \sum_{k=1}^{L} \frac{\partial \bar{f}^l_k}{\partial\bar{U}^{F_{j,i}}} \right), \quad h \in [1, \cdots, L] \tag{5.44}
\]

\[
\frac{\partial \bar{q}^r}{\partial\bar{U}^{F_{j,i}}} = \left( \frac{1}{N^2_i} \right) \left( -\bar{f}^l_i \sum_{k=1}^{L} \frac{\partial \bar{f}^l_k}{\partial\bar{U}^{F_{j,i}}} \right), \quad h \in [L + 1, \cdots, M] \tag{5.45}
\]

\[
\frac{\partial \bar{q}_{l,i}}{\partial\bar{U}^{F_{j,i}}} = \left( \frac{1}{N^2_i} \right) \left( \frac{\partial \bar{f}_{l,i}}{\partial\bar{U}^{F_{j,i}}} N_{r} - \bar{f}_{l,i} \sum_{k=R+1}^{M} \frac{\partial \bar{f}_{l,k}}{\partial\bar{U}^{F_{j,i}}} \right), \quad h \in [R + 1, \cdots, M] \tag{5.46}
\]

\[
\frac{\partial \bar{q}_{r,i}}{\partial\bar{U}^{F_{j,i}}} = \left( \frac{1}{N^2_i} \right) \left( -\bar{f}_{l,i} \sum_{k=R+1}^{M} \frac{\partial \bar{f}_{l,k}}{\partial\bar{U}^{F_{j,i}}} \right), \quad h \in [1, \cdots, R] \tag{5.47}
\]

Using product t-norm, then for the \( i \)th rule, \( \bar{f}_i = \bar{p}_{F_{1,i}} \times \cdots \times \bar{p}_{F_{n,i}} \). The \( \bar{U}^{F_{j,i}} \) is only associated with \( \bar{p}_{F_{j,i}} \) for input \( X_j \) after rule reordering. Hence, the partial derivative
can be revised as:

\[
\frac{\partial q_h^l}{\partial U_{F_j,i}} = \left( \frac{1}{N_l^2} \right) \left( \frac{\partial \overline{F}_{j,k}^l}{\partial U_{F_j,i}} \cdot ( \prod_{N=1, N \neq j}^{n} \overline{F}_{N,h}^i ) \cdot N_l - \frac{f_h}{n} \sum_{k=1}^{L} \left( \frac{\partial \overline{F}_{j,k}^l}{\partial U_{F_j,i}} \cdot ( \prod_{N=1, N \neq j}^{n} \overline{F}_{N,h}^i ) \right) \right),
\]

\[
\frac{\partial q_h^r}{\partial U_{F_j,i}} = \left( \frac{1}{N_r^2} \right) \left( \frac{\partial \overline{F}_{j,k}^r}{\partial U_{F_j,i}} \cdot ( \prod_{N=1, N \neq j}^{n} \overline{F}_{N,h}^i ) \cdot N_r - \frac{f_r}{M} \sum_{k=R+1}^{M} \left( \frac{\partial \overline{F}_{j,k}^r}{\partial U_{F_j,i}} \cdot ( \prod_{N=1, N \neq j}^{n} \overline{F}_{N,h}^i ) \right) \right),
\]

Since the term \( \sum_k \left( \frac{\partial \overline{F}_{j,k}^l}{\partial U_{F_j,i}} \cdot ( \prod_{N=1, N \neq j}^{n} \overline{F}_{N,h}^i ) \right) \) requires information about whether the \( k \)th rule after reordering shares the same antecedent set for input \( X_j \) as the \( i \)th rule or is related to the reordered MF variable \( U_{F_j,i}^j \), the antecedent indicator vector defined in Equation (5.34) for the original rule-base \( I_{U}^X \) can be utilized to solve the problem. After the reordering, the reordered antecedent indicator vector can be obtained by using the permutation matrix \( Q \) as, \( I_{U}^X = Q I_{U}^X \). The reordered indicator vectors after reordering can help to simplify derivation of the partial derivative.

As the calculation of derivative of membership function with respect to the MF variable, a simple notation can be defined to provide a concise rule:

\[
\frac{\partial \overline{F}_{j,i}^l}{\partial U_{F_j,i}} = \overline{F}_{j,i}^l(X_j) \quad (5.48)
\]

The derivatives in Equation (5.48) are functions of inputs and MF variables, so the values of these derivatives are available before reordering. The derivative vectors for original rule base are defined as:

\[
\Psi_{X_j}^o = \begin{bmatrix}
\overline{F}_{j,i}^o \\
\vdots \\
\overline{F}_{j,M}^o
\end{bmatrix} 
\]
where $\mu_{\tilde{F}_{j,i}}'(X_j) = \frac{\partial \mu_{\tilde{F}_{j,i}}'(X_j)}{\partial U_{F_{j,i}}}$, which is based on the fuzzy set $\tilde{F}_{j,i}$ in the original rule-base.

The derivatives associated to $i$th rule after reordering can be obtained from the reordered derivative vectors:

$$\Psi_{X_j} = Q\Psi_{X_j} = \begin{bmatrix} \mu_{\tilde{F}_{j,1}}' \\ \vdots \\ \mu_{\tilde{F}_{j,M}}' \end{bmatrix}$$ (5.50)

The indicator vectors contain information about which of the derivatives in the $M$ rules are the same and thus the partial derivative can be further revised as:

$$\frac{\partial q_{h}}{\partial U_{F_{j,i}}} = \begin{cases} \left( \frac{1}{N_f^2} \right) \left( \Psi_{X_j}(h) \cdot \left( \prod_{N=1, N\neq j}^{n} \Psi_{X_j}(h) \right) \cdot N_l - \bar{f}(h) \sum_{s} (\Psi_{X_j}'(s)) \cdot \prod_{N=1, N\neq j}^{n} \Psi_{X_k}(s) \right), & I_{U_{F_{j,i}}}(h) = I_{U_{F_{j,i}}}'(i); s \in [1, L], I_{U_{F_{j,i}}}'(s) = I_{U_{F_{j,i}}}'(i) \\
\left( \frac{1}{N_f^2} \right) \left( -\bar{f}(h) \sum_{s} (\Psi_{X_j}'(s) \cdot \prod_{N=1, N\neq j}^{n} \Psi_{X_k}(s)) \right), & I_{U_{F_{j,i}}}(h) \neq I_{U_{F_{j,i}}}'(i); s \in [1, L], I_{U_{F_{j,i}}}(s) = I_{U_{F_{j,i}}}'(i) \end{cases}$$

$$\frac{\partial q_{h}}{\partial U_{F_{j,i}}} = \begin{cases} \left( \frac{1}{N_f^2} \right) \left( \Psi_{X_j}(h) \cdot \left( \prod_{N=1, N\neq j}^{n} \Psi_{X_j}(h) \right) \cdot N_r - \bar{f}(h) \sum_{k} (\Psi_{X_j}'(k)) \cdot \prod_{N=1, N\neq j}^{n} \Psi_{X_k}(k) \right), & I_{U_{F_{j,i}}}(h) = I_{U_{F_{j,i}}}'(i); k \in [R + 1, M], I_{U_{F_{j,i}}}'(k) = I_{U_{F_{j,i}}}'(i) \\
\left( \frac{1}{N_f^2} \right) \left( -\bar{f}(h) \sum_{k} (\Psi_{X_j}'(k) \cdot \prod_{N=1, N\neq j}^{n} \Psi_{X_k}(k)) \right), & I_{U_{F_{j,i}}}(h) \neq I_{U_{F_{j,i}}}'(i); k \in [R + 1, M], I_{U_{F_{j,i}}}(k) = I_{U_{F_{j,i}}}'(i) \end{cases}$$

$$\frac{\partial q_{h}}{\partial U_{F_{j,i}}} = \begin{cases} \left( \frac{1}{N_f^2} \right) \left( \Psi_{X_j}(h) \cdot \left( \prod_{N=1, N\neq j}^{n} \Psi_{X_j}(h) \right) \cdot N_f - \bar{f}(h) \sum_{k} (\Psi_{X_j}'(k)) \cdot \prod_{N=1, N\neq j}^{n} \Psi_{X_k}(k) \right), & k \in [R + 1, M], I_{U_{F_{j,i}}}(k) = I_{U_{F_{j,i}}}'(i) \end{cases}$$
The partial derivative can be simplified as:

\[
\left( \frac{\partial u_l}{\partial F_{j,i}} + \frac{\partial u_r}{\partial F_{j,i}} \right) = \sum_{a=1}^{L} w_a \frac{\partial q_a^l}{\partial F_{j,i}} + \sum_{b=L+1}^{M} w_b \frac{\partial q_b^{l_j}}{\partial F_{j,i}} + \sum_{c=1}^{R} w_c \frac{\partial q_c^r}{\partial F_{j,i}} + \sum_{d=R+1}^{M} w_d \frac{\partial q_d}{\partial F_{j,i}}
\]

\[
= \frac{1}{N_l} \sum_s \left( \psi_{x_j}'(s) \cdot \prod_{N=1, \ N \neq j}^n \psi_{x_N}(s) \right) - \left( \sum_{a=1}^{L} w_a \frac{\partial \bar{f}(a)}{N_l^2} + \sum_{b=L+1}^{M} w_b \frac{\partial \bar{f}(b)}{N_l^2} \right).
\]

Using the Equations (5.20) to (5.23) for \( u_l \) and \( u_r \) as:

\[
u_l = \sum_{i=1}^{L} w_i \bar{f}_{l_i} + \sum_{i=L+1}^{M} w_i \bar{f}_{l_i}
\]

\[
u_r = \sum_{i=1}^{R} w_i \bar{f}_{l_i} + \sum_{i=R+1}^{M} w_i \bar{f}_{l_i}
\]

the partial derivative can be simplified as:

\[
\left( \frac{\partial u_l}{\partial F_{j,i}} + \frac{\partial u_r}{\partial F_{j,i}} \right) = \sum_{a=1}^{L} w_a \frac{\partial q_a^l}{\partial F_{j,i}} + \sum_{b=L+1}^{M} w_b \frac{\partial q_b^{l_j}}{\partial F_{j,i}} + \sum_{c=1}^{R} w_c \frac{\partial q_c^r}{\partial F_{j,i}} + \sum_{d=R+1}^{M} w_d \frac{\partial q_d}{\partial F_{j,i}}
\]

\[
= \frac{1}{N_l} \sum_s \left( \psi_{x_j}'(s) \cdot \prod_{N=1, \ N \neq j}^n \psi_{x_N}(s) \right) - \frac{u_r}{N_r} \sum_k \left( \psi_{x_j}^r(k) \cdot \prod_{N=1, \ N \neq j}^n \psi_{x_N}(k) \right).
\]

The final form of gradient descent rule for updating \( \bar{F}_{j,i} \) is:

\[
U_{\bar{F}_{j,i}}(t) = U_{\bar{F}_{j,i}}(t - 1) + \delta \cdot e(t) \cdot \left( \frac{\partial u_l}{\partial F_{j,i}} + \frac{\partial u_r}{\partial F_{j,i}} \right), \quad \delta = \frac{\eta}{2\gamma}
\]
where $U^F_{j,i}$ is either $U^F_{L_{j,i}}$ or $U^F_{R_{j,i}}$, which is determined by the value of input $X_j$.

