Fuzzy Interpolation Systems with Knowledge Extraction and Adaptation

JIE LI

PhD

2019

Fuzzy Interpolation Systems with Knowledge Extraction and Adaptation

JIE LI

A thesis submitted in partial fulfilment of the requirements of the

University of Northumbria at Newcastle

for the degree of

Doctor of Philosophy

Research undertaken in the

Faculty of Engineering and Environment

April 2019

Abstract

Fuzzy inference system provides an effective means for representing and processing vagueness and imprecision. Conventional fuzzy modelling requires either complete experts' knowledge or given datasets to generate rule bases such that the input spaces can be fully covered. Although fuzzy interpolation enhances the power of conventional fuzzy inference approaches by addressing the problem of lack of knowledge represented in the rule bases, it is still difficult for real-world applications to obtain sufficient experts' knowledge and/or data to generate a sparse rule base to support fuzzy interpolation. Also, the generated rule bases are usually fixed and therefore cannot support dynamic situations. In addition, all existing fuzzy interpolation approaches were developed based on the Mamdani fuzzy model, which are not applicable for the TSK fuzzy model. It significantly restricts the applicability of the TSK fuzzy inference systems.

This PhD work, in the first part, presents a novel fuzzy inference approach, termed "TSK+ fuzzy inference", to address the issue of performing the TSK inference over sparse rule bases. The proposed TSK+ fuzzy inference approach extends the conventional TSK fuzzy inference by considering the degree of similarity between given inputs and corresponding rule antecedents instead of conventional overlapped match degree, which allows TSK inference to be performed over sparse rule bases, dense rule bases, and imbalanced rule bases. In order to support the proposed TSK+ inference approach, a data-driven rule base generation method is also presented in this work. In addition, the proposed TSK+ inference approach has been further extended to deal with interval type-2 fuzzy sets. The effectiveness of this system in enhancing the TSK fuzzy inference is demonstrated through two real-world applications: a network intrusion detection system, and a network quality of service management system.

In addition, in the second part of this work, a new rule base generation and adaptation method is developed in order to relax the requirement of rule base generation, which allows the fuzzy rule base to be generated with minimal or even without a priori knowledge. The proposed method mimics the pedagogic approach of experiential learning, which achieves automatic rule base generation and adaptation by transferring the proceeding performance experiences when performing inferences. The proposed rule base generation and adaptation method has been evaluated by not only a mathematical model but also a well-known control problem, inverted pendulum. The experimental results show that the control system can generate an applicable rule base to support the system running, thus demonstrating the effectiveness of the proposed approach.

Table of contents

List of figures

List of tables

List of acronyms

Roman Symbols

k Number of clusters in k-Means clustering algorithm

Rep Representative value of a fuzzy set

Greek Symbols

λ Relative placement factor

π The ratio of a circle's circumference to its diameter $\simeq 3.14$

Acronyms / Abbreviations

API Application Programming Interface

ATMS Assumption-based Truth Maintenance System

CD Cooling Down Factor

COG Centre of Gravity

DSCP Differentiated Service Code Point

EF Experience Factor

FIS Fuzzy Inference System

FL Fuzzy Logic

FOU Footprint of Uncertainty

FRI Fuzzy Rule Interpolation

GA Genetic Algorithm

GDE General Diagnostic Engine

GPS Global Positioning System

IDWF Inverse Distance Weighting Factor

IT2 Interval Type-2

KH Kóczy-Hirota Approach

LMF Lower Membership Function

NIDS Network Intrusion Detection System

QoS Quality of Service

RMSE Root mean Square Error

SSE Sum of Squares Error

T-FRI Transformation-based Fuzzy Rule Interpolation

TSK Takagi-Sugeno-Kang

UMF Upper Membership Function

Acknowledgements

Foremost, I would like to take this opportunity to express my deepest gratitude and thanks to my principal supervisor, Dr. Longzhi Yang, for his patience, encouragement, and guidance throughout this PhD research. I must also give thanks to my second supervisor Dr. Graham Sexton. His discussions and advice are also appreciated. I have enjoyed the opportunity to learn from their knowledge and experience.

I would also like to acknowledge my gratitude to the staff in the Department of Computer and Information Sciences at Northumbria University for providing such a positive and nurturing environment in which to conduct my research.

I am grateful to all of my friends, who have helped me during my PhD study.

Lastly, my sincere gratitude goes to my parents and my wife. I would not have completed this work without their love, support, and understanding over the last a few years. I, therefore, dedicate this thesis to them.

Declaration

I declare that the work contained in this thesis has not been submitted for any other award and that it is all my own work. I also confirm that this work fully acknowledges opinions, ideas and contributions from the work of others.

Any ethical clearance for the research presented in this thesis has been approved. Approval has been sought and granted by the University Ethics Committee in 2015.

I declare that the Word Count of this Thesis is 35,712 words

Name: Jie LI

Signature:

Date:

Chapter 1

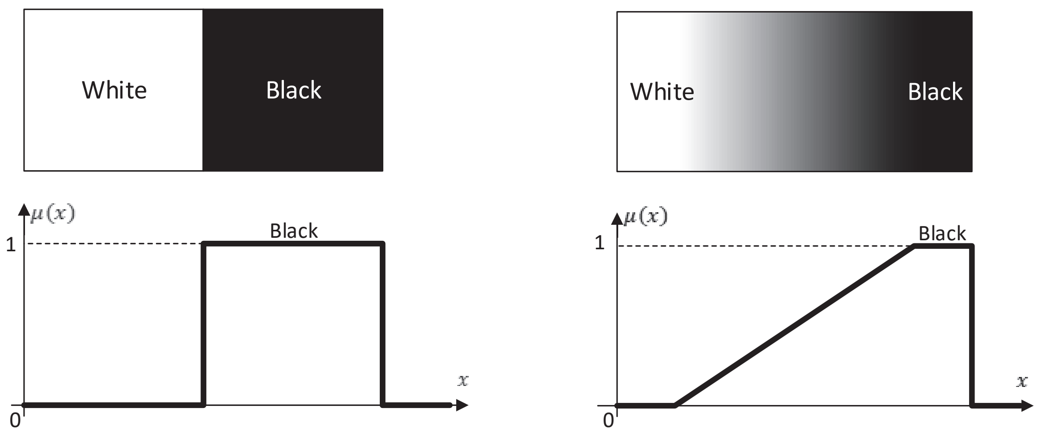
Introduction

Since Lotfi A. Zadeh separated hard computing, which often means the conventional computing based on boolean logic, binary systems, numerical analysis and crisp software, from soft computing, usually referring to probabilistic, possibilistic and nature-inspired reasoning approaches such as neural networks, fuzzy logic and genetic algorithms, in the early 1990s [221], soft computing has become a popular subfield of computer science. Compared with hard computing, which requires a precisely stated analytical model often associated with big computational costs, soft computing is tolerant to imprecision, uncertainty, partial truth and approximation [72]. Also, soft computing techniques resemble biological processes more closely than conventional computing methods, given the fact that biological processes, including common sense reasoning, are usually not based on binary logic or exact crisp numerical analysis.

Soft computing is a wide-ranging term encompassing such varied techniques as fuzzy logic [221], machine learning [163, 164], evolutionary computation [5, 47] and other probabilistic and possibilistic reasoning approaches [72, 141, 150]. All these methods aim to solve complex problems by exploiting the imprecision and uncertainty during decision-making processes; these approaches are complementary to each other and can be used together to solve a single given problem. In particular, the principle of fuzzy logic offers a practical way to represent and reason the ambiguity in human thinking with real-life uncertainty [107, 108]. Formally, fuzzy logic is a technology that uses fuzzy sets and logical connectives for the development of intelligent systems and for achieving machining intelligence by representing and reasoning upon human knowledge. The key contribution of fuzzy logic is the capturing of the vagueness of human thinking and expressing it with appropriate mathematical tools. By mimicking the human reasoning process using linguistic terms, fuzzy logic enables the comprehensibility and transparency of reasoning. In addition, due to its strong ability in dealing with non-linear, uncertain and complex systems, fuzzy logic has been widely used for information processing and mechanical control [22, 23, 48, 106, 128, 191].

1.1 Fuzzy Inference System

Fuzzy sets and fuzzy logic theory offers a formal way of handling vague information that arises due to the lack of sharp distinctions or boundaries between pieces of information. Fuzzy logic theory is based on the fuzzy sets which are a natural extension of the classical crisp set theory, where the elements have varying degrees of membership instead of being true or false in the classical boolean logic. In particular, in the crisp set theory, the membership value of an element takes only two values: 0 or 1. Fuzzy logic defines the concept of the fuzzy sets that use membership functions to represent the relationships between elements and their degrees of membership, expressed in the range of $[0,1]$ [65]. For instance, classical crisp set theory can only deal with the concepts of pure 'Black' and pure 'White', but nothing in the middle as grey, as shown in Fig. 1.1(a). However, in fuzzy logic, the grey area between the 'Black' and 'White' belongs to complete 'Black' and complete 'White' to a certain degree, and the membership can be represented as a fuzzy set, as illustrated in Fig 1.1(b), where $\mu(x)$ denotes the degree of membership of element x in the fuzzy set.



(a) Crisp set (b) Fuzzy set

Fig. 1.1 Crisp set and fuzzy set

Fuzzy inference systems (FIS) are developed based on fuzzy logic and fuzzy sets theory, which basically provide a mapping mechanism to map an input space to an output space [53].

With an inherent ability to model uncertain and imprecise knowledge in complex and non-linear datasets, as well as to effectively represent and reason on natural human language, fuzzy inference system is considered as an advanced methodology in many fields, including but not limited to control theory, decision-making and machine learning [185, 218]. The general architecture of a fuzzy inference system is shown in Fig. 1.2, which is comprised of four conceptual components: a *fuzzifier*, an *inference engine*, a *rule base* and a *defuzzifier*.

- *Fuzzifer*: Converting the input, either fuzzy or crisp, to membership values of the rule antecedents in the fuzzy knowledge base, also termed as the fuzzy rule base.
- *Inference engine*: Performing fuzzy inference, which uses fuzzy *IF-THEN* rules to map fuzzy inputs to fuzzy outputs.
- *Rule Base*: Containing a number of fuzzy *IF-THEN* rules as well as the definition of membership functions of the fuzzy sets used in the fuzzy rules.
- *Defuzzifier*: Converting fuzzy outputs generated by the inference engine to crisp values for real-world use.

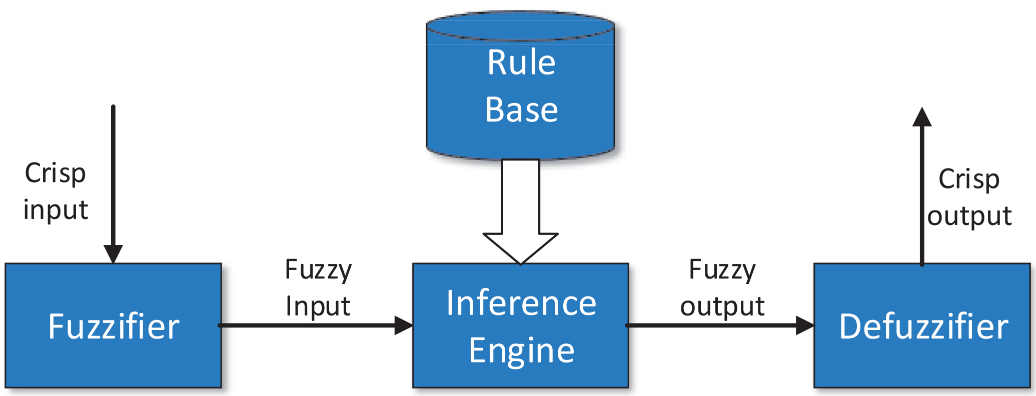


Fig. 1.2 Fuzzy inference system

Essentially, each inference step takes an input, which can be either crisp or fuzzy. Then the inference engine fires the rules in the rule base based on matching degree between the input and each rule antecedents, and the output is the aggregation of the inferred results from all fired rules. A number of inference engines have been developed, with the Mamdani method [127] and the TSK approach [177] being the most widely used. The Mamdani fuzzy model is more intuitive and suitable for handling linguistic inputs; its outputs are usually fuzzy sets, and thus a defuzzification process is often required. In contrast, the TSK inference approach produces crisp outputs directly, as TSK fuzzy rules use polynomials as rule consequences. Therefore, the expressive power and interpretability of Mamdani outputs are reduced in the TSK systems since the consequences of the rules are crisp values [74]. There are two types of rule bases used to support the two fuzzy inference engines, which are Mamdani-style and TSK-style rule bases accordingly.

Conventionally, a fuzzy rule base can either be directly translated from human expert knowledge, called knowledge-driven approaches [21, 124, 206, 208], or be extracted from data by applying machine learning techniques, termed as data-driven approaches [143, 192, 217]. Data-driven rule base generation was proposed to minimise the involvement of human experts, in an effort to automate the generation of rule bases during the system modelling process. Rule base generation from data is usually processed in two steps. Firstly, a raw rule base is generalised from the data, by which the fuzzy partitions of variable domains and the number of rules are determined. For instance, the neuro-fuzzy system has been employed to initialise membership functions and to extract fuzzy rules from a large dataset [64]. Secondly, the raw rule base is fine-tuned using optimisation algorithms, with the Genetic Algorithm (GA) being a popular choice [32].

1.2 Fuzzy Interpolation

The conventional fuzzy inference approaches, such as Mamdani and TSK, are only workable with dense rule bases. Briefly, a dense rule base cover the entire input domain, that is, every input condition is covered by at least one rule. Given an input (also termed as an observation), the output (or conclusion) can only be obtained from the fired rules that intersect with the given observation, if the conventional fuzzy inference approaches are applied. However, if there is only a sparse rule base available, which is led by the lack of expertise or data, and a given input does not overlap with any rule antecedent, no rule can be fired. As a consequence, no result can be inferred by employing the conventional fuzzy inference approaches.

Fuzzy interpolation alleviates the problem of lack of expertise or data during rule base generation, as fuzzy interpolation enables the performance of inferences upon sparse rule bases [95]. When a given observation does not overlap with any rule antecedent values, the conventional fuzzy inference systems cannot be applied. However, fuzzy interpolation, through a sparse rule base, can still obtain certain conclusions and thus improve the applicability of fuzzy models. Fundamentally, fuzzy interpolation techniques were originally developed based on the concept of a crisp linear interpolation mechanism, which interpolates the results from the two closest neighbouring rules defined by the distances between the given input and rule antecedents. Fuzzy interpolation is not only able to relax the requirement of the conventional fuzzy inference system, but is also able to simplify very complex fuzzy models by removing those rules that can be approximated by their neighbouring ones. Various fuzzy rule interpolation methods have been developed in the literature, including [79, 80, 121–123].