The update rules for lower MF variables $U$ are shown in Appendix B as the derivation is very similar. The only difference is that $U^F_{j,i}$ is associated only with the lower firing level $f$.

After all the MF variables in the reordered rule base are updated, permutation matrix may be used to transform the MF variable vectors back to the form in original rule base.

$$U^o_{X_j} = Q^{-1}U^o_{X_j} = Q^{-1}egin{bmatrix} U^F_{j,1} \\ \vdots \\ U^F_{j,M} \end{bmatrix}, \quad U^o_{X_j} = Q^{-1}U^o_{X_j} = Q^{-1}egin{bmatrix} U^F_{j,1} \\ \vdots \\ U^F_{j,M} \end{bmatrix}$$ (5.54)

Finally, the antecedent indictor vectors can be used to pick up the updated MF variables related to the antecedent fuzzy sets $\tilde{X}_{j,p}$, $p = 1, \cdots, N_j$:

$$U^{\tilde{X}_{j,p}} = U^F_{j,1}, \quad U^{\tilde{X}_{j,p}} = U^F_{j,1}, \quad \text{when } p = I^X_{j,o}(i), \quad i = 1, \cdots, M$$ (5.55)

The flow chart of the full update algorithm is provided in Figure 5.4. Once the inputs $X_1, \cdots X_n$ are fed into the system, which side of membership function for $\tilde{X}_{j,p}$ ($i = 1, \cdots, n$ and $p = 1, \cdots, N_j$) is activated can be determined by the value of $X_j$. Then the values of the membership grades and its derivatives with respect to the related MF variable can be calculated. The indicator vectors $I^X_{j,o}$ for the original rule-base then can be used to initialize the parameter vectors, $\Gamma^o$, $\Gamma^o$, $\Psi^o_{\tilde{X}_j}$, $\Psi^o_{\tilde{X}_j}$, $\bar{\Psi}^o_{\tilde{X}_j}$ and $\psi^o_{\tilde{X}_j}$. At the same time, the MF variables that need to be updated can be written into vector form as $U^o_{X_j}$ and $U^o_{X_j}$.

### 5.5 Case Study: pH Neutralization Process

Section 5.4 proposes an online self-learning algorithm for training the weights and MF variables for a general type-2 FNC. In this section, the performance of the learn-
Chapter 5. On-line Learning Algorithm for Type-2 Fuzzy-Neural Controller

Inputs $X_1, \ldots, X_n$

Original Rule Base with weights $z$

Reordered rule base, obtain $w$ and $Q$

Calculate the partial derivatives, $w(t-1)$

Weight update rule to calculate new weights

Karnik-Mendel method for switch points $L$ & $R$.
Compute parameters $N_l, N_r, q_l$ & $q_r$.

Reordered $f^o$, $f^r$, $\Psi_{\tilde{X}_r}, \Psi_{\tilde{X}_L}$, $\Psi_{\tilde{X}_r}, \Psi_{\tilde{X}_L}$,

$L^o_i$, determine variable vectors $\tilde{U}^{o}_{\tilde{X}_r}, \tilde{U}^{o}_{\tilde{X}_L}$ for update

Calculate the partial derivatives,

Gradient descent rule to update MF variables $\tilde{U}^{F_l}(t)$ & $\tilde{U}^{F_r}(t)$.

$Q^{-1}$

Reverse permutation for new weights and MF variables of original rule-base

Figure 5.4. Flow of full update algorithm
ing algorithm for the type-2 FNC will now be investigated using a pH neutralization process. The pH neutralization process is a benchmark nonlinear control problem and the titration curve is a severely nonlinear relationship. Thus the modelling ability of the type-2 FNC will be examined to show the advantage of FOU in the type-2 FNC to model the nonlinear relationship over the type-1 FNC with more fuzzy sets and rules.

Figure 5.5 shows a diagram of the pH neutralization process[18]. The process consists of a main tank (Tank 1) which receives a carbonic acid stream ($H_2CO_3$) from a secondary tank (Tank 2) at volumetric flow rate $q_{1e}$ ml/s, a buffer stream of $NaOH$ at rate $q_2$ ml/s and a base stream ($NaOH$) at rate $q_3$ ml/s. The three flows are mixed in Tank 1 with the acid stream coming via acid flow $q_1$ ml/s into Tank 2, which introduces additional dynamics into the plant. In this investigation, $q_1$ and $q_2$ are regarded as unmeasured disturbances whose values can be set manually. Tanks 1 and 2 have liquid levels $h_1$ cm and $h_2$ cm respectively. The reading $h_1$ passes through a level transmitter (LT). The output flow rate from Tank 1 is $q_4$ and the pH
value of the stream is measured using a pH probe. For the purposes of subsequent
simulation it is assumed that the level and height transmitters have unity transfer
functions. The reference input of the type-2 FNC is the value of pH at the output
flow and the control output of the type-2 FNC is only the flow rate $q_3$.

The main chemical reactions occurring in the tank are the following dissociation
equilibrium equations involving carbonic acid:

$$
H_2CO_3 \leftrightarrow HCO_3^- + H^+
$$
$$
HCO_3^- \leftrightarrow CO_3^{2-} + H^+
$$
$$
H_2O \leftrightarrow OH^- + H^+
$$

The following equilibrium constants are defined based on the concentration of
the ionic species present in the chemical equations in (5.56)

$$
K_{a1} = \frac{[HCO_3^-][H^+]}{[H_2CO_3]}
$$
$$
K_{a2} = \frac{[CO_3^{2-}][H^+]}{[HCO_3^-]}
$$
$$
K_{a3} = [OH^-][H^+]
$$

Further, the following reaction invariants are defined for the outlet stream $q_4$ in
Tank 1:

$$
W_{a4} = [H^+] - [OH^-] - [HCO_3^-] - 2[CO_3^{2-}]
$$
$$
W_{b4} = [H_2CO_3] + [HCO_3^-] + [CO_3^{2-}]
$$

The chemical reactions of pH neutralization take place in a CSTR. Given the
equilibrium constants and reaction invariants from (5.57) and (5.58), the dynamics
of CSTR for pH neutralization process can be developed as follows:
The equation modelling Tank 2’s dynamics is given by:

\[ A_2 \frac{dh_2}{dt} = q_1 - q_{1e} \] (5.59)

where \( A_2 \) is the cross sectional area of Tank 2 in \( cm^2 \) and \( h_2 \) is the tank liquid level in \( cm \). For Tank 1, the differential equation governing the input-output relationship is:

\[ A_1 \frac{dh_1}{dt} = q_{1e} + q_2 + q_3 - q_4 \] (5.60)

where \( A_1 \) is the cross sectional area of Tank 1 in \( cm^2 \) and \( h_1 \) is the liquid level in \( cm \). The acid flow \( q_{1e} \) into Tank 1 and flow \( q_4 \) out of Tank 1 are modelled by the following equations:

\[ q_{1e} = C_1 \sqrt{h_2} \] (5.61)
\[ q_4 = C_2(h_1 + z)^n \] (5.62)

where \( C_1 \) and \( C_2 \) are proportionality constants and \( z \) represents the vertical distance between the bottom of Tank 1 and the valve output for \( q_4 \). Equations (5.61) and (5.62) are solved for tank levels \( h_1 \) and \( h_2 \). These values are then passed to the following pair of differential equations that update the reaction invariants \( W_{a4} \) and \( W_{b4} \):

\[ A_1 h_1 \frac{dW_{a4}}{dt} = q_{1e}(W_{a1} - W_{a4}) + q_2(W_{a2} - W_{a4}) + q_3(W_{a3} - W_{a4}) \] (5.63)
\[ A_2 h_1 \frac{dW_{b4}}{dt} = q_{1e}(W_{b1} - W_{b4}) + q_2(W_{b2} - W_{b4}) + q_3(W_{b3} - W_{b4}) \] (5.64)

where \( W_{ai} \) and \( W_{bi} \) \( (i = 1 \cdots 3) \) are assumed to be constant reaction invariants. The updated reaction invariants \( W_{a4} \) and \( W_{b4} \) are used in solving the following fourth
order equation for the hydrogen ion concentration $[H^+]$:

$$
[H^+]^4 + (K_{a1} - W_{a4})[H^+]^3 + (K_{a1}K_{a2} - W_{a4}K_{a1} - W_{b1}K_{a1} - K_w)[H^+]^2 - \\
(K_wK_{a1} + 2W_{b1}K_{a1}K_{a2} + W_{a4}K_{a1}K_{a2})[H^+] - (K_wK_{a1}K_{a2}) = 0
$$

(5.65)

Finally the pH value is calculated using:

$$
pH = -log_{10}[H^+]
$$

(5.66)

![Titration Curve](image)

Figure 5.6. Titration curve of the pH neutralization process

The nominal conditions of the pH neutralization process are shown in the Table 5.1[19]. With these initial conditions, the titration curve of the pH neutralization process within the typical range of pH 4 to pH 10 is shown in Figure 5.6. The titration curve is a severe nonlinear relationship between the pH value and the base stream flow rate $q_3$ $(ml/s)$. The nonlinear titration curve may be roughly divided into four segments, namely (pH 4 → pH 5.5), (pH 5.5 → pH 7), (pH 7 → pH 9) and (pH 9 → pH 10). The titration curve actually shows the nonlinearity of control
gain, which changes according to the condition of pH. The range of pH 7 to pH 9 has
the largest control gain as a slight change in $q_3$ will cause a big change in pH value.

There are two major turnings on the titration curve at the pH of 7 and 9 which
make the modelling a difficult task. In order to model the titration relationship
well, the reference trajectory will covers these four segments with the local setpoints
at 4.5, 5.5, 6, 7, 8, 9 and 10 as shown in the Figure 5.7. The step signal is passed
through a reference model \( \frac{1}{300s+1} \) since the general rise time of the responses in
[18] is around 300 seconds and thus the reference would be within the capability of
the actuators.

### 5.5.1 Performance of type-2 FNC with online weights and
MF variables update

The performance of the type-2 FNC with 9 rules is shown in this subsection. For
the type-2 FNC, there are two inputs fed into the FNC, one is the reference signal
\( X_1 = r = y_{ref} \) and the other is the derivative of reference signal \( X_2 = \dot{r} = \dot{y}_{ref} \) as
in Figure 5.3. There are three fuzzy sets, \( \tilde{X}_{1,1} \), \( \tilde{X}_{1,2} \) and \( \tilde{X}_{1,3} \) for describing input \( r \);
there are three fuzzy sets, \( \tilde{X}_{2,1} \), \( \tilde{X}_{2,2} \) and \( \tilde{X}_{2,3} \) for describing input \( \dot{r} \). Figure 5.8 shows
the interval type-2 fuzzy sets of the antecedents. The universes of the discourses of
both inputs \( X_1 = r \) and \( X_2 = \dot{r} \) are all normalized into [0,1]. The original input

---

Table 5.1. Parameter and initial conditions of the pH plant

<table>
<thead>
<tr>
<th>$h_1$</th>
<th>14.0 cm</th>
<th>$W_{a_1}$</th>
<th>$3 \times 10^{-3} M$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_2$</td>
<td>3.0 cm</td>
<td>$W_{a_2}$</td>
<td>$-0.03 M$</td>
</tr>
<tr>
<td>$A_1$</td>
<td>207 cm$^2$</td>
<td>$W_{a_3}$</td>
<td>$-3.05 \times 10^{-3} M$</td>
</tr>
<tr>
<td>$A_2$</td>
<td>42 cm$^2$</td>
<td>$W_{a_4}$</td>
<td>$-4.32 \times 10^{-4} M$</td>
</tr>
<tr>
<td>$z$</td>
<td>0 cm</td>
<td>$W_{b_1}$</td>
<td>0 M</td>
</tr>
<tr>
<td>$n$</td>
<td>0.607</td>
<td>$W_{b_2}$</td>
<td>0.03 M</td>
</tr>
<tr>
<td>$q_1$</td>
<td>16.6 ml/s</td>
<td>$W_{b_3}$</td>
<td>$5 \times 10^{-3} M$</td>
</tr>
<tr>
<td>$q_{1e}$</td>
<td>16.6 ml/s</td>
<td>$W_{b_4}$</td>
<td>$5.28 \times 10^{-4} M$</td>
</tr>
<tr>
<td>$q_2$</td>
<td>0.55 ml/s</td>
<td>$C_1$</td>
<td>9.584</td>
</tr>
<tr>
<td>$q_4$</td>
<td>32.8 ml/s</td>
<td>$C_2$</td>
<td>4.593</td>
</tr>
<tr>
<td>$K_{a_1}$</td>
<td>$4.47 \times 10^{-7}$</td>
<td>$K_{a_1}$</td>
<td>$5.62 \times 10^{-11}$</td>
</tr>
<tr>
<td>$K_w$</td>
<td>$1.00 \times 10^{-14}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
domains for $r$ and $\dot{r}$ are very different, the reason of normalizing the domains is to reduce the number of design parameters and it is better to have a same learning rate for MF variables of both inputs. This will require the derivatives of the membership grades with respect to the MF variables to be within the similar magnitude for both the antecedents of $r$ and $\dot{r}$ and thus normalization is needed. This may help the updating of MF variables for $r$ and $\dot{r}$ to keep the similar pace when using the MF variable update rules which involves the derivatives of membership grades. Beside that, the values of the MF variables are assumed to be positive, since the type-2 fuzzy sets are assumed to be just shifted from the type-1 principal MFs by the MF variables. Since there are only three fuzzy sets to describe the nonlinear relationship for $r$, the first one is for pH at 4 and the last one is for pH at 10, the peak of the middle fuzzy set is located at pH=8 as the most nonlinear period for modelling is from pH at 7 to 9. After scaling, the value 4 is normalized as 0, and the value 10 is normalized as 1, thus the value of 8 is normalized as 0.67. The fuzzy sets for
input $\dot{r}$ are uniformly distributed along the normalized domain. The possible range of signal $\dot{r}$ is $[-0.01, 0.01]$ for the reference model $\frac{1}{300s + 1}$, thus the value $-0.01$ for $\dot{r}$ is normalized as 0 and the value $0.01$ for $\dot{r}$ is normalized as 1.

Figure 5.8. Type-2 antecedent fuzzy sets of type-2 FNC with 9 rules

Since there are four fuzzy sets which are half-side shape, thus there are eight MF variables for antecedents of input $r$, namely $U_{L}^{X_{1,1}}, U_{R}^{X_{1,1}}, U_{L}^{X_{1,2}}, U_{R}^{X_{1,2}}, U_{L}^{X_{1,3}}$, and $U_{R}^{X_{1,3}}$; similarly there are eight MF variables for antecedents of input $\dot{r}$, namely $U_{R}^{X_{2,1}}, U_{R}^{X_{2,1}}, U_{L}^{X_{2,2}}, U_{L}^{X_{2,2}}, U_{R}^{X_{2,2}}, U_{R}^{X_{2,2}}, U_{L}^{X_{2,3}}$, and $U_{L}^{X_{2,3}}$. Including the nine weights, it means there are 25 free parameters in the type-2 FNC to be updated.

The rule base of the type-2 FNC with 9 rules is constructed using the relationship defined in Equation (5.34). The information in the titration curve is used to initialize the weights. From Figure 5.6, three key points are read as $(q_{3} = 10 \rightarrow \text{pH}=4)$, $(q_{3} = 16.5 \rightarrow \text{pH}=8)$ and $(q_{3} = 20 \rightarrow \text{pH}=10)$. Hence, all the weights of the rules which share the fuzzy set $X_{1,1}$ as antecedent for $r$ are set as 10; and the weights of the rules which share the fuzzy set $X_{1,2}$ as 16.5; and the weights of the rules which

Figure 5.8. Type-2 antecedent fuzzy sets of type-2 FNC with 9 rules

Since there are four fuzzy sets which are half-side shape, thus there are eight MF variables for antecedents of input $r$, namely $U_{L}^{X_{1,1}}, U_{R}^{X_{1,1}}, U_{L}^{X_{1,2}}, U_{R}^{X_{1,2}}, U_{L}^{X_{1,3}}$, and $U_{R}^{X_{1,3}}$; similarly there are eight MF variables for antecedents of input $\dot{r}$, namely $U_{R}^{X_{2,1}}, U_{R}^{X_{2,1}}, U_{L}^{X_{2,2}}, U_{L}^{X_{2,2}}, U_{R}^{X_{2,2}}, U_{R}^{X_{2,2}}, U_{L}^{X_{2,3}}$, and $U_{L}^{X_{2,3}}$. Including the nine weights, it means there are 25 free parameters in the type-2 FNC to be updated.

The rule base of the type-2 FNC with 9 rules is constructed using the relationship defined in Equation (5.34). The information in the titration curve is used to initialize the weights. From Figure 5.6, three key points are read as $(q_{3} = 10 \rightarrow \text{pH}=4)$, $(q_{3} = 16.5 \rightarrow \text{pH}=8)$ and $(q_{3} = 20 \rightarrow \text{pH}=10)$. Hence, all the weights of the rules which share the fuzzy set $X_{1,1}$ as antecedent for $r$ are set as 10; and the weights of the rules which share the fuzzy set $X_{1,2}$ as 16.5; and the weights of the rules which
Table 5.2. Simulation parameters of the type-2 FNC

<table>
<thead>
<tr>
<th>Simulation Environment Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference Model</td>
<td>$\frac{1}{300s + 1}$</td>
</tr>
<tr>
<td>Plant</td>
<td>pH plant</td>
</tr>
<tr>
<td>Simulation iterations</td>
<td>50</td>
</tr>
<tr>
<td>Sampling period</td>
<td>5 secs</td>
</tr>
<tr>
<td>Weights learning rate</td>
<td>0.2</td>
</tr>
<tr>
<td>MF variables learning rate</td>
<td>0.005</td>
</tr>
<tr>
<td>Proportional Gain $K_p$</td>
<td>2.4</td>
</tr>
</tbody>
</table>

share the fuzzy set $\tilde{X}_{1,3}$ as 20. The weights are initialized as:

$$w = [10, 10, 10, 16.5, 16.5, 16.5, 16.5, 20, 20, 20]^T \quad (5.67)$$

The simulation environment of the type-2 FNC is given in Table 5.4. The proportional gain is the same as the proportional gain $K_c = 2.4ml/s$ used in the seminal work on pH control[19]. The sampling time is 5 seconds. The weights learning rate is set as 0.2. Since the input domains are all normalized into [0,1], the possible range of MF variables is smaller. Hence, smaller learning rate is required for updating the MF variables comparing with the weight learning rate, which is set as 0.005. Other parameters of the system are shown in Table 5.1. For the MF variables, they are all initialized as zeros so that the initial type-2 FNC is a type-1 FNC. The full update algorithm, including both the weights and MF variables update rules (Equation (5.32), (5.53) and (B-12) ), is used to self-generate a type-2 FNC for modelling the nonlinear titration relationship.

Figure 5.9 shows the response of type-2 FNC with 9 rules in the 1st iteration. Generally speaking, the steady state response is very good. However, there are some transient response errors, especially during the step change between pH at 9 and 10 where the period of titration curve has a big turning and thus the tracking task is more difficult. The response of the type-2 FNC with 9 rules at the 50th iteration is shown in Figure 5.10. The self-learning type-2 FNC is able to regulate the pH value well. The ISE of the response along one iteration of the reference trajectory
Figure 5.9. Response of type-2 FNC with 9 rules in 1st iterations

Figure 5.10. Response of type-2 FNC with 9 rules after 50 iterations
Figure 5.11. ISEs of the performance of type-2 FNC with 9 rules

Figure 5.12. Weights of type-2 FNC with 9 rules during the 50 learning iterations
Chapter 5. On-line Learning Algorithm for Type-2 Fuzzy-Neural Controller

Figure 5.13. Weights of type-2 FNC with 9 rules at the 50th iteration

Figure 5.14. MF variables for lower MFs of input \( r \) of type-2 FNC with 9 rules at the 50th iteration
Figure 5.15. MF variables for upper MFs of input $r$ of type-2 FNC with 9 rules at the 50th iteration

Figure 5.16. MF variables for input $\dot{r}$ of type-2 FNC with 9 rules at the 50th iteration
is chosen as the performance index. Figure 5.11 shows the ISEs for each of the 50 learning iterations. The ISE drops very quickly and reach nearly 1.9 at the 5th iteration. Finally, the ISE settles at an ISE value of 0.9729. It may indicate that the MF variables of the FNC has provided a flexible way to model the nonlinear relationship well with less fuzzy sets and weights.

Figure 5.12 shows the nine weights of the type-2 FNC during the 50 iterations. Generally speaking, most weights converge very fast. After 18 iterations, the average value of each updating weight nearly converges. Figure 5.13 shows that a majority of the weights nearly does not vary during the 50th iteration. The exception are $w_2$ and $w_5$ which have only slight updates. It also indicates the MF variables have performed very well to model the nonlinear relationship and the weights may be regarded as to fine tune the control surface.