1.3 Motivations

Common to both types of rule base generation methods, either knowledge-driven approaches or data-driven approaches, a priori knowledge (expertise or data) is always required for system modelling. Although fuzzy interpolation techniques enhance the power of conventional fuzzy inference approaches by allowing the fuzzy inference still to be performed over sparse rule bases, it is still difficult for real-world applications to obtain sufficient a priori knowledge to generate a sparse rule base to support fuzzy interpolation reasoning. That will lead to the difficulty in applying the fuzzy inference approaches for such real-world applications. In addition, the situations may change over time, and the fixed rule bases generated by conventional methods will fail. For instance, although the fuzzy inference system has been widely applied in the smart home control systems, such as [43, 186], the rule base of such systems was commonly pre-defined and transformed from expertise. Such pre-defined rules not only fail to accurately reflect the lifestyles of different groups of people, but also hardly evolve themselves to adapt to the new situation once the users' daily routine changed. Can a fuzzy rule base be created without any a priori knowledge as well as evolve itself for changing situations? Therefore, an automatic rule base generation and adaptation system need to be considered to allow the creation of fuzzy rules with minimal or even without any a priori knowledge, as well as to support the changing situations through self-evolvement.

In addition, as stated earlier, fuzzy interpolation techniques relax the coverage requirement of the rule base by approximating rules using their neighbouring ones, which allow an inference consequent still to be obtained in the case of lacking of knowledge in the rule base. However, all existing fuzzy interpolation approaches were developed based on the Mamdani-style rule bases, which are not applicable for the TSK-style rule bases. It significantly restricts the applicability of the TSK fuzzy inference systems. How does a crisp output be obtained from a sparse TSK-style rule base? As this reason, to enable an fuzzy interpolation reasoning over sparse TSK-style rule bases is also required to be investigated.

1.4 Aims and Objectives

This PhD thesis aims to 1) develop a novel fuzzy interpolation approach, which enables TSK fuzzy inference still to be performed over sparse TSK fuzzy rule bases, 2) propose an automatic rule base generation and adaptation system that allow the rule bases to be generated with minimal or even without any a priori knowledge to support the fuzzy interpolation reasoning. In addition, to propose a mechanism that keeps the rule base to be evolved themselves in order to support the changing situations in real time. The following objectives will be achieved by this thesis:

- To investigate the existing fuzzy interpolation techniques.
- To develop a novel fuzzy interpolation approach to support not only the type-1 but also the interval type-2 TSK fuzzy models, thus to allows the crisp inference results to be obtained from sparse, dense or imbalanced TSK-style rule bases.
- To develop an automatic rule base generation and adaptation system, which allows the creation of fuzzy rules with minimal or even without any a priori knowledge, and to support the changing situations through self-evolvement.
- To evaluate the proposed approaches by applying real-world applications, including the network intrusion detection system, the dynamic quality of service solution for the enterprise network, and the inverted pendulum control problem.

1.5 Structure of the Thesis

The structure of the remainder of the thesis is outlined in this section. Briefly, the works carried out in Chapter 2, Chapter 5, Chapter 6 and 9 achieved the first objective. Chapter 3 and 4 are linked to the second objective. The third objective is achieved by the works detailed in Chapter 5. And the various applications of the proposed approaches, which are the fourth objective, are described in Chapter 6 to 9.

Chapter 2: Background

This chapter provides the background theories of the fuzzy inference system, fuzzy interpolation techniques, and the fuzzy rule base generation methods. In particular, two groups of fuzzy interpolation approaches, namely resolution principle-based and analogy-based fuzzy interpolation approaches, are reviewed, as well as the limitation of existing approaches and possible improvements. In addition, the two types of rule base generation methods, i.e., the knowledge-driven and data-driven approaches, are also reviewed in this chapter.

Chapter 3: TSK+ Fuzzy Inference System

The novel fuzzy interpolation approach, namely TSK+ fuzzy inference approach, is presented in this chapter. Briefly, the proposed TSK+ fuzzy interpolation approach was an extension of the conventional TSK fuzzy inference approach. In order to facilitate such an extension, a new similarity degree measurement was introduced first, which calculates the similarity degrees between observations and each rule antecedent in the rule base. Different from the similarity measure used in the conventional TSK approach, the introduced one always generates similarity degrees greater than 0, even when the observation and the rule antecedent do not overlap at all. Then, the TSK fuzzy model is extended using this new matching degree to obtain crisp inference results from sparse TSK fuzzy rule bases. The proposed fuzzy inference engine is workable with sparse, dense or imbalanced TSK-style rule bases. In addition, a data-driven TSK-style rule base generation approach was also developed to extract compact and concise TSK rule bases from incomplete, imbalanced, normal or over-dense datasets. Note that sparse and imbalanced datasets are still commonly seen, regardless of the magnitude of the datasets in the era of big data. The proposed approach has been applied to two benchmark problems to demonstrate the power of the proposed approach in enhancing the conventional TSK inference method by means of broader applicability and better system efficiency, and competitive performance in reference to other machine learning approaches. The work in this chapter has been published in [111, 115].

Chapter 4: Interval Type-2 TSK+ Fuzzy Inference System

The proposed TSK+ fuzzy inference in Chapter 3 is extended in this chapter to allow the utilisation of IT2 TSK fuzzy rule bases. The extended fuzzy inference system TSK+ is able to work with: 1) sparse rule bases, 2) dense rule bases, 3) Type-1 fuzzy sets, and 4) interval type 2 fuzzy sets. The work presented in this chapter has been published in [114].

Chapter 5: Experience-based Rule Base Generation and Adaptation

This chapter reports an initial investigation into automatic rule base generation and adaptation from very limited a priori knowledge. The proposed system mimics the pedagogic approach of experience-based learning, which is driven by the grasping and transferring of the proceeding performance experiences. In order to implement such a system, a weight value is associated with each fuzzy rule, which is the integrated information of the rule's previous experience, including the usage frequency and the performance information. Then, two neighbouring rules are selected by a combined metric of the weights of rules and the distances from these rules to the given input. After a fuzzy interpolation step is performed, a performance index, which is generally available for most of the intelligent control problems, is employed to update the weights of rules regarding the previous performance information. Based on the updated usage frequency and previous performance information, the rule base is updated by means of adding a new rule or deleting an existing rule, which is either out-of-date or proven to be faulty. Dissimilar with most of the existing data-driven approaches, which require sufficient training data, the proposed method collects data while it is performing inferences. By such a mechanism, the system benefits from both good and bad performance. In particular, the bad inference steps are analysed, and the dependent rules of such bad inference steps are traced and modified, which helps significantly in quickly generating rule bases and also adapting the existing rule base to a changed problem. The presented system is evaluated by numerical simulations. The evaluation results confirmed that the system is not only able to generate a rule base to support the fuzzy interpolation reasoning while performing inference but also able to adapt itself to a new situation when the circumstances are changed. The work documented in this chapter has been published in [112].

Chapter 6: Intelligent Home Heating Control by Fuzzy Rule Interpolation
This chapter proposes an application of the conventional fuzzy interpolation technique, which is a novel smart home heating control system by efficiently utilising the personal data captured by smart portable devices. The system can enable the preheating of the home by predicting when the residents will arrive home and thus addresses the limitation of the existing heating controllers by using them as complementary parts of the existing ones. The work discussed in this chapter has been published in [116].

Chapter 7: Network Intrusion Detection by TSK+
This chapter presents an application of the proposed TSK+ fuzzy interpolation approach, the network intrusion detection system (NIDS). In particular, the data-driven method is adopted in the proposed NIDS to generate a sparse TSK rule base. By employing the TSK+ inference approach, the proposed NIDS system is not only able to generate security alerts for already

known attack types, but is also able to detect potential unknown network threats. The works in this chapter have been published in [207].

Chapter 8: Dynamic Quality of Service (QoS) Solution for Enterprise Network by TSK+

The proposed TSK+ fuzzy interpolation approach is also applied in enterprise network environments to help manage the priority of the network traffic in real time. This chapter presents a dynamic QoS solution based on the differentiated services (DiffServ) approach for enterprise networks, which is able to modify the priority level of a packet in real time by adjusting the value of the Differentiated Services Code Point (DSCP) in the Internet Protocol (IP) header of network packets. This is implemented by a 0-order TSK fuzzy model with a sparse rule base, which is developed by considering the current network delay, the application desired priority level and the user current priority group. DSCP values are dynamically generated by the TSK fuzzy model and updated in real time. The proposed system has been evaluated in a real network environment with promising results generated. The works in this chapter have been published in [113].

Chapter 9: Inverted Pendulum Control by FRI with Experience-based Rule Base Generation

This chapter presents an application of FRI with the proposed rule base generation and adaptation method to the well-known control problem, the inverted pendulum. In particular, the system takes the pendulum's angle as the input to produce the output, which is the input voltage to the motor responsible for arm moving. The system starts with an empty rule base, and initialises the rule base by randomly moving the arm for balancing the inverted pendulum. After a number of iterations, the system is able to balance the pendulum with a reasonable rule base generated.

Chapter 10: Conclusion

This chapter concludes the thesis with a summary of the achievements of the research presented, together with a discussion of possible future directions for research and potential areas for implementation of the work.

Appendices

Appendix A lists the initialised rule base generated by the proposed TSK+ rule base generation approach for the network intrusion detection system.

Appendix B lists the optimised rule base by using the GA for the network intrusion detection system.
Appendix C lists the publications arising from work presented in this thesis.

Chapter 2

Background

Conventional fuzzy inference systems, such as the Mamdani [127] and the TSK inferences [177], are only applicable to dense rule bases by which the entire input domains are fully covered; otherwise, no rule can be fired when a given observation does not overlap with any rule antecedent. Fuzzy interpolation techniques, initially proposed in [94], address such an issue, which allow the inference to still be performed over a sparse rule base, thus improving the applicability of fuzzy models. Fuzzy interpolation can also be used to reduce the complexity of fuzzy models by excluding rules that can be approximated by their neighbouring ones. Although a dense fuzzy rule base is not required by fuzzy interpolation reasoning, a sparse rule base is still needed.

A fuzzy rule base is the core part of a fuzzy inference system that contains the knowledge which is utilised by the reasoning mechanism of the system. Fuzzy rule base generation has been intensively studied in the literature and is usually implemented in two ways: knowledge-driven approaches (generating rules from human expert knowledge) [86] and data-driven approaches (extracting rules from existing datasets) [34, 192]. Both types of rule base generation approaches provide the ability to generate a reasonable rule base for system use, although they may target different problems. This chapter systematically reviews the literature in fuzzy interpolation and the corresponding fuzzy rule base generation approaches.

The rest of this chapter is organised as follows. Chapter 2.1 discusses different fuzzy interpolation approaches. Chapter 2.2 explains the two types of the rule base generation approaches, i.e., the knowledge-driven and data-driven approaches, as well as the rule base simplification methods. Finally, Chapter 2.3 summarises the chapter.

2.1 Fuzzy Interpolation

Given an input, termed as an observation, conventional fuzzy inference systems integrate the final output by considering all overlapped rules based on the matching degree between the given input and rule antecedent. However, if the system can not find any rule that its rule antecedent overlaps with the given observation, no conclusion will be determined by using conventional fuzzy inference mechanism s[52, 54, 94–96, 202]. For instance, assume that there are only two fuzzy rules in the rule base, which are given below:

$$\begin{aligned} &R_1 : \textbf{IF } \textit{service is poor} && \textbf{THEN } \textit{tip is cheap} \\ &R_2 : \textbf{IF } \textit{service is excellent} && \textbf{THEN } \textit{tip is generous}. \end{aligned} \tag{2.1}$$

For a given observation that does not overlap with any rule antecedent, shown as:

$$Observation = \textit{service is average} \;, \tag{2.2}$$

the conclusion for this observation could not be obtained by using conventional fuzzy inference methods, as no rule could be fired. If the triangular membership functions are applied to represent the linguistic variables, the above example can be illustrated in Fig. 2.1. Fuzzy interpolation techniques were developed to strengthen the power of fuzzy inference, which supports reasoning on a sparse fuzzy base. Therefore, in the above case, a certain conclusion is still able to be obtained by employing the fuzzy interpolation techniques.

A number of fuzzy interpolation approaches have been developed in the literature, e.g. [8, 30, 35, 79, 80, 83, 85, 97, 100, 122, 123, 134, 169, 183, 204], which can typically be grouped into two classes: resolution principle-based approaches and analogy-based approaches. The resolution principle-based approaches represented each fuzzy set as a series of α-cuts ($\alpha \in (0,1]$), and the α-cuts of the consequence are calculated from the α-cuts of the observation and of all involved rules. The final fuzzy set is assembled from all the α-cuts of consequent fuzzy set by adopting the resolution principle [93, 159, 219]. Different from resolution principle-based methods, the analogy-based approaches work by first creating an intermediate rule such that its antecedent is as 'close' (given a fuzzy distance metric) to the given observation as possible. Then, a conclusion is derived from the given observation by firing the generated intermediate rule through the analogical reasoning mechanism. That is, the shape distinguishability between the resultant fuzzy set and the consequence of the intermediate rule is analogous to the shape distinguishability between the observation and

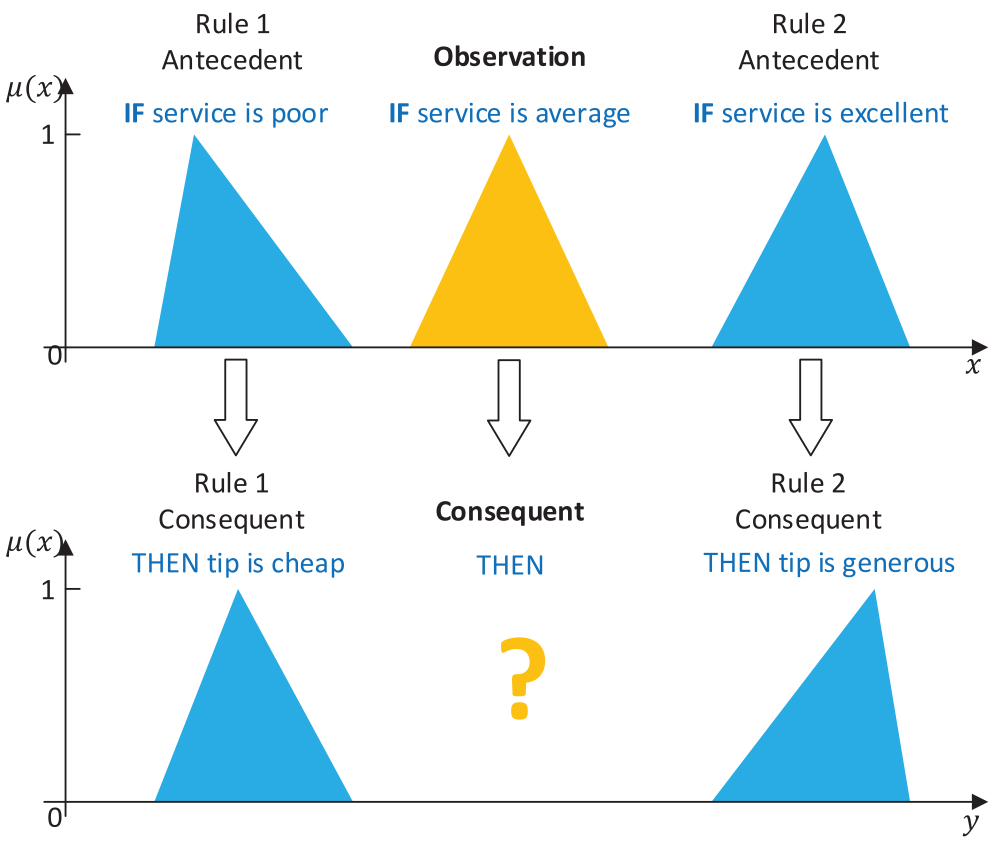


Fig. 2.1 Example of general fuzzy interpolation

the antecedent of the generated intermediate rule [18]. The rest of this chapter presents an overview of two classes of such fuzzy interpolation techniques.