Figure 5.14 and 5.15 show the trajectory of MF variables of input $r$ for lower and upper membership functions while Figure 5.16 shows the trajectory of MF variables of input $\dot{r}$. It can be seen that the MF variables are updated only when the related fuzzy sets are activated. It means when the MF variables change, the resulting control surface is reconstructed accordingly to minimize the error to the desired one. In order to examine the modelling ability of the type-2 FNC with updatable MF variables, the MF variables is chosen to be the maximum values during the 50th iteration, since a larger FOU may generally generate larger nonlinear relationship. From Figure 5.14, 5.15 and 5.16, the MF variables are chosen as:

\begin{align}
U_{X_1,1}^{R} &= 0.1799, \quad U_{L}^{X_1,2} = 0.2314, \quad U_{R}^{X_1,2} = 0.0000, \quad U_{L}^{X_1,3} = 0.3396 \\
U_{R}^{X_1,1} &= 0.3161, \quad U_{L}^{X_1,2} = 0.1867, \quad U_{R}^{X_1,2} = 0.1628, \quad U_{L}^{X_1,3} = 0.0433
\end{align}

\begin{align}
\hat{U}_{X_2,1}^{R} &= 0.1406, \quad \hat{U}_{L}^{X_2,2} = 0.0136, \quad \hat{U}_{R}^{X_2,2} = 0.0044, \quad \hat{U}_{L}^{X_2,3} = 0.0391 \\
\hat{U}_{R}^{X_2,1} &= 0.1171, \quad \hat{U}_{L}^{X_2,2} = 0.0090, \quad \hat{U}_{R}^{X_2,2} = 0.0101, \quad \hat{U}_{L}^{X_2,3} = 0.1382
\end{align}
Also, from Figure 5.13, the weights are chosen from the last sample at the 50th iteration:

\[ \mathbf{w} = [7.4369, 9.4838, 12.4046, 16.8551, 17.6861, 16.9036, 19.4135, 18.8924, 23.7932]^T \]

Figure 5.17. Optimized antecedents of type-2 FNC with 9 rules after 50 learning iterations

With these optimized MF variables, the type-2 antecedents of the optimized type-2 FNC are shown in Figure 5.17. Using the optimized weights and MF variables, the control surface of the optimized type-2 FNC is shown in Figure 5.18. The control slice when \( \dot{r} = 0 \), which means the steady state period, may be used to show the modelling performance on the titration curve. The control slice and the titration curve are both shown in Figure 5.19. It shows that the type-2 FNC models the nonlinear titration relationship very well, especially at the two big turning periods around pH of 7 and 9. With the assistant of MF variables, the type-2 FNC may model the nonlinear relationship very well so that the need of using weights to vary the control surface is now not so urgent and the weights are just used for fine tuning.
Figure 5.18. Control surface of type-2 FNC with 9 rules and optimized weights and MF variables

Figure 5.19. Control slice when $\dot{q} = 0$ of type-2 FNC with 9 rules and optimized weights and MF variables
The MF variables provide the type-2 FNC additional freedom and a flexible way to model the nonlinear relationship online using less fuzzy sets and weights. The more degrees of freedom in the system, better performance is easier to achieve. Hence, it is interesting to compare the performance of type-2 FNC with 9 rules and that of a type-1 FNC with more fuzzy sets and rules. The next subsection will give a detail discussion on such comparison.

### 5.5.2 Performance of type-1 FNC

The motivation of the study is to compare the performance on controlling the pH with the trajectory at different values of the type-1 FNC and the performance of type-2 FNC whose extra freedom can help to model the nonlinear relationship with less fuzzy sets and rules. There are two type-1 FNCs, one is with 9 rules and the other is with 25 rules, in this subsection for comparison. When the FOU of type-2 FNC reduces to zero, the type-2 FNC is actually reduces to a type-1 FNC with 9 rules. The performance of type-1 FNC with 9 rules will be shown first. However, in order to achieve a fair comparison, a type-1 FNC with 25 rules which has same degrees of freedom as the type-2 FNC will be examined later.

For the type-1 FNC, the basic structure is very similar to those of type-2 FNC but it uses type-1 fuzzy sets. Figure 5.20 shows the type-1 fuzzy sets of the antecedents for the type-1 FNC with 9 rules. The universes of the discourses of both inputs are consistent with the case in type-2 FNC. The rule base of the type-1 FNC with 9 rules is constructed using the relationship in Equation (5.34). The weights are also initialized to be the same as in Equation (5.67):

\[
    \mathbf{w} = [10, 10, 10, 16.5, 16.5, 16.5, 20, 20, 20]^T 
\]

Figure 5.21 compares the initial control slice with the titration curve. The control slice is actually a segmented line passing through the three key point at pH of 4, 8 and 10 with the respective desired \( q_3 \) values.
Figure 5.20. Antecedent fuzzy sets of type-1 FNC with 9 rules

Figure 5.21. Control slice with three key points at initialization and titration curve
Table 5.3. Simulation parameters of the type-1 FNC

<table>
<thead>
<tr>
<th>Simulation Environment Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference Model</td>
<td>( \frac{1}{300s + 1} ) pH plant</td>
</tr>
<tr>
<td>Plant</td>
<td>pH plant</td>
</tr>
<tr>
<td>Simulation iterations</td>
<td>50</td>
</tr>
<tr>
<td>Sampling period</td>
<td>5 secs</td>
</tr>
<tr>
<td>Controller learning rate (( \delta ))</td>
<td>0.2</td>
</tr>
<tr>
<td>Proportional Gain ( K_p )</td>
<td>2.4</td>
</tr>
</tbody>
</table>

The simulation environment is given in Table 5.3. The learning rate for updating the weights is maintained at 0.2. Other parameters of the system are the same as in previous section. The weights of the type-1 FNC are updated using the weights update rule for 50 iterations. The ISEs of the type-1 FNC with 9 rules during the 50 learning iterations are shown in Figure 5.22. The ISE drops at the beginning and nearly converges after 10 iterations and it reaches 3.192 at the 50th iteration which is much bigger that the ISE of type-2 FNC.

![Performance of T1FNC with 9 rules](image)

Figure 5.22. ISEs of the performance of type-1 FNC with 9 rules

The weights of the 9 rules are always updating because the desired control gain also varies as the reference pH value changes. Figure 5.23 shows the nine weights...
Chapter 5. On-line Learning Algorithm for Type-2 Fuzzy-Neural Controller

Figure 5.23. Weights of type-1 FNC with 9 rules during the 50 learning iterations

Figure 5.24. Weights of type-1 FNC with 9 rules at the 50th iteration
trajectories over 50 iterations. The variation of the weights is significantly larger than for the type-2 FNC in Figure 5.12. This may be due to the lack of fuzzy sets to model the nonlinear input-output relationship. Figure 5.24 shows the weight trajectories during the 50th iteration. For each reference setpoint change, four weights are updated at each sampling to change the current control surface to the desired control surface decided by the titration curve. The variation of the weights are much bigger than that of type-2 FNC, since the current type-1 FNC only has nine degrees of freedom.

Obviously, three key points are not enough to model the titration curve. Hence, a type-1 FNC with 25 rules, which provides five key points and has the same degrees of freedom as the type-2 FNC, is applied to model the titration curve. The basic structure is very similar to the one with 9 rules. Figure 5.25 shows the fuzzy sets of the antecedents.

![Type-1 FNC with 25 rules, antecedent fuzzy sets](image)

![Type-1 FNC with 25 rules, antecedent fuzzy sets](image)

Figure 5.25. Antecedent fuzzy sets of type-1 FNC with 25 rules

The five key points are selected based on the titration curve which are the two end points and the three major turning points at 5.5, 7.5 and 9. Hence, they are
(q_3 = 10 \rightarrow \text{pH}=4), (q_3 = 11.6 \rightarrow \text{pH}=5.5), (q_3 = 16.2 \rightarrow \text{pH}=7.5), (q_3 = 17 \rightarrow \text{pH}=9)\) and (q_3 = 20 \rightarrow \text{pH}=10). The weights are initialized as:

\[
\mathbf{w} = [10, 10, 10, 10, 11.6, 11.6, 11.6, 11.6, 11.6, 11.6, 16.2, 16.2, 16.2, 16.2, 16.2, 16.2, 17, 17, 17, 17, 20, 20, 20, 20]^T
\] (5.73)

Figure 5.26. Control slice with five key points at initialization and titration curve.

Figure 5.26 compares the initial control slice of the type-1 FNC with 25 rules and the titration curve. The modelling performance at initialization of the type-1 FNC with 25 rules is much better than that of the type-1 FNC with only 9 rules because the FNC has more degrees of freedom now. Figure 5.27 shows the ISEs of the type-1 FNC with 25 rules along the 50 learning iterations. As already shown in Figure 5.26, the initial weight vector better models the titration curve. Hence, the ISE of first iteration is 2.79, which is much better than the 6.2 for type-1 FNC with 9 rules. With the larger freedom provided by the 25 weights, the ISE drops more quickly to the minimum 1.221 at the 50th iteration. The performance of the type-1 FNC with 25 rules is much better than the type-1 FNC with 9 rules. The additional fuzzy
Chapter 5. On-line Learning Algorithm for Type-2 Fuzzy-Neural Controller

Figure 5.27. ISEs of the performance of type-1 FNC with 25 rules

Figure 5.28. Weights of type-1 FNC with 25 rules during the 50 learning iterations
Figure 5.29. Weights of type-1 FNC with 25 rules at the 50th iteration

sets and weights also contribute to reduce the variation in the weights as shown in
Figure 5.28. Comparing with Figure 5.24 and Figure 5.29, the variation of the 25
weights in the 50th iteration has already greatly reduced.

Even with the same 25 degrees of freedom as those in type-2 FNC with 9 rules,
the ISE of type-1 FNC with 25 rules is not as good as the ISE of type-2 FNC with 9
rules. In addition, the variation of some weights of type-1 FNC with 25 rules is still
a bit larger than that of type-2 FNC with 9 rules, e.g., $w_{13}$ and $w_{23}$ comparing Figure
5.29 with Figure 5.13. Generally speaking, the performance of the type-2 FNC is
the best one among the three kinds of FNCs. The type-2 FNC utilizes 25 degrees
of freedom and full online update algorithm to achieve the best performance.

5.6 Case Study: Thermal chamber

In this section, experimental results that assesses the performance of the type-2
FNC as a temperature controller in a thermal chamber with uncertain disturbance
are presented. The test bed is again the thermal chamber shown in Figure 3.16.
The uncertain disturbance of the thermal control system comes from the fan which
has uncertain rotation speed to blow wind into the thermal chamber. The uncertain rotation speed of fan is driven by a random duty cycle within the range of [0.2, 0.7] as shown in Figure 5.30. The objective of this study is to examine the performance of type-2 FNC on controlling the thermal chamber with uncertain disturbance and compare with the performances of conventional PI controller and type-1 FNC with similar degrees of freedom. The reference trajectory $y_{ref}$ is the step response of system $\frac{1}{5s+1}$ between $28^\circ C$ to $30^\circ C$. The reference model is chosen based on the heating capability of the lamp, thereby assuring that the response may be able to track the reference trajectory. Each step change is 100 seconds.

![Figure 5.30. Disturbance from the fan with uncertain rotation speed](image)

### 5.6.1 Performance of type-2 FNC

The performance of the type-2 FNC with 4 rules is shown in this subsection. The type-2 FNC have two inputs fed into the FNC. One is the reference signal $X_1 = r = y_{ref}$ and the other is the derivative of reference signal $X_2 = \dot{r} = y_{ref}'$ as in Figure 5.3. There are two fuzzy sets, $\tilde{X}_{1,1}$ and $\tilde{X}_{1,2}$ for describing input $r$ and two fuzzy
sets, \( \tilde{X}_{2,1} \) and \( \tilde{X}_{2,2} \) for describing input \( \dot{r} \).

![Type-2 antecedent fuzzy sets of type-2 FNC with 4 rules](image)

**Figure 5.31.** Type-2 antecedent fuzzy sets of type-2 FNC with 4 rules

The range of \( r \) is set as [28,31] based on the reference signal. With a sampling time of 0.1 second, the possible range of the signal \( \dot{r} \) is [-0.01, 0.01] for the reference model \( \frac{1}{5s + 1} \). As the un-scaled domain for \( r \) and \( \dot{r} \) are very different, the derivatives of the membership grades with respect to the MF variables would not have similar magnitude. Consequently, the pace at which the MF variables for the two inputs are updated may differ. In order to employ a common learning rate, the input domains, \( r \) and \( \dot{r} \), are normalized. After scaling \( r \), the value 28 is normalized as 0, and the value 31 is normalized as 1. Similarly, the value of 0.01 for \( \dot{r} \) is normalized as 0 and the value of 0.01 is normalized as 1. Beside that, the values of the MF variables are constrained to be positive, since the type-2 fuzzy sets are assumed to be just shifted from the type-1 principal MFs by the MF variables.

All the four antecedent fuzzy sets are half-side shape, thus there are four MF variables for antecedents of input \( r \), namely \( \underline{U}^{\tilde{X}_{1,1}}_L, \underline{U}^{\tilde{X}_{1,1}}_R, \underline{U}^{\tilde{X}_{1,2}}_L \) and \( \overline{U}^{\tilde{X}_{1,2}}_L \); similarly there are four MF variables for antecedents of input \( \dot{r} \), namely \( \underline{U}^{\tilde{X}_{2,1}}_R, \underline{U}^{\tilde{X}_{2,1}}_R, \underline{U}^{\tilde{X}_{2,2}}_L \) and \( \overline{U}^{\tilde{X}_{2,2}}_L \).
Figure 5.32. Response of type-2 FNC with 4 rules in the last iteration

Figure 5.33. ISEs of the performance of type-2 FNC with 4 rules
Table 5.4. Experiment parameters of the type-2 FNC

<table>
<thead>
<tr>
<th>Experimental Environment Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference Model</td>
<td>$\frac{1}{5s+1}$</td>
</tr>
<tr>
<td>Plant</td>
<td>thermal chamber</td>
</tr>
<tr>
<td>Sampling period</td>
<td>0.1 sec</td>
</tr>
<tr>
<td>Learning rate ($\delta$)</td>
<td>0.1</td>
</tr>
<tr>
<td>Proportional Gain $K_p$</td>
<td>1</td>
</tr>
</tbody>
</table>

and $\bar{U}_L^{\bar{X}_{2,2}}$. Including the four weights, it means there are 12 free parameters in the type-2 FNC to be updated.

The experiment parameters of the type-2 FNC are given in Table 5.4. The proportional gain is set as 1. The learning rate is set as 0.1. The weights are all initialized as zero. For the MF variables, they are all initialized as zeros so that the initial type-2 FNC is a type-1 FNC. The full update algorithm, including both the weights and MF variables update rules (Equation (5.32), (5.53) and (B-12) ), is used for the type-2 FNC.

![Learning curves of weights of type-2 FNC with 4 rules](image)

**Figure 5.34.** Weights of type-2 FNC with 4 rules during the learning iterations

Figure 5.32 shows the response during the 7th iteration of the type-2 FNC on controlling the thermal chamber under uncertain disturbance. Although the speed
Figure 5.35. Weights of type-2 FNC with 4 rules at the last iteration (dashed lines as estimated average values)

of the fan is uncertain as shown in Figure 5.30, the type-2 FNC does a good job of tracking the reference trajectory. The ISE of the response along one step change of the reference trajectory is chosen as the performance index. The ISEs of the response during each step change are shown in Figure 5.33. The ISE reduces very quickly to values around 1.5. Figure 5.34 shows the trajectories of the four weights during the learning iterations. Generally speaking, most weights settle very fast. After 500 seconds, the average value of each weight does not change much. Figure 5.35 shows that the four weights does not vary much during the last iteration even though the fan speed varies (the dashed lines are the estimated average values of weights).

Figure 5.36 shows the trajectory of MF variables of input $r$ while Figure 5.39 shows the trajectory of MF variables of input $\dot{r}$. It can be seen that the MF variables are always updated during the learning iterations since the four type-2 sets are always fired by the input. Figure 5.37 shows the deviation of MF variables from the mean for input $r$ during the learning iterations. Figure 5.40 shows the deviation of MF variables for input $\dot{r}$. Both figures show that the changes of most MF variables
Figure 5.36. MF variables for input $r$ of type-2 FNC with 4 rules

Figure 5.37. Deviation of MF variables learning trajectories from the mean for input $r$
Figure 5.38. MF variables for input $r$ of type-2 FNC with 4 rules at the 50th iteration

Figure 5.39. MF variables for input $\dot{r}$ of type-2 FNC with 4 rules
Figure 5.40. Deviation of MF variables learning trajectories from the mean for input $\dot{r}$

Figure 5.41. MF variables for input $\dot{r}$ of type-2 FNC with 4 rules at the 50th iteration
during updating quickly reduce after several iterations which means it can gradually handle the effect of uncertain disturbance. Figure 5.38 and Figure 5.41 show the trajectories of MF variables in the last iteration in which the oscillation has reduced a lot.

When the MF variables change, the resulting control surface is reconstructed accordingly to minimize the error to the desired reference and thus minimize the effect of uncertainty from the fan disturbance. With the assistance of the MF variables, the type-2 FNC may handle the uncertain disturbance very well so that the weights are not very oscillatory and just used for fine tuning. The MF variables provide the type-2 FNC additional freedom and a flexible way to handle the uncertainty online using less fuzzy sets and weights. The more degrees of freedom in the system, better performance is easier to achieve. Hence, it is interesting to compare the performance of type-2 FNC with 9 rules and that of a type-1 FNC with more fuzzy sets and rules. Next subsection will give a detail discussion on such comparison.

5.6.2 Performance of conventional PI controller and type-1 FNC with 12 rules

The motivation of the study is to compare the performances of controlling the thermal chamber with a conventional PI controller and type-1 FNC with the performance of type-2 FNC whose extra freedom can help to handle the uncertainty with less fuzzy sets and rules. The performance of conventional PI controller will be shown first. The proportional gain is chosen as 1 and the integral gain as 0.45. Other parameters are the same as in Table 5.4.

Figure 5.42 shows the response of conventional PI controller which is worse than that of type-2 FNC. Figure 5.43 shows that the ISEs of its response stay around the value of 3 which is much bigger than that of type-2 FNC. The PI controller cannot handle well the sudden change in the uncertain speed of fan and thus there are relatively big jumps away from the reference as in Figure 5.42.
Figure 5.42. Response of conventional PI controller in the last iteration

Figure 5.43. ISEs of conventional PI controller
Chapter 5. On-line Learning Algorithm for Type-2 Fuzzy-Neural Controller

Figure 5.44. Type-1 antecedent fuzzy sets of type-1 FNC with 12 rules

Figure 5.45. Response of type-1 FNC with 12 rules in the last iteration
For the type-1 FNC, the basic structure is very similar to those of type-2 FNC but it uses type-1 fuzzy sets. In order to compare under similar basis, the type-1 FNC has 12 rules which has the same degrees of freedom as the type-2 FNC. Figure 5.44 shows the type-1 fuzzy sets of the antecedents for the type-1 FNC with 12 rules. Four antecedent type-1 sets are used to describe the domain of input $r$; while three type-1 sets describe the domain of input $\dot{r}$. The weights are also initialized at zero and its learning rate is still 0.2. There is no MF variables update for type-1 FNC since its MF variables are zero. Other parameters are the same as in Table 5.4.

Figure 5.45 shows the response of type-1 FNC and it is better than that of conventional PI controller. The twelve weights in type-1 FNC are updated continuously to counter the effect of uncertain disturbance on the ability of the controller to track the reference. The ISEs of its response during learning iterations are shown in Figure 5.46.

Figure 5.47 shows the weight trajectories during the learning iterations. Some of the weights does not vary much during learning since there are more rules in the
Figure 5.47. Weights of type-1 FNC with 12 rules during the learning iterations

Figure 5.48. Weights of type-1 FNC with 12 rules at the last iteration
type-1 FNC so the weights associated with rules that have zero firing level will not be updated. The updating of weights in the last iteration is shown in Figure 5.48.

Figure 5.49. Comparison of performances for the three controllers

By comparing the performances of three controllers as in Figure 5.49, the best one is the type-2 FNC with 4 rules and its performance is quite stable as the ISEs always close to 1.5. The type-1 FNC with 12 rules performs in between the type-2 FNC and conventional PI controller. It has similar performance to type-2 FNC at 6th, 7th, 10th and 11th iteration. But it performs worse than type-2 FNC at other times since the ISEs jumps between 1.5 and 2.6. That means the performance of type-1 FNC is not stable as type-2 FNC.

5.7 Conclusion

The chapter presents the general structure of type-2 fuzzy neural system and proposes the control scheme for on-line updating of the FNC parameters. The on-line learning algorithm for both the weights and MF variables is derived. The most important and challenging part of the learning algorithm is computing the derivatives
that are needed for MF variables. This is mainly because the updated weights are not always in the same order and thus the rule-base has to be reordered for the type-reduction. The newly introduced antecedent indicator vectors help to track the reordering processes and provides information to calculate the derivatives for MF variables in shorter and simpler formula.

The proposed type-2 FNC with full update algorithm is tested using the pH neutralization process between carbonic acid and sodium hydroxide, a benchmark control problem. The reference signal covers general range from pH 4 to pH 10 and its titration curve is severely nonlinear with major three big turnings. The results show that the performance of the type-2 FNC with full update algorithm is the best, even comparing with the type-1 FNC with 25 rules which has the same degrees of freedom. By using type-2 fuzzy sets, the FOU or its MF variables have provide a flexible way to reconstruct the control surface with better modelling performance on the titration curve. Hence, the task to reshape the control surface by the weights of type-2 FNC is shared by MF variables and the weights may be regarded as a fine tuning tools. With the help of MF variables, less weights and rules are needed for the type-2 FNC to achieve similar performance comparing with its type-1 counterpart and the weights also converge faster and more smoothly.