2.1.1 Resolution Principle-based Interpolation Approaches

Based on the reasoning processes, the resolution principle-based interpolation can also be termed as single step rule interpolation [214], which directly interpolates a consequence by considering the given observation and existing rules. The types of fuzzy interpolation approaches are computationally efficient. This method is developed by adopting two fundamental principles: the Decomposition Principle and the Resolution Principle [93, 99, 219]. Accounting to these principles, each fuzzy set A is able to be denoted by a series of α-cut intervals, represented as $A_\alpha, \alpha \in (0,1]$. Given a α $(0 < \alpha \leq 1)$, the α-cut of the interpolated consequence fuzzy set can be computed from the α-cut of the observation fuzzy set and all the fuzzy sets involved in the rules used for interpolation. The final interpolated consequence fuzzy set can then be integrated from all computed α-cuts of the consequent fuzzy set for all $\alpha \in (0,1]$ by employing the decomposition principle.

2.1.1.1 KH Approach

The Kóczy-Hirota approach, termed as the KH approach, is the first proposed resolution principle-based fuzzy interpolation method that has been developed based on the concept that the proportional distance between the estimated conclusion and the consequent sets of used rules should be the same as the distance between the observation and the rule antecedents of those rules [94–97]. An important notion in this approach is that all fuzzy sets involved in the reasoning process must satisfy a partial ordering, denoted as '$\prec$'[220]. In particular, given two normal and convex fuzzy sets A_1 and A_2, which are associated with the partial ordering, represented by $A_1 \prec A_2$, if $\forall \alpha \in (0,1]$, the following conditions will be satisfied:

$$\inf\{A_{1\alpha}\} < \inf\{A_{2\alpha}\} \text{ and } \sup\{A_{1\alpha}\} < \sup\{A_{2\alpha}\} \;, \tag{2.3}$$

where $A_{1\alpha}$ and $A_{2\alpha}$ are the α-cut sets of A_1 and A_2, respectively, $\inf\{A_{i\alpha}\}$, $i = \{1,2\}$ is the infimum of $A_{i\alpha}$ and $\sup\{A_{i\alpha}\}$ denotes the supremum of $A_{1\alpha}$.

For simplicity, suppose a sparse rule base only contains two signal-antecedent fuzzy rules, which has been given in Eq.2.4.

$$\begin{aligned} &R_1 : \textbf{IF } x \textit{ is } A_1 \quad \textbf{THEN } y \textit{ is } B_1 \\ &R_2 : \textbf{IF } x \textit{ is } A_2 \quad \textbf{THEN } y \textit{ is } B_2, \end{aligned} \tag{2.4}$$

where A_1, A_2, B_1, B_2 are normal and convex fuzzy sets that triangular membership function are particularly used in this illustration, and assume that they are not overlapped each other. For a given observation A^*, which satisfies $A_1 \prec A^* \prec A_2$, a consequent B^* can be determined as:

$$\frac{D(A_{1\alpha}, A^*_\alpha)}{D(A_{2\alpha}, A^*_\alpha)} = \frac{D(B_{1\alpha}, B^*_\alpha)}{D(B_{2\alpha}, B^*_\alpha)} \; , \forall \alpha \in (0,1] \;, \tag{2.5}$$

where $D(.,.)$ is typically the Euclidean distance between two α-cut sets. For instance, given two fuzzy sets A_1, A_2 and $\forall \alpha \in (0,1]$, which are illustrated in Fig. 2.2, the distance $D(A_{1\alpha}, A_{2\alpha})$ can be defined by the interval $[D_L(A_{1\alpha}, A_{2\alpha}), D_U(A_{1\alpha}, A_{2\alpha})]$, which is defined by:

$$\begin{aligned} D_L(A_{1\alpha},A_{2\alpha}) &= D(\inf\{A_{1\alpha}\},\inf\{A_{2\alpha}\}) &= \inf\{A_{2\alpha}\}-\inf\{A_{1\alpha}\} \\ D_U(A_{1\alpha},A_{2\alpha}) &= D(\sup\{A_{1\alpha}\},\sup\{A_{2\alpha}\}) &= \sup\{A_{2\alpha}\}-\sup\{A_{1\alpha}\}\ . \end{aligned} \tag{2.6}$$

From Eq. 2.6, Eq. 2.5 can be rewritten as:

$$\begin{cases} \inf\{B^*_\alpha\} = \dfrac{\frac{\inf\{B_{1\alpha}\}}{D_L(A_{1\alpha},A^*_\alpha)}+\frac{\inf\{B_{2\alpha}\}}{D_L(A_{2\alpha},A^*_\alpha)}}{\frac{1}{D_L(A_{1\alpha},A^*_\alpha)}+\frac{1}{D_L(A_{2\alpha},A^*_\alpha)}} \\ \sup\{B^*_\alpha\} = \dfrac{\frac{\sup\{B_{1\alpha}\}}{D_U(A_{1\alpha},A^*_\alpha)}+\frac{\sup\{B_{2\alpha}\}}{D_U(A_{2\alpha},A^*_\alpha)}}{\frac{1}{D_U(A_{1\alpha},A^*_\alpha)}+\frac{1}{D_U(A_{2\alpha},A^*_\alpha)}}\ . \end{cases} \tag{2.7}$$

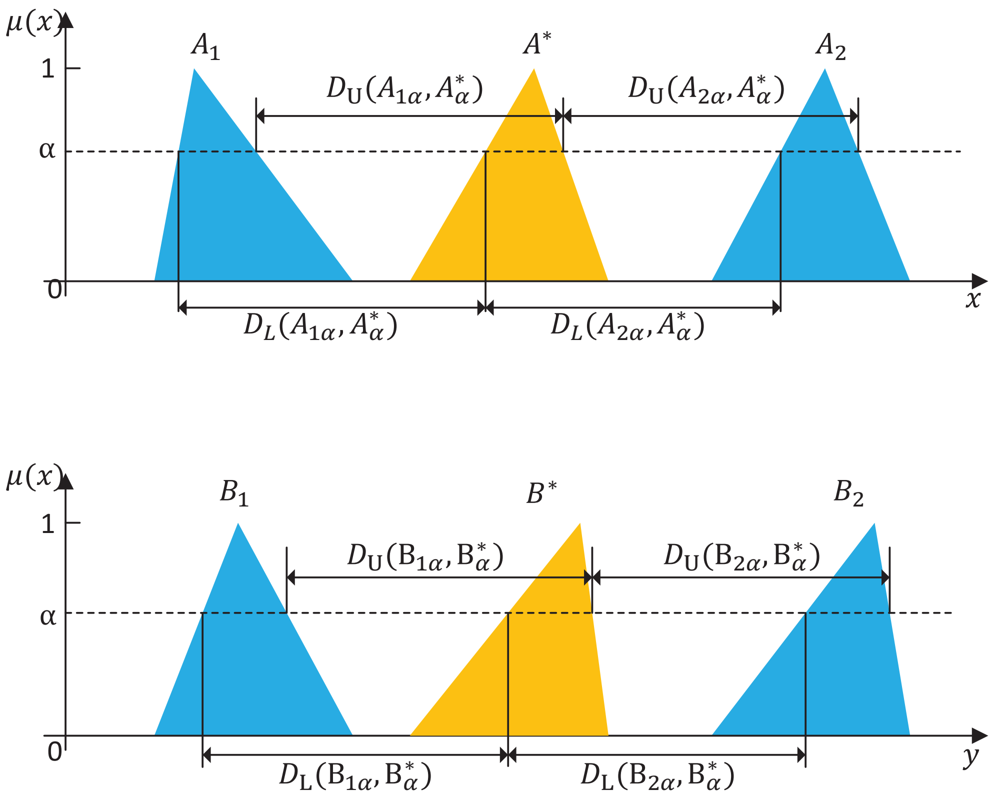


Fig. 2.2 KH approach distance determination

Alternatively, let

$$\begin{cases} \gamma_{L_\alpha} & = & \frac{D_L(A_{1\alpha},A^*_\alpha)}{D_L(A_{2\alpha},A^*_\alpha)} & = & \frac{\inf\{A^*_\alpha\}-\inf\{A_{1\alpha}\}}{\inf\{A_{2\alpha}\}-\inf\{A^*_\alpha\}} \\ \\ \gamma_{U_\alpha} & = & \frac{D_U(A_{1\alpha},A^*_\alpha)}{D_U(A_{2\alpha},A^*_\alpha)} & = & \frac{\sup\{A^*_\alpha\}-\sup\{A_{1\alpha}\}}{\sup\{A_{2\alpha}\}-\sup\{A^*_\alpha\}} \end{cases}, \tag{2.8}$$

where γ is represented as a *relative placement factor*, which expresses the overall relative location of the observation between rule antecedents of used rules for interpolation [209]. From this, Eq. 2.7 can be rewritten as:

$$\begin{cases} \inf\{B^*_\alpha\} & = & (1-\gamma_{L_\alpha})\inf\{B_{1\alpha}\}+\gamma_{L_\alpha}\inf\{B_{2\alpha}\} \\ \\ \sup\{B^*_\alpha\} & = & (1-\gamma_{U_\alpha})\sup\{B_{1\alpha}\}+\gamma_{U_\alpha}\sup\{B_{2\alpha}\} \end{cases}. \tag{2.9}$$

It is clear from Eq. 2.9 that the α-cut set for consequence B^* is generated, where $B^*_\alpha = [\inf\{B^*_\alpha\},\sup\{B^*_\alpha\}]$. The final consequence B^* can then be constructed by:

$$B^* = \bigcup_{\alpha\in(0,1]} \alpha B^*_\alpha . \tag{2.10}$$

The KH fuzzy interpolation approach has also been extended to deal with more complex cases, such as a multi-antecedent and signal-consequence systems. The working procedure of such cases is similar to the single-antecedent situation. In these instances, given an $\alpha, \alpha \in (0,1]$, a set of interval distances between the α-cut of each observation and the α-cut of its counterpart rule antecedents can be first determined. Those distances may have a different metric, as the various domains represent the different attributes. In order to describe a uniform metric over the whole space, a normalisation process is required. Then, the *relative placement factor* (γ) for the overall distance between the observation and the antecedents of a rule is calculated by applying the Minkowski distances instead of the Euclidean distance based on the normalised distance intervals in each input domain. Finally, the consequence of the given observation can be readily assembled in the same way as for the signal-antecedent situation.

In addition, the KH fuzzy interpolation approach has also been developed to deal with the problem of the fuzzy extrapolation. Assume that a given observation A^* is located outside of two fuzzy sets A_1 and A_2, where $A_1 \prec A_2 \prec A^*$ as shown in Fig. 2.3, the distances between A_1 and A^*, A_2 and A^*, B_1 and B^* and B_2 and B^* are illustrated in Fig. 2.3. The consequent B^* can be constructed as the same why as the KH interpolation approach by Eq. 2.10.

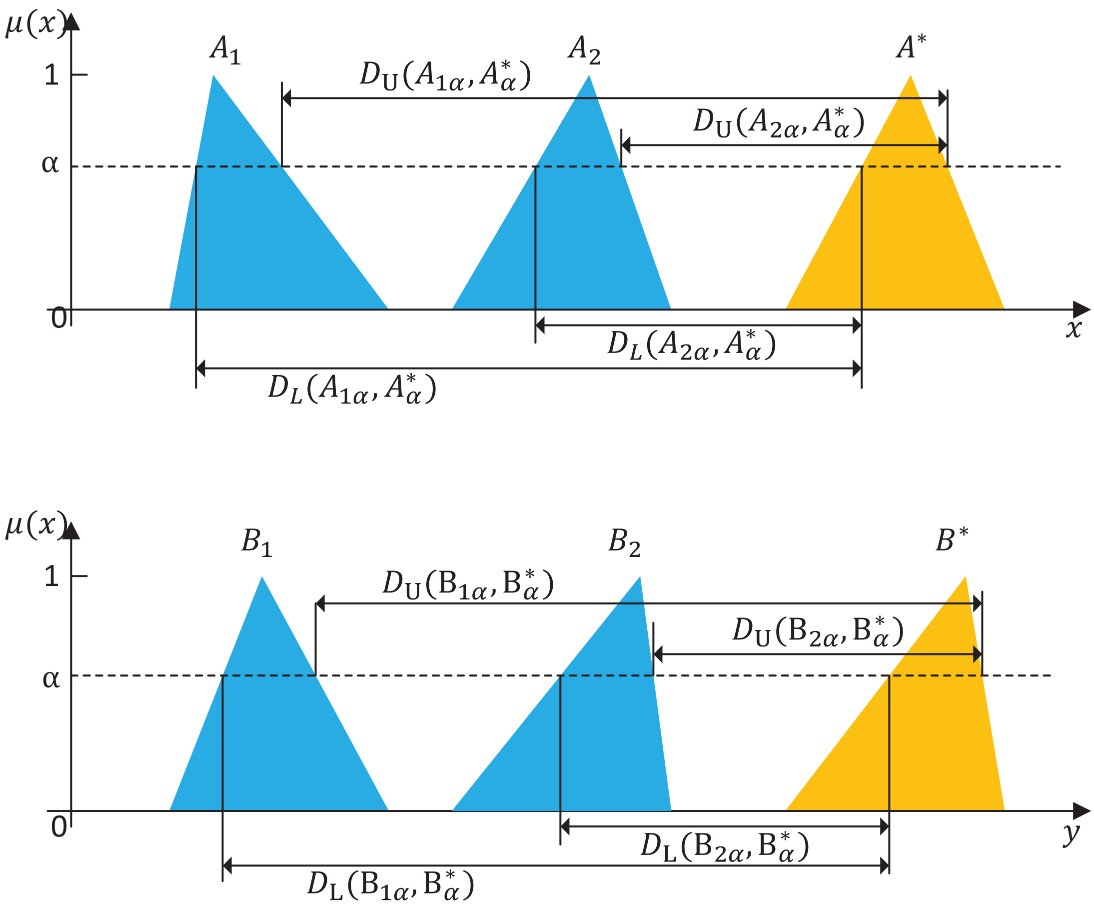


Fig. 2.3 KH fuzzy extrapolation approach distance determination

The KH approach is one of the simplest and earliest fuzzy interpolation approaches, and developed based on the concept of the α-cut and the classical linear interpolation mechanism. As it has the advantage of light computational complexity, this approach may be suitable for real-time application. Also, it is more suitable to deal with triangular and trapezoidal fuzzy sets, as those types of shapes can be easily represented by a number of characteristic points in the α-cuts process. Although the KH approach ensures the fast response performance of real-time applications, it may lead to 'the abnormal problem', which is to generate invalid interpolated results, such as a non-convex fuzzy set [203]. In order to eliminate this deficiency, a number of modifications or improvements have been proposed, including [29, 39, 54, 123, 183, 184, 205].