Finally, experiment has been done using the type-2 FNC to control the thermal chamber system with uncertain disturbance. The reference signal range from 28°C to 31°C. The results show that the performance of the type-2 FNC with full update algorithm is the best, even comparing with the type-1 FNC with 12 rules which has the same degrees of freedom. By using type-2 fuzzy sets, the FOU or its MF variables also prove to provide a flexible way to reconstruct the control surface with better performance on handling the uncertainty. With the extra degree of freedoms from MF variables, less weights and rules are needed for the type-2 FNC to achieve similar performance.
Chapter 6

Conclusions and Future Work

6.1 Conclusions

Uncertainty is the fabric that makes life interesting. For millennia, human beings have developed strategies to cope with a plethora of uncertainties. The major aim of this thesis is to investigate the feasibility of applying the extensional fuzzy theory, namely non-singleton FLSs and type-2 FLSs, to control different kinds of uncertain system and examine its efficiency on minimizing the effect of uncertainty. In the real world, noise is one of the most common kinds of uncertain factors. Most control systems are corrupted by different kinds of noise and minimizing effect of noise will bring about higher control performance. One common way to handle noise using fuzzy theory is to utilize non-singleton fuzzification in FLS. However, a study on the properties of traditional symmetric non-singleton fuzzifier shows that general fuzzy PI controller using symmetric triangular non-singleton fuzzifier with small spread based on noise level has similar output surface to a singleton fuzzy PI controller. Hence, Chapter 3 introduced a non-symmetric non-singleton fuzzifier to achieve better noise rejection on the noisy signals. The non-symmetric fuzzifier is designed based on the effect of traditional symmetric with large spread on firing the antecedent sets. The new non-symmetric fuzzifier also has balancing function on related antecedent sets and thus cancelling the noisy information introduced by
noise. Another advantage is that the parameters of the non-symmetric fuzzifier can be designed based on the noise level. Results for controlling a highly nonlinear pH neutralization process showed that such non-singleton fuzzy PI controller provided a smoother response at the steady state, which is very important for pH neutralization. In addition, the transient response is very good with less overshoot comparing to responses with moving average filter. Hence, the proposed non-symmetric non-singleton fuzzifier is a better alternative solution for the fuzzy PI controller to handle uncertainties in the inputs.

Non-singleton fuzzification is suitable for handling uncertain inputs but may not be good for other kinds of uncertainty. The extra dimension in the type-2 sets provide a type-2 FLS with more flexibility to handle other kinds of linguistic and numerical uncertainties. Chapter 4 aims to propose the framework of type-2 PI controller to provide flexible output surface based on uncertain information to maintain desired performance. By extending the procedure to construct type-1 PI controller, a procedure for construction of type-2 PI controller is proposed using pre-determined gains based on uncertain system parameters and type-2 operations, e.g. centroids and type-reduction. The benefit of such type-2 PI controller is that its output surface is bounded with gains in pre-determined ranges and thus making it easier to ensure stability. Reviews show that type-2 operation using symmetric type-2 sets may lead to same results as treating type-2 sets as type-1 sets. In order to handle uncertain system, the type-2 FLS should have flexible output surface. Hence, an adaptive algorithm to update switch points and thus vary relevant centroids based on the two new theorems is derived in Chapter 4. The motivation of the adaptive algorithm is to vary the centroids of consequent sets in the type-2 PI controller framework and thus provide a variable control surface based on uncertain information to minimize effect of uncertainties and maintain desired performance. Monte Carlo simulation results show that the performance is quite robust and stable on maintaining desired performance to control the uncertain FOPDT. On the other hand, the adaptive algorithm varying the centroids may result in varying the
equivalent control gains, which is similar to conventional fuzzy gain-scheduling PI controller. Using the nonlinear pH neutralization process as test bed, the simulation results show that the performance of type-2 PI controller with adjustable centroids is better than conventional fuzzy gain-scheduling on tracking reference trajectory for pH plants with uncertain system parameters.

The pH neutralization process is a severe nonlinear process that the system gain may vary greatly for different setpoints. This requires the controller to vary its controller gain according to the desired setpoint, which may be considered as an uncertain factor of the pH plant as well. To obtain a good performance on controlling pH neutralization process, the modelling on the titration curve is an important task. The aim of Chapter 5 is to examine the modelling ability of type-2 FLS with the help of extra degrees of freedom of fuzzy sets, FOU or MF variables. To facilitate the design of type-2 FLS, the type-2 fuzzy neural system scheme is adopted to construct an on-line learning control scheme, which is referred as type-2 FNC. The type-2 FNC uses type-2 antecedent sets and singleton weights with general rule-base where antecedent set in each rule is generally not unique and may be used in another rule. Chapter 5 derives in detail the rules for updating the weights and the MF variables of antecedent sets in the general rule-base. The pH neutralization process with setpoints ranges from 4 to 10 is used to test the performance of the type-2 FNC. The results show that the performance of the type-2 FNC with full update algorithm is the best, even comparing with the type-1 FNC with 25 rules which has the same degrees of freedom. The extra dimension in type-2 fuzzy sets provides the controller with more flexibility to reconstruct the control surface, thereby producing better modelling performance on the titration curve. Hence, the task to reshape the control surface by the weights of type-2 FNC is shared by MF variables and the weights may be regarded as a fine tuning tools. Hence, the type-2 FNC needs less weights and rules to outperform its type-1 counterpart and at the same time the weights and MF variables converge faster and more smoothly.

In conclusion, the research has led to the development of fuzzy controllers that
utilize non-singleton fuzzy logic and type-2 fuzzy logic for system with different kinds of uncertainty or nonlinearity. These control schemes have shown the advantages offered by non-singleton fuzzifier, type-2 fuzzy sets and related operations. The FLSs with extensional fuzzy theories have the potentials to outperform the type-1 counterpart in many practical fields when uncertainty is present. In order to further explore other properties and potential of the FLSs with extensional fuzzy theories, possible directions for future research are suggested in the next section.

6.2 Suggestions for Future Work

The design of parameters in non-symmetric non-singleton fuzzifier is based on the noise level of the measurement system. For the case of nonlinear pH neutralization process, the effect of noisy measurement may change according to the system gain at different setpoints. The design of parameter $B_v$ may indirectly differentiate the transience and steady state period. Hence, an adaptive scheme may be needed to evaluate the effect of noise on the response and thus provide enough information to adjust the parameters to maintain the performance when the setpoint is changing within a large range and uncertain.

Since non-singleton fuzzifier can handle noise and type-2 PI controller can handle system with uncertain parameters, it is worth examining to design a non-singleton type-2 PI controller with adjustable control surface to control uncertain systems corrupted with noise. Another issue can be further researched is to examine the effect of type-2 antecedents within the framework of type-2 PI controller if the inputs are corrupted by noise. It is interesting to compare the performances of type-2 PI controllers with non-singleton fuzzification and type-2 antecedents. The comparison of type-2 antecedents and type-1 non-singleton fuzzification on fuzzy control is still a sterile area for research.

At present, the consequent weights of type-2 FNC are chosen as singletons. To fully utilize the extra freedom provided by the type-2 FLS, the weights may be
chosen as interval type-1 sets in the future research. With extra dimension in the consequents, the control surface of the type-2 FNC may have extra degrees of freedom on modelling nonlinear relationship. The type-2 sets in the thesis are all in triangular shape. The results of using other kinds of membership function, e.g. Gaussian MFs, may be investigated. In addition, when the parameters of pH plant are uncertain, the titration curve may change greatly and this will burden the task of modelling. In the future, the type-2 FNC with more type-2 antecedent sets and consequent weights could be designed to control such a benchmark problem.
Appendix A

Relationship between FOU and control surface

This Appendix will mainly examine how the FOUs of the type-2 fuzzy sets of a type-2 FNC may help to reconstruct the control surface. For simplicity, the case where there are only two fuzzy sets for each input of the type-2 FNC is discussed in this Appendix. Hence, there are four rules in the rule-base. Similar to Chapter 5, the initialization strategy is to estimate static relationship with MF variables/FOU using prior knowledge and hopefully the FOU may learn dynamics well and fine tune static behavior. First of all, the domain of each input will be divided into one section with only two antecedents for each input. The fuzzy sets are all triangular membership functions. There are normally three kinds of type-2 triangular membership functions with three kinds of FOU structures which are shown in Figure A-1, A-2 and A-3. Figure A-1 shows the type-2 triangular membership functions with uncertain base, where each type-2 fuzzy set has only one point with maximum grade. The domains of two inputs are normalized into \([0 1]\). The MF variables are defined as in Figure A-1. Once the MF variables are determined for these membership functions, the FOU shape for handling the uncertainty will be determined. Figure A-2 shows the type-2 triangular membership functions whose upper membership functions and lower membership function are parallel. The upper or lower membership functions are derived by shifting the principal membership functions according to the offset defined by the MF variables. Figure A-3 shows the type-2 triangular membership functions with uncertain peak, i.e. each type-2 fuzzy set has only one point with
Figure A-1. MF variables for type-2 triangles with uncertain base

Figure A-2. MF variables for parallel type-2 triangles
The rule base of the two-inputs-one-output fuzzy system is:

\[ R_1 : \text{If } X_1 \text{ is } \tilde{X}_{11} \text{ and } X_2 \text{ is } \tilde{X}_{21} \text{ then } Y \text{ is } w_1 \]
\[ R_2 : \text{If } X_1 \text{ is } \tilde{X}_{11} \text{ and } X_2 \text{ is } \tilde{X}_{22} \text{ then } Y \text{ is } w_2 \]
\[ R_3 : \text{If } X_1 \text{ is } \tilde{X}_{12} \text{ and } X_2 \text{ is } \tilde{X}_{21} \text{ then } Y \text{ is } w_3 \]
\[ R_4 : \text{If } X_1 \text{ is } \tilde{X}_{12} \text{ and } X_2 \text{ is } \tilde{X}_{22} \text{ then } Y \text{ is } w_4 \]

For the type-2 FNC discussed in Chapter 5, the input \( X_1 \) is actually the reference signal \( r \) and the input \( X_2 \) is the derivative of reference signal \( \dot{r} \). The weights are initialized based on the antecedents for input \( X_1 \). For example, the desired control surface for controlling the pH neutralization process is based on the titration curve which is related to the reference pH. Hence, it is reasonable to initialize the weights based on only the antecedents of \( X_1 \). The values of weights, for example, are
assumed to be initialized as \( W = [w(1) \ w(2) \ w(3) \ w(4)]^T = [0 \ 0 \ 1 \ 1]^T \) (The weights are selected just for demonstration). The objective is to vary the MF variables of antecedents independently while the weights are fixed to study their effect on varying the control surface. All the three cases of type-2 triangular membership functions will be covered in the following discussion.