2.1.2 Analogy-based Interpolation Approaches

The analogy-based interpolation approaches are in the second class of the fuzzy interpolation methods, which have been developed based on the analogical reasoning mechanism [18]. This type of approaches artificially creates an intermediate rule during the interpolation such that the antecedent of the intermediate rule is as 'close' to the given observation as possible. Then, a consequence can be determined through the analogical reasoning mechanism, which considers the shape distinguishability between the given observation and the antecedent of the created intermediate rule. The scale and move transformation-based fuzzy interpolation (T-FRI) [77–80, 169] is a typical approach in this class.

2.1.2.1 T-FRI Approach

This method has developed based on the concepts of Representative Value, Scale Transformation, Move Transformation and the Analogical approach. Particularly, during the reasoning, an intermediated rule is constructed where its antecedent is close (or with the shortest distance) to and has the same representative value as the observed antecedent fuzzy set. And then the interpolative reasoning is performed by firing the created intermediated rule through the scale and move transformation, thus generating a final conclusion. The T-FRI approach is able to handle both signal-antecedent and multiple-antecedent situations, as well as the various shapes of fuzzy sets in fuzzy rules, including triangular, complex polygonal and Gaussian or bell-shaped fuzzy membership function [77, 80]. Compared with the KH approach, which may generate invalid interpolated results, the T-FRI approach guarantees the uniqueness as well as the normality and convexity of the resulting interpolated fuzzy sets.

For simplicity, the illustrated example used in Chapter 2.1.1 is re-used in this chapter for better explanation. Given a fuzzy rule base that is comprised of two signal-antecedent and signal-consequence fuzzy rules, listed below:

$$\begin{aligned} &R_1 : \textbf{IF } x \textit{ is } A_1 \quad \textbf{THEN } y \textit{ is } B_1 \\ &R_2 : \textbf{IF } x \textit{ is } A_2 \quad \textbf{THEN } y \textit{ is } B_2 \, , \end{aligned} \tag{2.11}$$

where each variable value A_i $(i = \{1,2\})$ is represented as a triangular fuzzy set and is conveniently denoted as (a_{i1}, a_{i2}, a_{i3}), and where (a_{i1}, a_{i3}) is the support of the fuzzy set and a_{i2} is the normal point, and given an observation ($A^* = (a_1^*, a_2^*, a_3^*)$), the calculation process of the conclusion using the T-FRI approach can be summarised in the following steps.

Step 1: Calculate the *representative value* of the given observation and each fuzzy set involved in the neighbouring rules for interpolation using the following:

$$Rep(A) = \frac{a_{i1} + a_{i2} + a_{i3}}{3},\ i = \{1,2\}\ . \tag{2.12}$$

Note that, the *representative value* of a fuzzy set reflects both the overall location and the shape of this fuzzy set, which is usually defined as the average of the x coordinate values.

Step 2: Compute the *relative placement factor* based on the relative location of the observation regarding the two antecedents:

$$\lambda = \frac{D(Rep(A_1), Rep(A^*))}{D(Rep(A_1), Rep(A_2))}\ , \tag{2.13}$$

where $D(Rep(A_1), Rep(A_2))$ represents the distance between two fuzzy sets A_1 and A_2, which is defined by:

$$D(Rep(A_1), Rep(A_2)) = Rep(A_2) - Rep(A_1)\ . \tag{2.14}$$

Step 3: Obtain the antecedent of the new intermediate rule $A^{'} = (a_1^{'}, a_2^{'}, a_3^{'})$ using:

$$\begin{cases} a_1^{'} = (1-\lambda)a_{11} + \lambda a_{21} \\ a_2^{'} = (1-\lambda)a_{12} + \lambda a_{22} \\ a_3^{'} = (1-\lambda)a_{13} + \lambda a_{23}\ . \end{cases} \tag{2.15}$$

Step 4: By comparing the areas of $A^{'}$ and A^*, obtain the Scale Rate (s) using the following:

$$s = \frac{a_3^{'} - a_1^{'}}{a_3^* - a_1^*}\ . \tag{2.16}$$

Step 5: Apply scale rate s to $A^{'}$ to obtain $A^{''}$ using the following equations:

$$\begin{aligned} a_1^{''} &= \frac{a_1^{'}(1+2s) + a_2^{'}(1-s) + a_3^{'}(1-s)}{3} \\ a_2^{''} &= \frac{a_1^{'}(1-s) + a_2^{'}(1+2s) + a_3^{'}(1-s)}{3} \\ a_3^{''} &= \frac{a_1^{'}(1-s) + a_2^{'}(1-s) + a_3^{'}(1+2s)}{3}\ . \end{aligned} \tag{2.17}$$

Step 6: By comparing the shape difference between A^* and $A^{''}$, obtain the Move Transformation Rate ($\mathbb{M}$).

$$\mathbb{M} = \begin{cases} \frac{3(a_1^{''} - a_1^*)}{a_2^* - a_1^*}, & if\ a_1^{''} \geq a_1^* \\ \frac{3(a_1^{''} - a_1^*)}{a_3^* - a_2^*}, & if\ a_1^{''} < a_1^*. \end{cases} \tag{2.18}$$

Step 7: Compute the consequence B' of the interpolated rule using Eq. 2.15 in the same way as for the antecedent of the intermediate rule, as given in Step 3.
Step 8: Apply the determined scale rate s to $B^{'}$ to calculate the consequence $B^{''}$ of the interpolated rule using Eq. 2.17 in the same way as for the antecedent of the intermediate rule, as expressed in Step 5.
Step 9: Obtain the consequence B^* of the given observation by applying $\mathbb{M}$ to $B^{''}$:

$$\begin{cases} \begin{cases} b_1^* & = & b_1^{''} + \mathbb{M}\frac{b_2^{''} - b_1^{''}}{3} \\ b_2^* & = & b_2^{''} - 2\mathbb{M}\frac{b_2^{''} - b_1^{''}}{3} \\ b_3^* & = & b_3^{''} + \mathbb{M}\frac{b_2^{''} - b_1^{''}}{3} \end{cases} & \text{if } \mathbb{M} \geq 0 \\ \\ \begin{cases} b_1^* & = & b_1^{''} + \mathbb{M}\frac{b_3^{''} - b_2^{''}}{3} \\ b_2^* & = & b_2^{''} - 2\mathbb{M}\frac{b_3^{''} - b_2^{''}}{3} \\ b_3^* & = & b_3^{''} + \mathbb{M}\frac{b_3^{''} - b_2^{''}}{3} \end{cases} & \text{if } \mathbb{M} < 0 \ \cdot \end{cases} \tag{2.19}$$

As mentioned previously, the T-FRI approach is also able to deal with complex situations such as a fuzzy rule containing a multi-antecedent. In such cases, each input dimension has its own parameter values for λ, s and $\mathbb{M}$. Obviously, all those values are supposed to contribute to the final conclusion. Therefore, an average aggregation method is employed in such complex situations to aggregate all of these values for the construction of the final conclusion, where

$$\lambda^{'} = \frac{1}{N}\sum_{j=1}^{N} \lambda_j \tag{2.20}$$

$$s^{'} = \frac{1}{N}\sum_{j=1}^{N} s_j \tag{2.21}$$

$$\mathbb{M}^{'} = \frac{1}{N}\sum_{j=1}^{N} \mathbb{M}_j\ , \tag{2.22}$$

and where N is the number of antecedent variables. From here, the interpolated conclusion B^* can then be calculated by using aggregated $\lambda^{'}, s^{'}$ and $\mathbb{M}^{'}$ in a similar way as presented above.

The T-FRI approach interpolates the conclusion based on the proposed scale and move transformation mechanism, which is well suited for polygonal-shaped fuzzy sets, as well as multi-dimensional situations. Although this approach considers the shapes of the fuzzy sets through the process of the interpolation, it is able to handle those fuzzy sets whose membership functions involve vertical slopes, such as a crisp number, very well. The main advantage of this approach is that it restricts all involved fuzzy sets to be convex and normal. Therefore, a valid conclusion can always be generated by using this approach. Despite extensive supporting of very complicated situations, this method may not always support a case where the given observation overlaps with the rule antecedent [109].

2.1.3 Adaptive Fuzzy Interpolation

Fuzzy interpolation techniques significantly strengthen the power of the fuzzy inference system, which not only provides a way to reduce the complexity of fuzzy systems by removing those rules that can be approximated by their neighbouring ones, but also improves the applicability of the systems by enabling the performance of fuzzy reasoning upon sparse rule bases. Common to both classes of fuzzy interpolation approaches introduced above is the fact that interpolation is carried out in a linear manner. This may conflict with the nature of some realistic problems, and consequently, this may lead to inconsistencies during the interpolation processes. For instance, variable x is used to represent a person's height. *x is tall* will be defined if comparing with a shorter person, and it is also possible that *x is short* may be held if comparing with a taller person. It is contradictory if *x is tall* and *x is short* are held simultaneously in one single situation. The adaptive fuzzy interpolation approach [121, 122, 210, 213] was proposed to address this issue. Fundamentally, this approach was developed based upon the T-FRI method, which is able to detect inconsistencies, locate possible fault candidates and modify the candidates in order to remove all the inconsistencies.

This method employed the assumption-based truth-maintenance system (ATMS) [91, 92] and the general diagnostic engine (GDE) [45] to support fuzzy interpolation by tracking and locating possible inconsistencies during the processes of the fuzzy interpolation, before making an effort to restore reasoning consistency once a fault had been located. In this method, each pair of neighbouring rules is defined as a *fuzzy reasoning component*, which takes certain fuzzy values as inputs and produces another fuzzy set as outputs. The reasoning

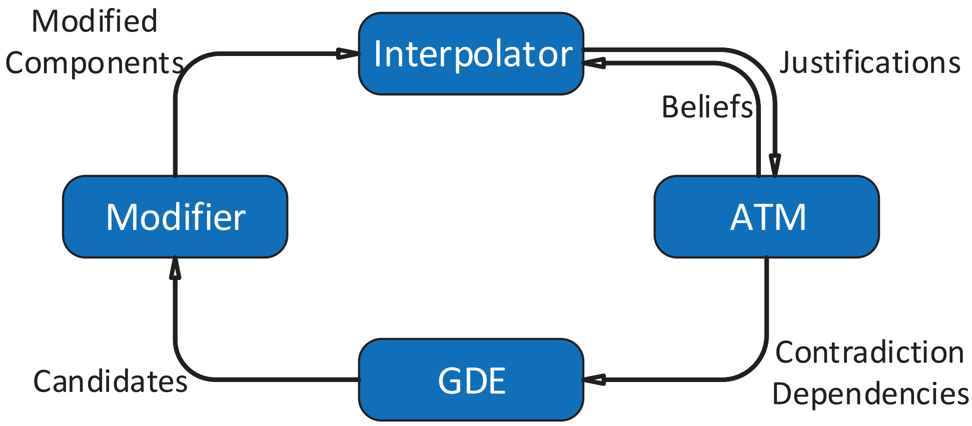


Fig. 2.4 Adaptive fuzzy interpolation reasoning process

process of the proposed adaptive fuzzy interpolation is summarised in Fig. 2.4. In particular, the interpolator performs the fuzzy interpolation and passes the interpolated results to the ATMS. And then, the ATMS records the dependencies between an interpolated result and its proceeding fuzzy interpolative reasoning components. From this, the ATMS relays any inconsistency or contradiction as well as their dependent fuzzy reasoning components to the GDE. And then, the GDE works out all possible candidates from those sets of contradictory dependent components. Finally, a modification process takes place to correct a certain candidate to restore consistency by modifying the original linear interpolation to piecewise linear interpolation.

2.1.3.1 Contradictions in Interpolation

In this work, the inconsistency between two different values is defined by an introduced degree of contradiction (β). Given the two propositions $P(x_1 \text{ is } A_{11})$ and $P^{'}(x_1 \text{ is } A_{12})$, where A_{11} and A_{12} are two fuzzy sets, the degree of contradiction between P and $P^{'}$ is defined by:

$$\beta = 1 - M(A_{11}, A_{12}) \;, \tag{2.23}$$

where $M(A_{11},A_{12})$ represents the matching degree between A_{11} and A_{12} in the domain D_{x_1} of variable x_1, which is defined by [24, 41]:

$$M(A_{11},A_{12}) = \sup_{x\in D_{x_1}} [\min(\mu A_{11}(x), \mu A_{12}(x))] \ . \tag{2.24}$$

Given a threshold of β_0 ($0 \leq \beta_0 \leq 1$), an inconsistency or a contradiction occurs, denoted as $\beta_0 - contradiction$, when the corresponding degree of contradiction is $\beta > \beta_0$.

2.1.3.2 Representation of Interpolation Concepts in ATMS

The ATMS is implemented in the work to record dependency of the interpolated results as well as contradictions in those fuzzy reasoning components from which they are inferred. Therefore, propositions, contradictions and fuzzy interpolative reasoning components are all represented as ATMS nodes. Typically, an ATMS node is formed as:

$$Node : < datum,\ label,\ justifications > \ , \tag{2.25}$$

where the *datum* is the physical meaning that the node represents.

A justification in an ATMS node describes how a node is derivable from other nodes. The three different main justification representations are defined:

1) For observation only: For a given observation O, the justification of its ATMS node has not antecedent, as it is supposed to hold universally, which is able to be represented as:

$$\Rightarrow O \ . \tag{2.26}$$

2) For inferred proposition: An ATMS node with an inferred proposition, which is inferred through fuzzy interpolation reasoning, can be represented by an ATMS justification as:

$$O, R_iR_j \Rightarrow C \ , \tag{2.27}$$

where R_iR_j denotes the fuzzy reasoning component which indicates the two fuzzy rules R_i and R_j that have been used to generate the conclusion C from the given observation O. In addition, if a node N is obtained by n other nodes, $M_1, M_2, \cdots, M-N$, by interpolation through two fuzzy rules R_i and R_j, its justification can be rewritten as:

$$M_1, M_2, \cdots, M_n, R_iR_j \Rightarrow N \ . \tag{2.28}$$

3) For propositions contradiction: Given two propositions, $P(x \text{ is } A_i)$ and $P^{'}(x \text{ is } A_j)$, if the degree of contradiction between P and $P^{'}$ is higher than a given threshold β_0, a $\beta_0 - contradiction$ is deduced, which is represented as:

$$P, P^{'} \Rightarrow_{\beta_0} \perp . \tag{2.29}$$

A label is a set of environments each supporting the associated node, which is extracted from its justification. An environment contains a minimal set of fuzzy reasoning components that jointly entail the node concerned, thereby describing how the node ultimately depends on those fuzzy reasoning components. For instance, suppose that an ATMS node P_3 is inferred from nodes P_1 and P_2 by the fuzzy reasoning component F_1 from observation C^*, whose justification is, therefore, $P_2, P_3, F_1 \Rightarrow P_1$, assume that the label of fuzzy reasoning component F_1 is $\{R_5R_6\}$, and the propositions P_1 and P_2 are $\{R_1R_2\}$ and $\{R_3R_4\}$, respectively, the ATMS node for P_3 can then be formed as:

$$P_3 \ : < x_3 = C^*, \{\{R_1R_2,\ R_3R_4,\ R_5R_6\}\}, \{P_1, P_2, F_1 \Rightarrow P_3\} > \ . \tag{2.30}$$

During the interpolation processes, the ATMS records each interpolated result and its proceeding interpolation components by generating the ATMS node. At the same time, the degree of contradiction β between two propositions is obtained. Once an inconsistency, termed as a $\beta - contradiction$, is detected, a contradiction node is created. Taking the above illustration (Eq. 2.30) as an example, if $\beta - contradiction$ has been detected in this case, a contradiction node will be created, shown in Eq. 2.31:

$$\perp_1 \ : < \perp,\ \{\{R_1R_2,\ R_3R_4,\ R_5R_6\}\},\ \{P_1, P_2 \Rightarrow_{\beta_0} \perp\} > \ . \tag{2.31}$$

In the end, a specific ATMS node 'false', which collectively represents all the contradiction, is generated by the ATMS label-updating algorithm [91], and this node will be passed to the GDE to diagnose the problem.