![Control Surface](image)

Figure A-4. Control surface of the type-1 FLS

For a type-1 FLS when the FOU of current type-2 FLS reduces to zero, the control surface is a plane and its range is \([0 \ 1]\) as shown in Figure A-4. Since the weights are initialized based on input \(X_1\), the control slices of the surface for different \(X_2\) will be the same. Thus, the control slice will be used to study the FOUs’ effect on varying the control surface. The control slice for the type-1 FLS is just a straight line joining between 0 and 1.

## A.1 Control surface using type-2 triangles with uncertain base

Since the study assumes that the desired control surface is mainly determined by the input \(X_1\), the FOUs of the antecedents of input \(X_1\) will use type-2 triangles
and the antecedents of input $X_2$ are described by type-1 sets. Assuming that the MF variables of fuzzy sets $\tilde{X}_{11}$ and $\tilde{X}_{12}$ are chosen as below just for demonstration:

$$
\text{Upper bound } \tilde{X}_{11} = 0.25, \quad \text{Lower bound } \tilde{X}_{11} = 0.4, \quad \text{Upper bound } \tilde{X}_{12} = 0.15, \quad \text{Lower bound } \tilde{X}_{12} = 0.25 \quad (A-1)
$$

Figure A-5 shows the control surface of the type-2 FLS. Since the weights are initialized based on $X_1$, the control slice may be used for study as shown in Figure A-6. The upper part of the Figure shows the FOUs for the antecedents of $X_1$. The lower part of the Figure shows the upper and lower bounds of the output set ($Y_r$ and $Y_l$), and thus the final output $Y$.

The upper bound $Y_r$ and lower bound $Y_l$ are both curves with obvious nonlinear features. Hence, the output slice $Y$ is not a straight line but has two obvious turning points corresponding to the two turning points of the lower membership functions of $\tilde{X}_{11}$ and $\tilde{X}_{12}$ sets, namely $X_1 = 0.25$ and $X_1 = 0.6$. The two turning points have also divided the output slice $Y$ into three non-linear segments. On the other hand, the range of the output is still $[0, 1]$ and the starting and end points correspond to
the turning points of the upper membership functions, $X_1 = -0.15$ and $X_1 = 1.25$. The turning points of the control slice will result in decreasing the derivatives of the two sections at ends. That means, the lower MF variables, $\overline{U}_{\bar{X}_{11}}$ and $\overline{U}_{\bar{X}_{12}}$, are main factors to vary the control surface when using type-2 triangles with uncertain base.

To further support the hypothesis that the turning points of the lower membership functions may change the control slice, another combination of MF variables is chosen as

$$\overline{U}_{\bar{X}_{11}} = 0.4, \; \overline{U}_{\bar{X}_{11}} = 0.2, \; \overline{U}_{\bar{X}_{12}} = 0.1, \; \overline{U}_{\bar{X}_{12}} = 0.4 \quad (A-2)$$

Figure A-7 and A-8 show that the general shapes of control surface and slice are very similar to those with previous combination of MF variables. The results indicate that the turning points of the lower membership functions correspond to the turning points at the control surface or slice regardless of the values of the MF
Figure A-7. Control surface of type-2 FLS using type-2 triangles with uncertain base for X1 only–A new case of MF variables’ combination

A.2 Control surface using parallel type-2 triangles

The type-2 FLS with parallel type-2 triangles for the antecedents of input X1 and type-1 sets for the antecedents of input X2 are investigated here. The MF variables of fuzzy sets $\tilde{X}_{11}$ and $\tilde{X}_{12}$ are the same as the previous study in Equation (A-1) to provide a common basis for comparison.

Figure A-9 shows the control surface of the type-2 FLS with type-2 sets only for the antecedents of X1. The control surface for parallel type-2 triangular FLS is very similar to the control surface for the type-1 FLS in Section A.1. The effect of the MF variables is similar to previous case as well. Although there are also two turning points on the control slice as in Figure A-10, the control slice here is flatter than previous one in Figure A-6. This may be due to that the lower and upper membership functions are parallel and thus the upper and lower bounds of output set, $Y_l$ and $Y_r$, will be less nonlinear. Hence, it results in a smaller degree of turning when unparallel membership functions are employed. Also, the three
Figure A-8. Control slice of type-2 FLS using type-2 triangles with uncertain base for X1 only–A new case of MF variables’ combination
Figure A-9. Control surface of type-2 FLS using parallel type-2 triangles for X1 only

Figure A-10. Control slice of type-2 FLS using parallel type-2 triangles for X1 only
segments divided by the turning points is nearly linear. That means the effect of FOUs using parallel type-2 triangles is not so strong as that using type-2 triangles with uncertain base.

A.3 Control surface using type-2 triangles with uncertain peak

From the study in Section A.1, it is known that the discontinuity of the slope of the lower membership function due to the turning point at the base causes the final control slice correspondingly to generate a turn point. Now, the type-2 triangles with uncertain peak will be examined. As the type-2 set with uncertain base is the opposite of the type-2 set with uncertain peak, it is hypothesized that the MF variables of upper membership function may have the turning effect on the control slice. The MF variables of fuzzy sets $\tilde{X}_{11}$ and $\tilde{X}_{12}$ defined in Figure A-3 are also chosen as:

$\bar{U}_{\tilde{X}_{11}} = 0.25$, $\underline{U}_{\tilde{X}_{11}} = 0.4$, $\bar{U}_{\tilde{X}_{12}} = 0.15$, $\underline{U}_{\tilde{X}_{12}} = 0.25$

Figure A-11 shows that the upper membership functions only have limited turning effect on the final control slice which is quite straight. This may be because the values of the upper MF variables are not big enough to show the turning effect.

In order to test the conjecture that larger MF values will yield a control slice with obvious turning points, the FOUs of the antecedents are enlarged, and the upper MF variables are now increased as below:

$\bar{U}_{\tilde{X}_{11}} = 0.8$, $\underline{U}_{\tilde{X}_{11}} = 0.4$, $\bar{U}_{\tilde{X}_{12}} = 0.55$, $\underline{U}_{\tilde{X}_{12}} = 0.15$

Figure A-12 shows the control surface when the upper MF variables for antecedents of $X_{11}$ are large. The control slice and the fuzzy sets of antecedents are shown in Figure A-13. Similarly, the turning points of the upper membership func-
Figure A-11. Control slice of type-2 FLS using type-2 triangles with uncertain peak for X1 only

Figure A-12. Control surface of type-2 FLS using type-2 triangles with uncertain peak for X1 only—Large upper MF variables
tions also generate the corresponding turning points of the control slice. However, the turning points of the control slice now result in larger slopes at the two end sections.

Unfortunately, the upper MF variables are now so large that the uncertain peak sections with largest membership of both the upper membership functions for $\tilde{X}_{11}$ and $\tilde{X}_{12}$ overlap each other. However, the possible range of the upper MF variables should be $\bar{U} \leq 1$, otherwise, the grade of upper membership function is not unique due to the shape of triangle and thus FOU will be meaningless. Hence, there is not much space left for the MF variables to vary the control slice when only large MF variables can generate obvious tuning effect. This would restrict the FOU on reshaping the control surface to model the complex relationship.

After studying the effect of FOUs in a type-2 FNC on the control surface when using the three kinds of type-2 triangular sets, the basic properties of the MF variables for the three kinds of membership functions is drawn. The FOUs of the parallel type-2 triangular sets may generate similar control surface comparing with the one provided by the type-2 triangles with uncertain base, but the nonlinearity is not so obvious. To achieve the same degree of turning effect, larger FOU values may be needed. The least effective case is the one with type-2 triangles with uncertain peak, in such a case, normally large MF variables are needed in order to vary the control surface. In conclusion, the most effective one for varying the control surface is the MF variable for the type-2 triangles with uncertain base. Consequently, the recommendation is to construct the type-2 FNC using the type-2 triangles with uncertain base.
Figure A-13. Control slice of type-2 FLS using type-2 triangles with uncertain peak for X1 only—Large upper MF variables
Appendix B

Update rules for lower MF variables

Similar to the procedures for upper MF variables, the partial derivatives of \( \bar{q}_1 \), \( q_l \), \( q_r \) and \( q_r \) with respect to \( U^{F_j,i} \) can be derived as:

\[
\frac{\partial \bar{q}_h}{\partial U^{F_j,i}} = \left( \frac{1}{N^2} \right) \left( \frac{\partial \bar{f}_h}{\partial U^{F_j,i}} N_l - \bar{f}_h \frac{\partial N_l}{\partial U^{F_j,i}} \right), \quad h \in [1, \cdots, L] \quad (B-1)
\]

\[
\frac{\partial q_l}{\partial U^{F_j,i}} = \left( \frac{1}{N^2} \right) \left( \frac{\partial f_l}{\partial U^{F_j,i}} N_l - f_l \frac{\partial N_l}{\partial U^{F_j,i}} \right), \quad h \in [L+1, \cdots, M] \quad (B-2)
\]

\[
\frac{\partial q_r}{\partial U^{F_j,i}} = \left( \frac{1}{N^2} \right) \left( \frac{\partial f_r}{\partial U^{F_j,i}} N_r - f_r \frac{\partial N_r}{\partial U^{F_j,i}} \right), \quad h \in [R+1, \cdots, M] \quad (B-3)
\]

\[
\frac{\partial q_r}{\partial U^{F_j,i}} = \left( \frac{1}{N^2} \right) \left( \frac{\partial f_r}{\partial U^{F_j,i}} N_r - f_r \frac{\partial N_r}{\partial U^{F_j,i}} \right), \quad h \in [1, \cdots, R] \quad (B-4)
\]

But now \( U^{F_j,i} \) is associated only with the lower firing level \( f \), the partial derivatives can be simplified as:

\[
\frac{\partial \bar{q}_h}{\partial U^{F_j,i}} = \left( \frac{1}{N^2} \right) \left( -\bar{f}_h \sum_{k=L+1}^{M} \frac{\partial f_k}{\partial U^{F_j,i}} \right), \quad h \in [1, \cdots, L] \quad (B-5)
\]

\[
\frac{\partial q_l}{\partial U^{F_j,i}} = \left( \frac{1}{N^2} \right) \left( \frac{\partial f_l}{\partial U^{F_j,i}} N_l - f_l \sum_{k=L+1}^{M} \frac{\partial f_k}{\partial U^{F_j,i}} \right), \quad h \in [L+1, \cdots, M] \quad (B-6)
\]

\[
\frac{\partial q_r}{\partial U^{F_j,i}} = \left( \frac{1}{N^2} \right) \left( -f_r \sum_{k=1}^{R} \frac{\partial f_k}{\partial U^{F_j,i}} \right), \quad h \in [R+1, \cdots, M] \quad (B-7)
\]

\[
\frac{\partial q_r}{\partial U^{F_j,i}} = \left( \frac{1}{N^2} \right) \left( \frac{\partial f_r}{\partial U^{F_j,i}} N_r - f_r \sum_{k=1}^{R} \frac{\partial f_k}{\partial U^{F_j,i}} \right), \quad h \in [1, \cdots, R] \quad (B-8)
\]
Hence,

\[
\frac{\partial \eta^i_h}{\partial U^{F,j,i}} = \left( \frac{1}{N_i^2} \right) \left( -\bar{T}_h \sum_{k=L+1}^{M} \left( \frac{\partial \mu_{F,j,k}}{\partial U^{F,j,i}} \cdot \left( \prod_{N=1, N\neq j}^{n} \mu_{F_N,k} \right) \right) \right),
\]

\[
\frac{\partial q^i_h}{\partial U^{F,j,i}} = \left( \frac{1}{N_i^2} \right) \left( \frac{\partial \mu_{F,j,h}}{\partial U^{F,j,i}} \cdot \left( \prod_{N=1, N\neq j}^{n} \mu_{F_N,k} \right) \right) \cdot N_i - \bar{T}_h \sum_{k=L+1}^{M} \left( \frac{\partial \mu_{F,j,k}}{\partial U^{F,j,i}} \cdot \left( \prod_{N=1, N\neq j}^{n} \mu_{F_N,k} \right) \right),
\]

\[
\frac{\partial \bar{\eta}^i_h}{\partial U^{F,j,i}} = \left( \frac{1}{N_i^2} \right) \left( -\bar{T}_h \sum_{k=1}^{R} \left( \frac{\partial \mu_{F,j,k}}{\partial U^{F,j,i}} \cdot \left( \prod_{N=1, N\neq j}^{n} \mu_{F_N,k} \right) \right) \right),
\]

\[
\frac{\partial q^i_h}{\partial U^{F,j,i}} = \left( \frac{1}{N_i^2} \right) \left( \frac{\partial \mu_{F,j,h}}{\partial U^{F,j,i}} \cdot \left( \prod_{N=1, N\neq j}^{n} \mu_{F_N,k} \right) \right) \cdot N_r - \bar{T}_h \sum_{k=1}^{R} \left( \frac{\partial \mu_{F,j,k}}{\partial U^{F,j,i}} \cdot \left( \prod_{N=1, N\neq j}^{n} \mu_{F_N,k} \right) \right).
\]

The notations of derivative vectors are also defined as:

\[
\Psi^0_X = \begin{bmatrix}
\mu'_{F,j,1} \\
\vdots \\
\mu'_{F,j,M}
\end{bmatrix}, \quad \mu'_{F,j,0}(X_j) = \frac{\partial \mu_{F,j}}{\partial U^{F,j,i}}
\]

and

\[
\Psi_X = \begin{bmatrix}
\mu'_{F,1} \\
\vdots \\
\mu'_{F,M}
\end{bmatrix}, \quad \mu'_{F,0}(X_j) = \frac{\partial \mu_{F,j}}{\partial U^{F,j,i}}
\]

To revise them into concise forms,

\[
\frac{\partial \eta^i_h}{\partial U^{F,j,i}} = \left( \frac{1}{N_i^2} \right) \left( -\bar{T}(h) \sum_{s} \left( \Psi_{\bar{X}_j}(s) \cdot \prod_{N=1, N\neq j}^{n} \Psi_{\bar{X}_N}(s) \right) \right), \quad s \in [L + 1, M], I_{U}^{X_j}(s) = I_{U}^{X_j}(i)
\]

\[
\frac{\partial q^i_h}{\partial U^{F,j,i}} = \begin{cases}
\left( \frac{1}{N_i^2} \right) \left( \Psi_{\bar{X}_j}(h) \cdot \prod_{N=1, N\neq j}^{n} \Psi_{\bar{X}_N}(h) \right) \cdot N_i - \bar{T}(h) \sum_{s} \left( \Psi'_{\bar{X}_j}(s) \cdot \prod_{N=1, N\neq j}^{n} \Psi_{\bar{X}_N}(s) \right) \right), & \text{if } I_{U}^{X_j}(h) = I_{U}^{X_j}(i); s \in [L + 1, M], I_{U}^{X_j}(s) = I_{U}^{X_j}(i)
\end{cases}
\]

\[
\frac{\partial \bar{\eta}^i_h}{\partial U^{F,j,i}} = \begin{cases}
\left( \frac{1}{N_i^2} \right) \left( -\bar{T}(h) \sum_{s} \left( \Psi'_{\bar{X}_j}(s) \cdot \prod_{N=1, N\neq j}^{n} \Psi_{\bar{X}_N}(s) \right) \right), & \text{if } I_{U}^{X_j}(h) \neq I_{U}^{X_j}(i); s \in [L + 1, M], I_{U}^{X_j}(s) = I_{U}^{X_j}(i)
\end{cases}
\]
\[
\frac{\partial q_{k}^{r}}{\partial U^{F_{j},s}} = \left( \frac{1}{N_{r}^2} \right) \left( -\bar{f}(h) \sum_{k} \left( \Psi' \Xi_{j} (k) \cdot \prod_{N=1, N\neq j}^{n} \Psi \Xi_{N} (k) \right) \right),
\]

\[
k \in [1, R], I_{U}^{X_{j}} (k) = I_{U}^{X_{j}}(i)
\]

\[
\frac{\partial q_{r}^{s}}{\partial U^{F_{j},s}} = \begin{cases} \left( \frac{1}{N_{r}^2} \right) \left( \Psi' \Xi_{j} (h) \cdot \left( \prod_{N=1, N\neq j}^{n} \Psi \Xi_{N} (h) \right) \right) \cdot N_{r} - f(h) \sum_{k} \left( \Psi' \Xi_{j} (k) \cdot \prod_{N=1, N\neq j}^{n} \Psi \Xi_{N} (k) \right), & I_{U}^{X_{j}}(h) = I_{U}^{X_{j}}(i); k \in [1, R], I_{U}^{X_{j}}(k) = I_{U}^{X_{j}}(i) \\
\left( \frac{1}{N_{r}^2} \right) \left( -f(h) \sum_{k} \left( \Psi' \Xi_{j} (k) \cdot \prod_{N=1, N\neq j}^{n} \Psi \Xi_{N} (k) \right) \right), & I_{U}^{X_{j}}(h) \neq I_{U}^{X_{j}}(i); k \in [1, R], I_{U}^{X_{j}}(k) = I_{U}^{X_{j}}(i)
\end{cases}
\]

Finally,

\[
\left( \frac{\partial u_{l}}{\partial U^{F_{j},s}} + \frac{\partial u_{r}}{\partial U^{F_{j},s}} \right) = \sum_{a=1}^{L} w_{a} \frac{\partial q_{a}^{l}}{\partial U^{F_{j},s}} + \sum_{b=L+1}^{M} w_{b} \frac{\partial q_{b}^{l}}{\partial U^{F_{j},s}} + \sum_{c=1}^{R} w_{c} \frac{\partial q_{c}^{r}}{\partial U^{F_{j},s}} + \sum_{d=R+1}^{M} w_{d} \frac{\partial q_{d}^{r}}{\partial U^{F_{j},s}}
\]

\[
= \frac{1}{N_{r}} \sum_{s} \left( w_{s} \cdot \left( \prod_{N=1, N\neq j}^{n} \Psi \Xi_{N} (s) \right) \right) - \left( \sum_{a=1}^{L} w_{a} \bar{f}(a) \frac{1}{N_{r}^2} + \sum_{b=L+1}^{M} w_{b} \bar{f}(b) \frac{1}{N_{r}^2} \right). \tag{B-11}
\]

\[
\sum_{k} \left( \Psi' \Xi_{j} (k) \cdot \prod_{N=1, N\neq j}^{n} \Psi \Xi_{N} (k) \right) = \frac{1}{N_{r}} \sum_{k} \left( w_{k} \cdot \left( \prod_{N=1, N\neq j}^{n} \Psi \Xi_{N} (k) \right) \right)
\]

\[
= \frac{1}{N_{r}} \sum_{s} \left( w_{s} \cdot \left( \prod_{N=1, N\neq j}^{n} \Psi \Xi_{N} (s) \right) \right) - \frac{u_{l}}{N_{r}} \sum_{s} \left( \Psi' \Xi_{j} (s) \cdot \prod_{N=1, N\neq j}^{n} \Psi \Xi_{N} (s) \right)
\]

\[
\frac{u_{r}}{N_{r}} \cdot \sum_{k} \left( \Psi' \Xi_{j} (k) \cdot \prod_{N=1, N\neq j}^{n} \Psi \Xi_{N} (k) \right) = \frac{1}{N_{r}} \sum_{k} \left( w_{k} \cdot \left( \prod_{N=1, N\neq j}^{n} \Psi \Xi_{N} (k) \right) \right)
\]

\[
(s \in [L + 1, M], I_{U}^{X_{j}} (s) = I_{U}^{X_{j}}(i); k \in [1, R], I_{U}^{X_{j}}(k) = I_{U}^{X_{j}}(i))
\]

Also, the gradient descent rule for updating $U^{F_{j},s}$ is:

\[
U^{F_{j},s}(t) = U^{F_{j},s}(t - 1) + \delta \cdot e(t) \cdot \left( \frac{\partial u_{l}}{\partial U^{F_{j},s}} + \frac{\partial u_{r}}{\partial U^{F_{j},s}} \right), \quad \delta = \frac{\eta}{2\gamma} \tag{B-12}
\]

where $U^{F_{j},s}$ is either $U^{F_{j},s}_{L}$ or $U^{F_{j},s}_{R}$ which is determined by the value of input $X_{j}$. 
Author’s Publications

List of publications


Bibliography


[4] S. Auephanwiriyakul, A. Adrian, and J.M. Keller. Type 2 fuzzy set analysis
in management surveys. In *IEEE International Conference on Fuzzy Systems*,

using type-2 fuzzy sets. *Journal of Manufacturing Technology Management*,

using neural fuzzy controller. In *Second IEEE International Conference on

diagnostics using type-2 fuzzy logic and fractal theory. In *Proceedings of the
12th IEEE International Conference on Fuzzy Systems*, volume 1, pages 102

and neural networks. In *IEEE Conference on Cybernetics and Intelligent Sys-


[12] F. Doctor, H. Hagras, and V. Callaghan. A type-2 fuzzy embedded agent to re-
alise ambient intelligence in ubiquitous computing environments. *Information


tracking mobile objects in the context of robotic soccer games. In *Proceedings
2005 IEEE International Conference on Fuzzy Systems*, pages 359 – 364, Reno,
NV, May 2005.

[16] H. Hagras. A hierarchical type-2 fuzzy logic control architecture for au-


[61] J.M. Mendel. On a 50% savings in the computation of the centroid of a