2.1.3.3 Minimal Candidate Generation by GDE

Traditionally, the GDE has been developed to address physical world problems, which are usually represented by component-based diagrams. In this work, the GDE is employed to generate minimal faulty reasoning component candidates by manipulating the label of the specific 'false' node. A minimal candidate is a possible minimal set of defective components, which need to be corrected at one time in order to remove all the contradictions. Obviously, inconsistencies that result are generated from the failure of interpolation; this is, at least one of its reasoning components is defective. Therefore, a minimal faulty reasoning component candidates mush has a non-empty intersection with each environment that the $\beta_0 - contradiction$ is derivable from. For instance, given a specific ATMS node 'false', shown as:

$$P_{\perp} \ :< \perp,\ \{\{R_1R_2,\ R_3R_4,\ R_7R_8\},\ \{R_3R_4,\ R_5R_6\}\} > , \tag{2.32}$$

three minimal candidates can be generated as:

$$C_1 = [R_3R_4]\ ,\ \ C_2 = [R_1R_2,\ R_5R_6]\ ,\ \ C_3 = [R_5R_6,\ R_7R_8]\ , \tag{2.33}$$

which means that fuzzy reasoning component R_3R_4 may be defective or fuzzy reasoning components R_1R_2 and R_5R_6 or R_5R_6 and R_7R_8 may both be defective at the same time. Once the faulty fuzzy reasoning components are identified, a modification process takes place to correct such defective components to restore consistency.

2.1.3.4 Candidate Modification

Given a set of defective components, which has been generated by the GDE, the modification procedure is illustrated in Fig. 2.5. In particular, the algorithm randomly picks up a candidate which has the smallest size from the non-empty defective candidate set for the modification. And then, the modifier corrects each defective fuzzy reasoning component in the selected candidate and propagates the modification to all the interpolated rules which are based on the defective component. From here, the algorithm checks inconsistency again, and if all contradictions have been removed, the candidate modification process will terminate with an inconsistency free result. Otherwise, the algorithm will pick up a candidate again for modification until all contradictions have been removed.

As the fuzzy interpolation techniques, including the KH and T-FRI approaches, are developed based on the linear interpolation technique, the relation between an antecedent variable and the corresponding consequent variable in computing the T-FRI can be represented by a linear line (a straight line in Fig. 2.6 between point P_1 and P_2). The modification breaks this straight line into two connected straight line segments, P_1P_3 and P_3P_2, as illustrated in Fig. 2.6. Therefore, the original linear interpolation method has been replaced by a first-order piecewise linear method. As a consequence, for a given antecedent of interpolated rule A^*, the computed conclusion is modified from B^* to $\widehat{B^*}$. In order to achieve such modification, a pair of *correction rates*, denoted as $\underline{c}$ and $\overline{c}$, are introduced in this method. Particularly, $\underline{c}$

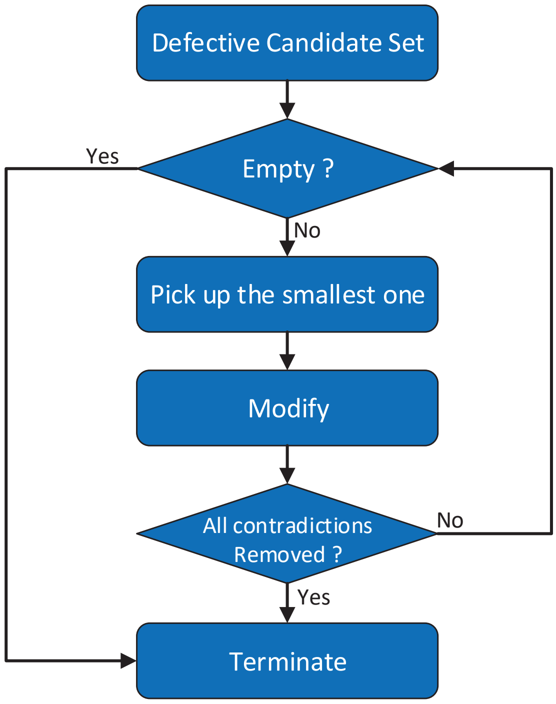


Fig. 2.5 Flowchart of the consistency-restoring algorithm

represents the modification rate of the interpolated rules whose antecedents are on the left side of the modified point (P_3 in Fig. 2.6), such as the antecedents between A_1 and A^* in Fig. 2.6, which is defined as:

$$\underline{c} = \frac{\lambda_{\widehat{B^*}}}{\lambda_{B^*}}\,, \tag{2.34}$$

where λ is the *relative placement factor*, which has been defined in the T-FRI approach (Eq. 2.13), and can be obtained by:

$$\lambda_{\widehat{B^*}} = \frac{D(Rep(B_1),\, Rep(\widehat{B^*}))}{D(Rep(B_1),\, Rep(B_2))}\,, \qquad \lambda_{B^*} = \frac{D(Rep(B_1),\, Rep(B^*))}{D(Rep(B_1),\, Rep(B_2))}\,. \tag{2.35}$$

Similarly, $\overline{c}$ represents the same on the right side, typically those antecedents located from A^* to A_2 in Fig. 2.6, which can be computed by:

$$\overline{c} = \frac{1-\lambda_{\widehat{B^*}}}{1-\lambda_{B^*}}\,. \tag{2.36}$$

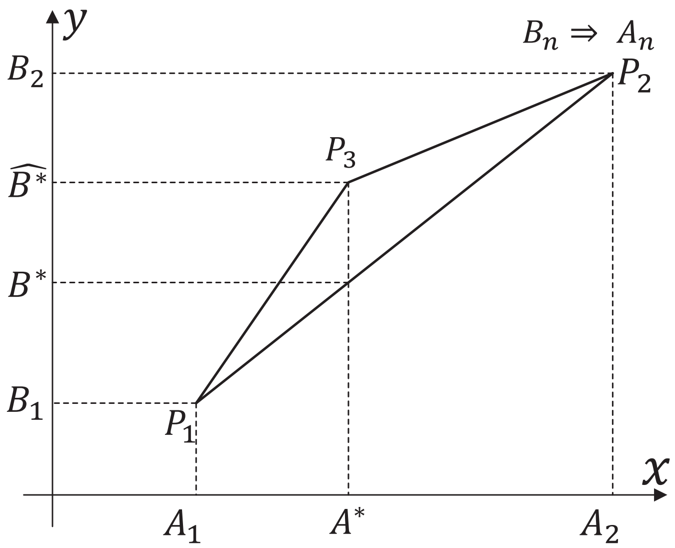


Fig. 2.6 Defective reasoning component modification

From here, for any antecedent $\underline{A^*}$, which is located between A_1 and A^*, its consequent $\underline{B^*}$ is modified to $\widehat{\underline{B^*}}$, whose corresponding relative placement factor $\lambda_{\widehat{\underline{B^*}}}$ is:

$$\lambda_{\widehat{\underline{B^*}}} = \lambda_{\underline{B^*}} \cdot \underline{c} \,. \tag{2.37}$$

Similarly, for any antecedent $\overline{A^*}$, which is located between A^* and A_2, the *corresponding relative placement factor* $\lambda_{\widehat{\overline{B^*}}}$ of its modified consequent $\widehat{\overline{B^*}}$ can be obtained by:

$$\lambda_{\widehat{\overline{B^*}}} = 1 - (1 - \lambda_{\overline{B^*}}) \cdot \overline{c} \,. \tag{2.38}$$

From Eq. 2.35, the modified consequence can than be finally computed.

The adaptive fuzzy interpolation approach was originally proposed to efficiently find and isolate possible faulty interpolated rules, which have caused the inconsistency. This approach assumes that the employed interpolation mechanism is the only cause of contradictions; that is, all given observations and rules are believed to be true and fixed. However, it may not always be the case in the real world situations. Thus, this approach has been further extended to allows the identification and modification of observations and rules [204, 211, 212]. This is achieved by treating both observations and rules as diagnosable and modifiable components in addition to interpolation procedures. The extended work has significantly improved the efficacy of the original adaptive system by exploiting more information during both the diagnosis and modification processes.

2.1.4 Other Approaches

A number of other existing fuzzy interpolation approaches have also been reported in the literature [9, 16, 17, 28, 37, 49, 54, 76, 101, 102, 183, 197]. The approach proposed in [37] developed a weighting-based fuzzy interpolative reasoning approach. Different from the KH and T-FRI approaches, which perform the fuzzy interpolation by only considering two neighbouring rules, this approach allows the multiple fuzzy rules to be used for the conclusion calculation. In this method, a weight factor is allocated to each antecedent variable of fuzzy rules, and the interpolated result is aggregated by using a weighted average mechanism. In the meantime, a genetic algorithm (GA)-based weight learning algorithm is also proposed in this approach to allow the system to automatically learns the optimal weights of the antecedent variables of the fuzzy rules. The advantage of this approach is that it can deal with both polygonal membership functions and bell-shaped membership as well as guaranteeing that the convex and normal fuzzy sets are generated. The same as for [37], the approach of [49] also points out that the rule base may contain irrelevant, redundant or even misleading rule antecedents, which will significantly affect the interpolated results. Compared with the approach in [37], which identifies the importance of each rule antecedent by giving different weight values, the approach of [49] employs harmony search-based feature selection (HSFS) [50] to remove less important antecedent variables, which reduced the complexity of the fuzzy model. The T-FRI approach is then employed to exploit the information analysed by HSFS in an effort to facilitate more effective interpolation.

The approach of [197] is developed based on the concept of the core of the fuzzy set, which is conventionally termed as the support of the fuzzy set. This approach constructs intermediate rules by considering the middle point of the core instead of the representative value, which is introduced by the T-FRI approach. Then, from similarities of fuzzy sets in the antecedent and consequent parts, interpolative reasoning is performed with the constructed intermediate rule by employing the similarity transfer concept. This approach guarantees that the convex and normal fuzzy set can be generated, which overcomes the drawback of the KH approach.

As the drawback of the KH approach may lead to invalid fuzzy sets being generated, a number of methods have been proposed to overcome and avoid such an issue. One of the methods suggested in [76] extends the KH approach by exploiting the notion of the slopes of flanking edges. Fundamentally, the slope of the conclusion can be estimated from the linear combination of the respective (left or right) slopes of the consequence of the neighbouring rules, in a similar manner in which the slopes of the observation can be estimated from the linear combination of the respective (left or right) slopes of the antecedents of the neighbouring rules. Another extension of the KH approach was proposed in [183], which uses the vector representation of fuzzy sets instead of the α-cut sets representation. By employing this representation method, each fuzzy set can be transformed to a vector of its characteristics points. And then, the KH approach is applied to infer the conclusion based on the processed vectors. In the end, the interpolated conclusion is able to be converted back into the original coordinate system. Both approaches extend the original KH approach and guarantee the consequence is a valid normal and convex fuzzy set. However, the approach of [76] is restricted to triangular membership function only, which limits the applicability of this method.

2.2 Fuzzy Rule Base Generation

A rule base, both Mamdani-style and TSK-style, can either be translated from expert knowledge or extracted from data. The rule base led by the knowledge-driven approaches is therefore essentially a representation of the human experts' knowledge in the format of fuzzy rules [142]. Recognising that the expert knowledge may not always be available, data-driven approaches were proposed, which extract fuzzy rules from a set of training data using machine learning and data mining techniques [158]. However, successful data-driven approaches are usually built on a large amount of training data, and they normally lead to dense rule bases to support conventional fuzzy inference systems. The rest of this chapter presents a brief review of both knowledge-driven and data-driven rule base generation approaches.

2.2.1 Knowledge-driven Rule Base Generation

The knowledge-driven rule base generation is a traditional way to build the rule base in which the requisite rules are provided by domain experts and knowledge engineers, such as [126, 132, 170, 187, 188]. The rule base generation processes can be summarised by six steps that are illustrated in Fig. 2.7.

Step 1: The experts have to identify the problems. In order to represent a problem in linguistic rules, the particular issue has to be firstly understood by human experts, thus to considering the possible factors that could contribute to this problem.

Step 2: The critical factors need to be determined by experts. A number of factors may be identified through the first step, which either directly or indirectly contribute to the problem.

Fig. 2.7 The processes of knowledge-driven rule base generation approaches

In order to reduce the complexity of the generated rule base, the most important factors have to be determined by experts, which will be used as the inputs features in the next step.

Step 3: The domain experts have to define the domain for each input and output feature, which can usually be achieved by either human experts or from real data, such as considering the lower and upper boundaries of corresponding data.

Step 4: The domain experts need to partition the each identified input or output domain to a number of areas, in order to represent the problem domain in a set of linguistic variables. In the knowledge-driven approach, it is usually guided by the human experts' knowledge based on the previous experience. Note that, it is not necessary to partition the entire domain equally. Based on the different situations, the input domain can be partitioned to a number of unequal areas.

Step 5: The domain experts then designs the membership functions to represent the linguistic terms. There are various membership functions that can commonly be used for the linguistic terms representation, such as triangle, trapezoid and Gaussian membership functions, each of which is suitable for the different situation. Therefore, the domain experts have to decide on the best representation for each linguistic term.

Step 6: Finally, the experts construct the fuzzy rule by creating the mapping between the input and output, and the final rule base can then be generated by combining all established rules.

In the knowledge-driven approach, the trade-off between the fuzzy model's accuracy and its complexity/simplicity, such as the number of inputs, the number of variables, etc., always has to be considered. Typically, sufficiently increasing the number of rules will generically provide better accuracy. However, it will increase the computation cost. By contrast, a small amount of rules may not be able to obtain a satisfactory level of accuracy. Besides, a drawback of this type of approach is that the experts' knowledge is not always available in the certain areas, which extremely limits the applicability of the knowledge-driven rule base generation approach.

2.2.2 Data-driven Rule Base Generation

Data-driven rule base generation was proposed to minimise the involvement of human expertise, which considers generating a fuzzy rule base from historical data by employing the soft computing techniques, such as neural networks, genetic algorithms, k-Means clustering, etc. [131]. Reviewing those approaches, most of them, in general terms, consist of three general modules, as illustrated in Fig. 2.8.

- *Data pre-processing*: In this stage, raw data is represented in a pre-designed form, which can be fed into soft computing techniques. The processes of data feature extraction and feature selection are also deployed at this stage.

- *Fuzzy rule extraction*: In this stage, the soft computing techniques are adapted to determine the relationship between input space and output space, thus extracting the fuzzy rules, and then, combining all extracted rules to construct the rule base.

- *Membership function optimisation*: In order to increase the system performance, an optimisation process may be adopted to fine-tune the membership functions. As the constructed rule base is able to work with an inference engine to predict the output, it is an optional process in the data-driven rule base generation.

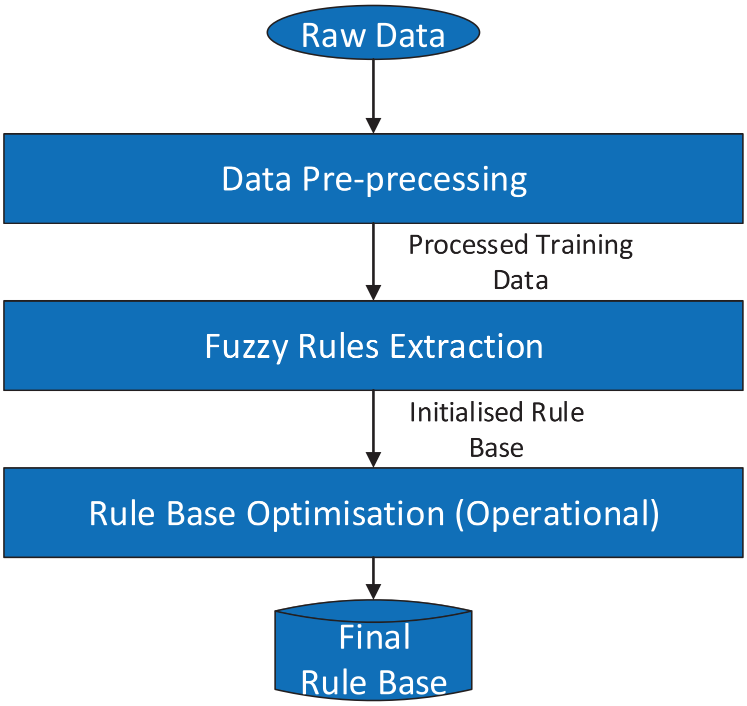


Fig. 2.8 The processes of data-driven rule base generation approaches

In general, based on the different soft computing techniques used, a number of representation methods have been developed for fuzzy rule extraction from numerical data, such as one rule per data instance, one rule per class of data instances and one rule per cluster of data instances. In addition, because of the different forms of rule consequence between Mamdani and TSK fuzzy models, the fuzzy rule extraction methods are slightly different.

2.2.2.1 Mamdani Fuzzy Model

An important milestone in data-driven rule base generation was established in [192], which proposed a non-iterative method to generate complete linguistic rules from data [56]. This approach firstly fuzzy partitions the input and output domains. After that, each numerical data sample is used to represent one fuzzy rule with a special weight, which is allocated based on the degrees in every fuzzy partition of the input and output spaces. And then, all conflicting rules are combined to generate the final rule base. This approach provides a fast and non-iterative way to generate the rule base from data. However, the method of one rule per data instance approaching is very susceptible to noise in the data.

A number of machine learning and data mining methods have been employed to automatically extract fuzzy rules from numerical data, which extremely reduces the influence of noisy data, such as k-NN [42], support vector machine [145], decision tree [56], Naive Bayes [149], random forest [46], artificial neural network [140, 224], genetic algorithm [147, 171] and k-Means clustering [161]. These approaches usually use either classification or clustering techniques to divide the given training dataset into a number of classes or clusters, and then the fuzzy rules can be extracted based on different classes or clusters. Commonly, such approaches require a large number of training data, and they often leads to dense rule bases to support conventional fuzzy inference systems.

Recent development of rule base generation has been reported, with compact sparse rule bases targeted [178]. This approach is developed based on the concept of profile curvature values of different parts of the data pattern. In particular, the method divides the problem domain into a number of components (sub-regions). As the ‘flat’ or ‘straight’ parts of the pattern can be approximated by linear fuzzy interpolation techniques, only the parts with higher curvature values are selected for fuzzy rules extraction. The approach comprises two fundamental processes: raw rule base generation and rule base optimisation, which are illustrated in Fig. 2.9.

Domain partition: Given a training dataset, which is distributed in a 3-dimensional space (2-inputs and signal output), the input domain is equally partitioned into $a \times b$ grid areas, where a and $b \in \mathbb{N}$ indicate the number of partitions on a horizontal axis and a vertical axis, respectively.

Curvature value calculation: The profile curvature values are usually used in geospatial analysis, which represents the steepest downward gradient for a given direction [151]. The profile curvature values are used in this approach to help indicate the importance of each sub-region. Given a sub-region $f(x,y)$, the profile curvature, k_p, is the rate at which a surface slope, S, changes whilst moving in the direction of $grad(f)$, which can be calculated by the directional derivative:

$$D_{(\hat{n})}(F) = \bigtriangledown F \cdot \hat{n} \ . \tag{2.39}$$

The directional derivative refers to the rate at which any given scalar field, $F(x,y)$, is changing as it moves in the direction of some unit vector, $\hat{n}$, such as $\hat{n} = -(\bigtriangledown f/S)$, where S is the slope defined as the magnitude of the gradient vector and is a scalar field:

$$S(x,y) = |\bigtriangledown f| = \sqrt{f_x^2 + f_x^2} \ , \tag{2.40}$$

where $\bigtriangledown f = (f_x, f_y, 0)$ denotes the gradient of this surface, which is a 2D vector that points in the steepest uphill and downhill directions. From here, the profile curvature values, k_p, can be expressed as:

$$k_p = -S^{-1}(\bigtriangledown S \cdot \bigtriangledown f) \ , \tag{2.41}$$

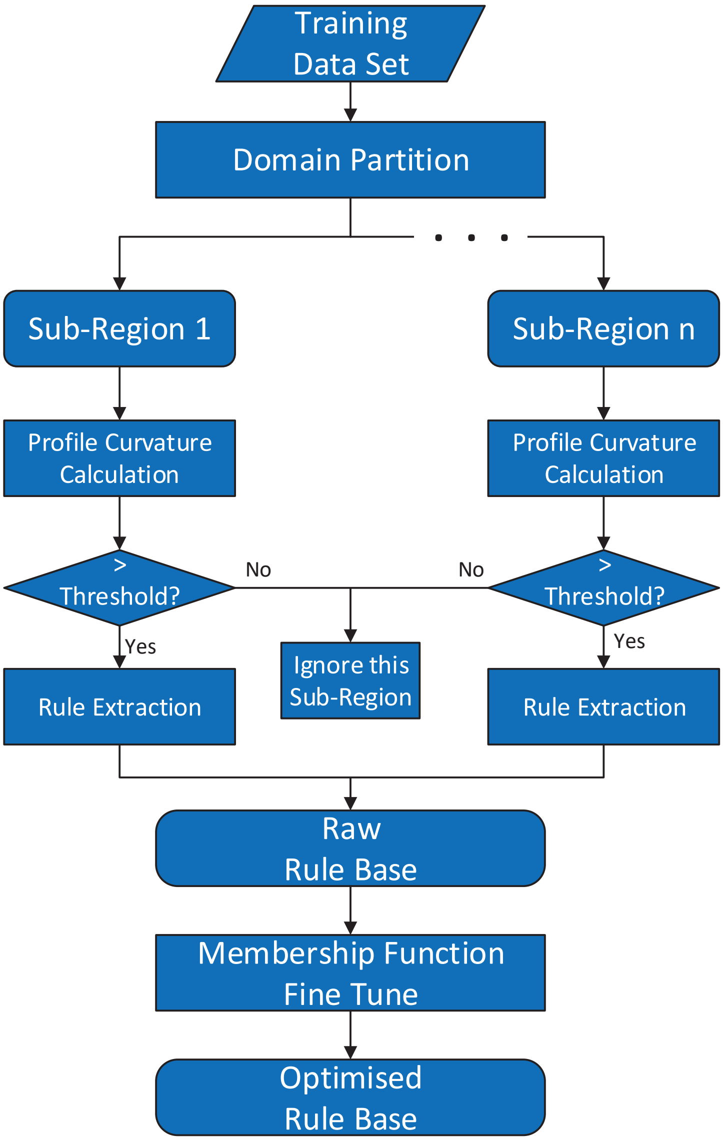


Fig. 2.9 The work flow of the curvature based rule base generation

In order to calculate the overall linearity of a sub-region, eight directions of profile curvature values, which are defined from the centre of the sub-region to the four corners and the central points of the four edges, are defined, such as $k_{p_i}, i = \{1, 2, \cdots, 8\}$. That is, the final profile curvature value takes the maximum value of the eight directional curvature values: $k_p = max(k_{p_i}),\ i = \{1, 2, \cdots, 8\}$.

Rule extraction: Given a curvature threshold θ, if the curvature value of a sub-region is greater than θ, the corresponding sub-region will be selected to form a fuzzy rule. In this approach, each selected sub-region is represented by one fuzzy rule. For simplicity, only isosceles triangular fuzzy sets are employed in this approach, each of which can be precisely represented as $A = (a_1, a_2, a_3)$, where a_2 is the core and (a_1, a_3) is the support of the fuzzy set. In this approach, the core of the fuzzy set is set to the centre of the sub-region, and the support of the fuzzy set is equal to twice the span of the corresponding sub-region. Given a selected sub-region $f(x, y)$, the extracted fuzzy sets are illustrated in Fig. 2.10. The raw rule base can then be constructed from all extracted rules.

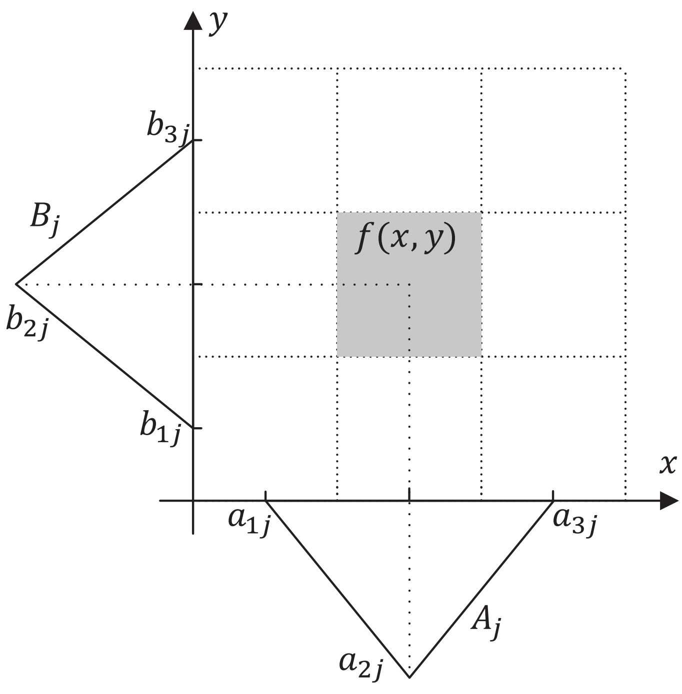


Fig. 2.10 Fuzzy sets extraction

Optimisation: To increase the performance, the genetic algorithm (GA) is adopted for the raw rule base optimisation. In general, a chromosome, denoted as I, in the GA usually represents a possible solution. Therefore, the chromosome is designed in this approach to represent the entire rule base. For simplicity, this approach is only considered to fine-tune the support of each fuzzy set. Given an isosceles triangle fuzzy set A, its support is denoted as $supp(A)$. With this requirement, the length of a chromosome is set to $3 \times n$, as illustrated in Fig. 2.11, where n denotes the number of rules in the rule base. The objective function used

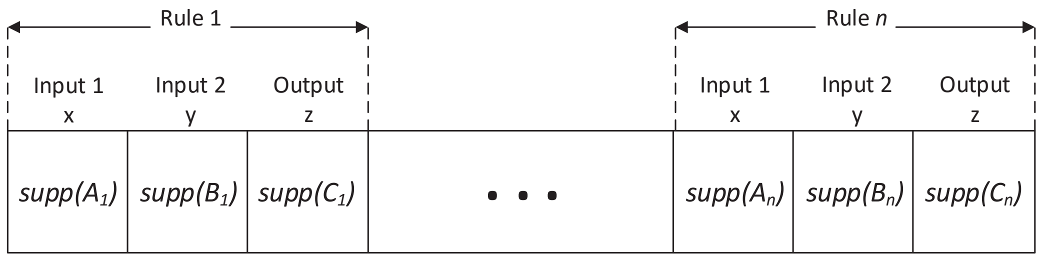


Fig. 2.11 Chromosome encoding

in this work is to minimise the root mean square error (RMSE), which can be calculated as follows:

$$RMSE_i = \sqrt{\frac{\sum_{j=i}^{m} (z_j - \hat{z}_j)^2}{m}} \ , i \in \{1, \cdots, |\mathbb{P}|\} \ , \tag{2.42}$$

where m is the size of the training dataset, $|\mathbb{P}|$ is the size of population, z_j is the labelled defuzzified output value of the j^{th} training data sample and $\hat{z}_j$ represents the defuzzified output that is generated by applying a particular fuzzy interpolation approach over the i^{th} solution of the rule base in the population. Two genetic operators, signal point crossover and mutation, are employed to generate the next generation of population with particularly the 'roulette wheel' selection mechanism for chromosome selection. In addition, the termination condition is set to the maximum number of iterations is reached or the RMSE value is less than a predefined threshold.

The profile curvature-based fuzzy rule base generation approach successfully applied a concept in geography to solve the soft computing problem, which adopts the profile curvature values to identify the importance of each divided sub-region. This approach is not only able to reduce the complexity of the generated rule base but is also able to produce a sparse rule base particularly for fuzzy interpolation techniques. The main advantage of this approach is that it does not require a large number of training datasets. However, as the property of the profile curvature, this approach is only able to deal with a 2-inputs and a signal output Mamdani fuzzy model, which significantly reduces the applicability of this method.

2.2.2.2 TSK Fuzzy Model

The antecedent variables of a TSK rule are represented as fuzzy sets and the consequence is represented by a linear polynomial function, shown as:

$$\begin{aligned} R: \textbf{IF } x_1 \text{ is } A_1 \text{ and } \cdots \text{and } x_m \text{ is } A_m \\ \textbf{THEN } y = f(x_1, \cdots, x_m), \end{aligned} \tag{2.43}$$

where A_i $(i \in \{1,...,m\})$ represents the fuzzy set in the TSK rule antecedent, and $f(x_1,\cdots,x_m)$ is the polynomial function in the rule consequence, which is usually denoted as $\beta_0 + \beta_1 x_1 + \cdots + \beta_m x_m$, where β_j $(j \in \{0,1,\cdots,m\})$ are constant parameters of the polynomial function. Data-driven fuzzy rule extraction approaches then usually first partition the problem domain into multiple regions or rule clusters. Then, each region is represented by a TSK rule. Various clustering algorithms, such as K-Means [125], fuzzy C-Means [55] and mountain clustering [201], can be used to divide the problem domain into sub-regions or rule clusters [27, 33, 155, 158].

Given a rule cluster that contains a set of multi-dimensional data instances, a typical TSK fuzzy rule extraction process is performed in two steps: 1) rule antecedent determinations and 2) consequent polynomial determination [138]. The rule antecedent determination process prescribes a fuzzy set to represent the information of the cluster on each dimension, such as Gaussian membership functions [33, 129], triangle membership function [173] and trapezoid membership function [88]. The consequent polynomial can usually be determined by employing linear regression techniques [138, 158], such as linear least squares regression [144], maximum likelihood estimation [175] and principal component regression [216]. Once each rule cluster has been expressed by a TSK rule, the TSK rule base can be assembled by combining all the extracted rules.

Note that a 0-order TSK fuzzy model is required if only symbolic labels are included in the output of the dataset [90]. In this case, the step of consequent polynomial determination is usually omitted. Instead, discrete crisp numbers are typically used in representing the symbolic output values. Accordingly, the rule base generation process is different by firstly dividing the labelled dataset into multiple sub-datasets, each sharing the same label. Then, a clustering algorithm is applied to each sub-dataset to generate rule clusters. Finally, each rule cluster is represented as the antecedents of a rule, with an integer number used as the 0-order consequence representing the label.

A number of data-driven TSK rule base generation methods have been proposed in the literature. Most of them were developed based on the clustering techniques as introduced above, which use the number of clusters to decide the number of rules. Therefore, the system accuracy and the number of rules (the number of clusters) have to be simultaneously considered. Also, unlike the Mamdani fuzzy model, there is no fuzzy interpolation technique available to perform the TSK inference when a sparse rule base is provided. Therefore, common to all existing TSK rule base generation methods, all of them are targeted on a dense rule base, in which the entire input domain has been fully covered, although most of them are able to deal with an imbalanced dataset.

2.3 Summary

This chapter has presented a literature review of typical fuzzy rule interpolation methods, and the fuzzy rule base generation approaches. Notably, the first part of the chapter has mainly reviewed the basic concepts of the fuzzy interpolation techniques. Briefly, the fuzzy interpolation techniques are not only able to perform the fuzzy inference on the sparse rule bases but are also able to reduce the system complexity by omitting the rules that may be approximated by their neighbouring ones. Fundamentally, according to the implementation methods, the existing fuzzy interpolation approaches can be grouped into two categories, the resolution principle-based approaches and the analogy-based approaches. In particular, the resolution principle-based approaches are able to directly interpolate a consequence by considering the given observation and the existing rules. The analogy-based approaches first generate an intermediate rule such that the antecedent of the intermediate rule is as ‘close’ to the given observation as possible. And then, a consequence is able to be interpolated by considering the shape distinguishability between the given observation and the antecedent of the generated intermediate rule. The KH approach, the T-FRI approach and the adaptive fuzzy interpolation approach are taken as the representatives to illustrate the implementations of fuzzy interpolation techniques in the first part of this chapter.

The second part of the chapter has reviewed the methodologies of the fuzzy rule base generation. Generally speaking, a rule base can either be translated from expert knowledge or data. The knowledge-driven approaches are the traditional way to build a rule base, which requires the particular problem to be understood by domain experts. The data-driven approaches were proposed to minimise the involvement of human expertise. In the data-driven rule base generation approaches, the fuzzy rules are extracted from numerical data. In particular, the procedures of data-driven methods for different fuzzy models, the Mamdani fuzzy model and the TSK fuzzy model, have also been reviewed in the second part of this chapter.

Chapter 3

TSK+ Fuzzy Inference System

A rule base covering the entire input domain is required for the conventional Mamdani inference and the Takagi–Sugeno–Kang (TSK) inference. Fuzzy interpolation enhances conventional fuzzy rule inference systems by allowing the use of sparse rule bases by which certain inputs are not covered. Given that most of the existing fuzzy interpolation approaches were developed to support the Mamdani inference, this chapter presents a fuzzy interpolation approach that extends the TSK inference. The chapter also proposes a data-driven rule base generation method to support the extended TSK inference system. The proposed system enhances the conventional TSK inference in two ways: (1) workable with incomplete or unevenly distributed datasets or incomplete expert knowledge that entails only a sparse rule base, and (2) simplifying complex fuzzy inference systems by using more compact rule bases for complex systems without the sacrificing of system performance. The experimentation shows that the proposed method overall outperforms the existing approaches with the utilisation of smaller rule bases.

The structure of the rest of this chapter is organised as follows. Chapter 3.1 introduces the theoretical underpinnings of the TSK fuzzy inference model. Chapter 3.2 presents the extended TSK system, and Chapter 3.3 discusses the proposed rule base generation approach. Chapter 3.4 details the experimentation for demonstration and validation. Chapter 3.5 summarises the chapter.

3.1 Conventional TSK Inference

Suppose a TSK-style fuzzy rule base comprises of n rules, each with m antecedents:

$$
\begin{aligned}
&R_1 : \textbf{IF } x_1 \text{ is } A_{11} \text{ and } \cdots \text{and } x_m \text{ is } A_{m1}\\
&\qquad \textbf{THEN } y = f_1(x_1, \cdots, x_m)\\
&\qquad\qquad\qquad = \beta_{01} + \beta_{11}x_1 + \cdots + \beta_{m1}x_m,\\
&\cdots\cdots\\
&R_n : \textbf{IF } x_1 \text{ is } A_{1n} \text{ and } \cdots \text{and } x_m \text{ is } A_{mn}\\
&\qquad \textbf{THEN } y = f_n(x_1, \cdots, x_m)\\
&\qquad\qquad\qquad = \beta_{0n} + \beta_{1n}x_1 + \cdots + \beta_{mn}x_m,
\end{aligned}
\tag{3.1}
$$

where β_{0r} and β_{sr}, ($r \in \{1,2,\cdots,n\}$ and $s \in \{1,2,\cdots, m\}$) are constant parameters of the linear functions of rule consequences. The consequent polynomials deteriorate to constant numbers β_{0r} when the outputs are discrete crisp numbers (to represent symbolic values). Given an input vector $(A_1^*,\cdots,A_m^*)$, the TSK engine performs inference in the following steps:

1: Determine the firing strength of each rule R_r ($r \in \{1,2,\cdots,n\}$) by integrating the similarity degrees between its antecedents and the given inputs:

$$\alpha_r = S(A_1^*,A_{1r}) \wedge \cdots \wedge S(A_m^*,A_{mr}), \tag{3.2}$$

where $\wedge$ is a t-norm usually implemented as a minimum operator, and $S(A_s^*,A_{sr})$ ($s \in \{1,2,\cdots,m\}$) is the similarity degree between fuzzy sets A_s^* and A_{sr}:

$$S(A_s^*,A_{sr}) = max\{min\{\mu_{A_s^*}(x), \mu_{A_{sr}}(x)\}\}, \tag{3.3}$$

where $\mu_{A_s^*}(x)$ and $\mu_{A_{sr}}(x)$ are the degrees of membership for a given x in the domains of fuzzy sets A_s^* and A_{sr}, respectively.

2: Calculate the sub-output led by each rule R_r based on the given observation $(A_1^*,\cdots,A_m^*)$:

$$
\begin{aligned}
&f_r(x_1^*,\cdots,x_m^*)\\
&= \beta_{0r} + \beta_{1r}Rep(A_1^*) + \cdots + \beta_{mr}Rep(A_m^*),
\end{aligned}
\tag{3.4}
$$

where $Rep(A_s^*)$ is the representative value or defuzzified value of fuzzy set A_s^*, which is often calculated as the centre of the area of the fuzzy set membership function.

3: Generate the final output by integrating all the sub-outputs from all the rules:

$$y = \frac{\sum_{r=1}^{n} \alpha_r f_r(x_1^*, \cdots, x_m^*)}{\sum_{r=1}^{n} \alpha_r}. \tag{3.5}$$

It is clear from Eq. 3.3 that the firing strength will be 0 if a given input vector does not overlap with any rule antecedent. In this case, no rule will be fired and the conventional TSK approach will fail.

3.2 TSK+ Inference

The conventional TSK fuzzy inference system is extended in this chapter by allowing the interpolation and extrapolation of inference results. The extended system is thus workable with sparse, dense and imbalanced rule bases, which is termed as the TSK+ inference system. It has been proven in the literature that different types of membership functions do not pose a significant difference in inference results if they are properly fine-tuned [26]. Based on this, only triangular membership functions are used in this work for computational efficiency. Note that the Gaussian membership function will not be considered in this chapter, because the proposed similarity measure is not suitable for the Gaussian membership functions.

3.2.1 The Similarity Measure

Conventional TSK will fail if a given input does not overlap any rule antecedent in the rule base. This can be addressed using fuzzy interpolation such that the inference consequence can be approximated from the neighbouring rules of the given input. In order to enable this, the measure of firing strength used in the conventional TSK inference is modified based on a revised similarity measure proposed in [36]. In particular, the similarity measure proposed in [36] is not sensitive to distance in addition to membership functions. This similarity measure is further extended in this chapter such that its sensitivity to distance is flexible and configurable to support the development of the TSK+ inference engine.

Given two triangular fuzzy sets $A = (a_1, a_2, a_3)$ and $A' = (a'_1, a'_2, a'_3)$ in a normalised variable domain, their similarity degree $S(A, A^{'})$ can be calculated as in [36]:

$$S(A, A') = \left(1 - \frac{\sum_{i=1}^{3} |a_i - a'_i|}{3}\right). \tag{3.6}$$

Eq. 3.6 is extended in this work by introducing a configurable parameter as:

$$S(A, A') = \left(1 - \frac{\sum_{i=1}^{3} |a_i - a'_i|}{3}\right) \cdot d, \tag{3.7}$$

where d, termed as the *distance factor*, is a function of the distance between the two concerned fuzzy sets:

$$d = \begin{cases} 1 & ; \quad \begin{array}{c} a_1 = a_2 = a_3 \\ \& \ a_1^{'} = a_2^{'} = a_3^{'} \end{array} \\ 1 - \frac{1}{1 + e^{(-s \cdot \|A, A'\| + 5)}} & ; \quad \text{otherwise}, \end{cases} \tag{3.8}$$

where $\|A, A'\|$ represents the distance between two fuzzy sets, usually defined as the Euclidean distance of their representative values, and s ($s > 0$) is an adjustable sensitivity factor. The smaller value of s leads to a similarity degree which is more sensitive to the distance between the two fuzzy sets. The number 5 in the equation is a fixed number which is determined in order to make sure the distance factor can be maximised when the distance between two fuzzy sets is 0. According to Eq. 3.8, the *distance factor* is not considered when fuzzy sets A and A' are both crisp. This is because the shapes of the fuzzy sets need to considered by the representative values as contributing elements of the *distance factor* when the objects are fuzzy sets, but there is no point in considering this element if the objects are crisp numbers [84].

The modified similarity measure $S(A, A')$ between fuzzy sets A and A' has the following properties:

1. larger value of $S(A, A')$ represents higher similarity degree between fuzzy sets A and A';

2. $S(A, A') = 1$ if and only if fuzzy sets A and A' are identical;

3. $S(A,A') > 0$ unless ($a_1 = a_2 = a_3 = 0$ and $a'_1 = a'_2 = a'_3 = 1$) or ($a_1 = a_2 = a_3 = 1$ and $a'_1 = a'_2 = a'_3 = 0$).

3.2.2 Extending TSK to Work with Sparse Rule Bases

Given a rule base as specified in Eq. 3.1 and an input vector $(A^*_1, \cdots, A^*_m)$, the TSK+ performs inferences using the same steps as those detailed in Chapter 3.1, except that Eq. 3.3 is replaced by Eq. 3.7. According to the third property of the modified similarity measure discussed above, $S(A^*_s, A_{sr}) > 0$ unless A^*_s and A_{sr} take boundary crisp values 0 and 1. This means the firing strength of any rule R_r is always greater than 0, i.e., $\alpha_r > 0$, except for the special case when only boundary crisp values are involved. As a result, every rule in the rule base contributes to the final inference result to a certain degree. Therefore, even if the given observation does not overlap with any rule antecedent in the rule base, certain inference results can still be generated, which significantly improves the applicability of the conventional TSK inference system.

3.3 Rule Base Generation

A data-driven TSK-style rule base generation approach for the proposed TSK+ inference engine is presented in this chapter, which is outlined in Fig. 3.1. Given a dataset $\mathbb{T}$ which might be sparse, unevenly distributed, or dense, the system firstly groups the data instances into clusters using certain clustering algorithms. Then, each cluster is expressed as a TSK rule by employing linear regression. From this, an initial rule base is generated by combining all the extracted rules. Finally, the initialised rule base is optimised by applying the genetic algorithm (GA), which fine-tunes the membership functions of fuzzy sets in the rule base.

Fig. 3.1 TSK+ rule base generation

3.3.1 Rule Base Initialisation

Centroid-based clustering algorithms are traditionally employed in TSK fuzzy modelling to group similar objects together for rule extraction [33], which is also the case in this work. However, differing from existing TSK-style rule base generation approaches, the proposed system is workable with dense, sparse and unevenly distributed datasets. Although a number of clustering approaches have been proposed in the literature to deal with sparse and unevenly distributed datasets, such as [69, 118, 165], they suffer from exponential computational complexity, and have no performance guarantees [103, 222]. Therefore, a two-level clustering scheme is applied in this work. The first level of clustering divides the given (dense/sparse) dataset into multiple sub-datasets using a sparse K-Means clustering algorithm, which performs well on both sparse and dense datasets [194]. Based on the feature of the sparse K-Means clustering, those divided sub-datasets can be considered as the dense datasets. The second level of clustering is applied on each obtained dense sub-dataset to generate rule clusters for TSK fuzzy rule extraction by employing the standard K-Means clustering algorithm [125]. Note that the number of clusters has to be pre-defined for both the sparse K-Means and the standard K-Means, which is discussed first below.

3.3.1.1 Number of Clusters Determination

A number of approaches have been proposed in the literature to determine the value of k, such as the Elbow method [13, 182], the Information Criterion Approach [20, 200], the Cross-validation [105, 98, 172], the Bayesian Information Criterion (BIC)-based approach [152], and the Silhouette-based approach [44, 119, 160]. In particular, the Elbow method is faster and effective [13], and this approach is therefore employed in this work. This method determines the number of clusters based on the criteria that adding another cluster does not lead to a much better modelling result based on the given objective function [13]. In particular, the implementation of the Elbow method is to run K-Means clustering on the dataset for a range of values of $k(k \in \{1,2,...,n\}, n \in \mathbb{N})$. For each value of k, to calculate the sum of squared errors (SSE_k), thus to obtain the performance improvement $PI_k(k \geq 2)$, such as $PI_k = SSE_k - SSE_{k-1}$. And then, to plot a line chart of the improved performance for each value of k. The line chart should looks like an arm, and the "elbow" on the arm is the determined value of k. For instance, for a given problem, the relationship between performance improvement and the value of k is shown in Fig. 3.2. The value of k in this case can be determined as 4, which is the obvious turning point (or the elbow point of the arm). Note that the performance improvement may fluctuate after the obvious turning point. In this case, the first obvious turning point will be used as the determined number of k in order to minimise the generated number of rules.

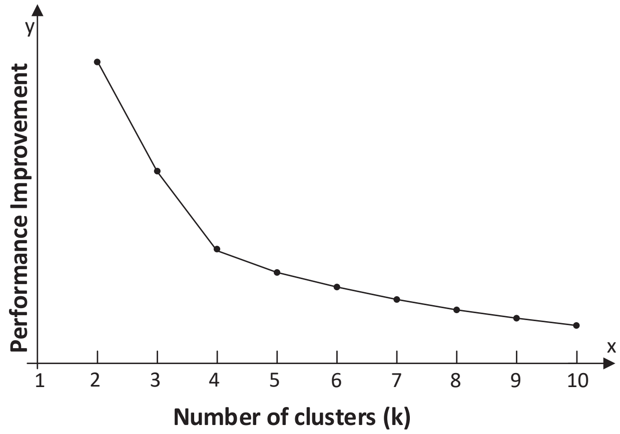


Fig. 3.2 Determination of k using the Elbow method

3.3.1.2 Dense Sub-dataset Generation

The sparse K-Means is an extension of the standard K-Means for handling sparse datasets [194]. Assuming k clusters are required, the sparse K-Means also starts with the initialisation of k centroids (usually randomly). This is followed by the assignment of data instances to centroids and the updating of centroids based on the assignments, which are iterated until there is no change in the assignments. Different from the standard K-Means, which assigns objects with the goal of minimising the within-cluster sum of squares error (SSE), the sparse K-Means assigns objects by maximising the between-cluster sum of squares error (BCSS), which is defined as:

$$BCSS = \sum_{q=1}^{m+1} (\sum_{t=1}^{p} (x_{tq} - \mu_q)^2 - SSE), \tag{3.9}$$

where p is the total numbers of data instances in the given dataset, m is the number of input features in the given dataset, μ_q is the mean of all the elements on the q^{th} feature, and x_{tq} is the q^{th} feature of the t^{th} data point in the given dataset.

The within-cluster sum of squares error (SSE) is defined as:

$$SSE = \sum_{j=1}^{k} \sum_{t=1}^{p_j} (\parallel x_{jt} - v_j \parallel)^2, \tag{3.10}$$

where k is the number of clusters determined by the Elbow approach, p_j is the number of data instances in the j^{th} cluster, x_{jt} is the t^{th} data point in the j^{th} cluster, v_j is the j^{th} cluster centre, and $\parallel x_{jt} - v_j \parallel$ is the Euclidean distance between x_{jt} and v_j. Note that if the labels in a given dataset are symbolic values, only 0-order TSK rules are required and thus Eq. 3.9 becomes:

$$BCSS = \sum_{q=1}^{m} (\sum_{t=1}^{p} (x_{tq} - \mu_q)^2 - SSE). \tag{3.11}$$

Note that a density-based K-Means was proposed in [61], which is also able to deal with sparse datasets. However, this approach is not applied in this work for a number of reasons. It is difficult to determine the right values of the two required parameters, including the distance between objects in one cluster, and the number of neighbouring objects in a cluster. This has been discussed and partially addressed in [156], but the approach is applicable to spatial-temporal data only.

3.3.1.3 Rule Cluster Generation

Once the given training dataset $\mathbb{T}$ has been divided into k dense sub-datasets, K-Means is employed to each determined sub-dataset T_i ($1 \leq i \leq k$) to generate rule clusters, each representing a rule. Assume that k_i clusters are required for a sub-dataset T_i. K-Means is initialised by k_i random cluster centroids. It then assigns every data instance to one cluster by minimising the SSE:

$$SSE = \sum_{j=1}^{k_i} \sum_{t=1}^{p_t^i} (\parallel x_{jt}^i - v_j^i \parallel)^2, \tag{3.12}$$

where p_t^i is the number of data points in the j^{th} cluster of the sub-dataset T_i, x_{jt}^i is the t^{th} data point in the j^{th} cluster in the sub-dataset T_i, v_j^i is the centre of the j^{th} cluster in the sub-dataset T_i, and $\parallel x_{jt}^i - v_j^i \parallel$ is the Euclidean distance between x_{jt}^i and v_j^i. Once all the data instances are assigned, the algorithm updates the cluster centroids accordingly to the newly assigned members. These two steps are iterated until there is no change in object assignments. After the K-Means is applied, the given training dataset $\mathbb{T}$ is divided into $n = \sum_{j=1}^{k} k_i$ clusters. For simplicity, the generated rule clusters are jointly represented as $\{RC_1, RC_2, \cdots, RC_n\}$.

3.3.1.4 Fuzzy Rule Extraction

Each determined cluster from the above steps is utilised to form one TSK fuzzy rule. A number of approaches have been proposed to use a Gaussian membership function to represent a cluster, such as [158]. However, given the fact that most of the real-world data are not normally distributed, the cluster centroid is not usually identical with to the centre of the Gaussian membership function, and thus Gaussian membership functions may not be able to accurately represent the distribution of the calculated clusters. In order to prevent this and also keep computational efficiency, triangular membership functions are utilised in this work.

Suppose that a dataset has m input features and a single output feature. Given a rule cluster RC_r $(1 \leq r \leq n)$, a TSK fuzzy rule R_r can be extracted from the cluster as follows:

$$\begin{aligned} R_r : \textbf{IF } x_1 \text{ is } A_{1r} \text{ and } \cdots \text{ and } x_m \ is \ A_{mr} \\ \textbf{THEN } y = f_r(x_1, \cdots, x_m). \end{aligned} \tag{3.13}$$

Without loss generality, take the s^{th} dimension ($1 \leq s \leq m$) of rule cluster RC_r as an example, denoted as RC_r^s. Suppose that RC_r^s has p_r elements, i.e., $RC_r^s = \{x_s^1, x_s^2, \cdots, x_s^{p_r}\}$. As only triangular fuzzy sets are used in this work, fuzzy set A_{sr} can be precisely represented as $(a_{sr}^1, a_{sr}^2, a_{sr}^3)$. The core of the triangular fuzzy set is set as the cluster centroid, that is $a_{sr}^2 = \sum_{q=1}^{p_r} x_s^q / p_r$; and the support of the fuzzy set is set as the span of the cluster, i.e., $(a_{sr}^1, a_{sr}^3) = (min\{x_s^1, x_s^2, \cdots, x_s^{p_r}\}, max\{x_s^1, x_s^2, \cdots, x_s^{p_r}\})$.

First-order polynomials are typically used as the consequences of TSK fuzzy rules. That is, $y = \beta_{0r} + \beta_{1r}x_1 + \cdots + \beta_{mr}x_m$, where the parameters β_{0r} and β_{sr}, $s \in \{1, 2, ..., m\}$ are estimated using a linear regression approach. Locally weighted regression (LWR) is particularly adopted in this work, due to its ability to generate an independent model that is only related to the given cluster of data in the training dataset ([138, 158]). The rule consequence will deteriorate to 0-order, if the values in the output dimension are discrete integer numbers. From this, the raw base is initialised by combining all the extracted rules, which is of the form of Eq. 3.1.

3.3.2 Optimisation

The generated raw rule base is optimised in this chapter by fine-tuning the membership functions using the general optimisation searching algorithm, the genetic algorithm (GA). GA has been successfully utilised in rule base optimisation, such as in [133] and [178]. Briefly, GA is an adaptive heuristic search algorithm for solving both constrained and unconstrained optimisation problems based on evolutionary ideas of the natural selection process that mimics biological evolution. The algorithm firstly initialises the population with random individuals. It then selects a number of individuals for reproduction by applying the genetic operators. The offspring and some of the selected existing individuals jointly form the next generation. The algorithm repeats this process until a satisfactory solution is generated or a maximum number of generations has been reached.

3.3.2.1 Problem Representation

Assume that an initialised TSK rule base is comprised of n rules, as expressed in Eq. 3.1. A chromosome or individual, denoted as I, in the GA is used to represent a potential solution, which is designed to represent the rule base in this proposed system, as illustrated in Fig. 3.3.

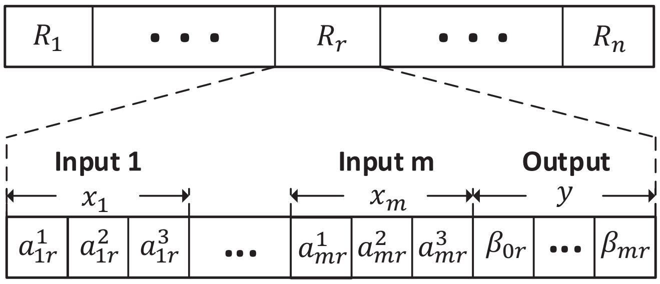


Fig. 3.3 Chromosome encoding

3.3.2.2 Population Initialisation

The initial population $\mathbb{P} = \{I_1, I_2, \cdots, I_{|\mathbb{P}|}\}$ is formed by taking the initialised rule base and its random variations. In order to guarantee all the fuzzy sets are valid and convex, the constraints $a^1_{sr} < a^2_{sr} < a^3_{sr}$ is applied to the genes representing each fuzzy set. The size of the population $|\mathbb{P}|$ is a problem-specific adjustable parameter, typically ranging from tens to thousands, with 20 to 30 being used most commonly [134].

3.3.2.3 Objective Function

An objective function is used in the GA to determine the quality of individuals. The objective function in this work is defined as the root mean square error (RMSE). Given a training dataset $\mathbb{T}$ and an individual I_i, $1 \leq i \leq |\mathbb{P}|$, the RMSE value can be calculated as:

$$RMSE_i = \sqrt{\frac{\sum_{j=1}^{|\mathbb{T}|} (z_j - \hat{z}_j)^2}{|\mathbb{T}|}}, \tag{3.14}$$

where $|\mathbb{T}|$ is the size of the given training dataset, z_j is the label of the j^{th} training data instance, and $\hat{z}_j$ represents the output value led by the proposed TSK+ inference approach. The individual with the smallest value of RMSE represents the fittest solution in the population.

3.3.2.4 Selection

A number of individuals need to be selected for reproduction, which is implemented in this work by the fitness proportionate selection method, also known as the roulette wheel selection. Assuming that f_i is the fitness of individual I_i in the current population $\mathbb{P}$, its probability of being selected to generate the next generation is:

$$p(I_i) = \frac{f_i}{\sum_{j=1}^{|\mathbb{P}|} f_j}, \tag{3.15}$$

where $|\mathbb{P}|$ is the size of the population. The fitness f_i of an individual I_i in the proposed system was determined by adopting the linear-ranking algorithm ([6]) that can be calculated by:

$$f_i = 2 - max + \frac{2(max-1)(r_i-1)}{|\mathbb{P}|}, \tag{3.16}$$

where r_i is the ranking position of individual I_i in the ordered population $\mathbb{P}$, and max is the bias or selective pressure towards the fittest individuals in the population.

3.3.2.5 Reproduction

Once a number of parents are selected, they then breed some individuals for the next generation using the genetic operators, crossover and mutation, as shown in Fig. 3.4. In particular, crossover swaps contiguous parts of the genes of two individuals. In this work, the single-point crossover approach is adopted, which swaps all data beyond this index point between the two parent chromosomes to generate two children. Note that the crossover point can only be between those genes which are employed to represent two different fuzzy sets, such that all the fuzzy sets are valid and convex all the time during the reproduction process.

The second genetic operator mutation is used to maintain genetic diversity from one generation of an individual to the next, which is analogous to a biological mutation. Mutation alters one gene values in a chromosome from its initial state. A pre-defined mutation rate is used to control the percentage of occurrence of mutations. In this work, in order to make sure the resultant fuzzy sets are valid and convex, the constraint $a_{sr}^1 \geq a_{sr}^2 \geq a_{sr}^3$ is applied to the genes representing each fuzzy set during the mutation operation. The newly bred individuals and some of the best one in the current generation $\mathbb{P}$ jointly form the next generation of the population ($\mathbb{P}'$).

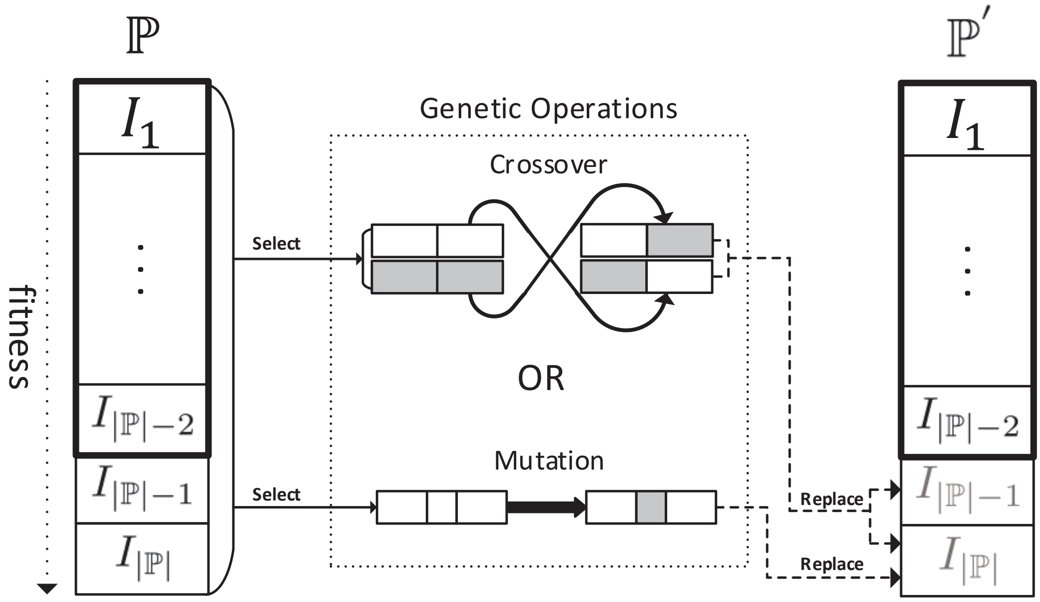


Fig. 3.4 Procedure of reproduction with only one crossover or mutation operation per generation

3.3.2.6 Iteration and Termination

The selection and reproduction processes are iterated until the pre-defined maximum number of iterations is reached, or the value of the objective function regarding an individual is less than a pre-specified threshold. When the termination condition is satisfied, the fittest individual in the current population is the optimal solution.

3.4 Experimentation

Two non-linear mathematical models are employed in this chapter to validate and evaluate the proposed TSK+ fuzzy inference approach.

3.4.1 Experiment 1

A 3-dimensional non-linear function has been used as a benchmark by a number of projects, including recent ones, such as [11, 178] and [111], which is re-considered in this section for a comparative study. The problem is given below:

$$f(x_1,x_2) = sin\left(\frac{x_1}{\pi}\right) \cdot sin\left(\frac{x_2}{\pi}\right), \tag{3.17}$$

which takes two inputs, x_1 ($x_1 \in [10,30]$) and x_2 ($x_2 \in [10,30]$), and produces a single output $y = f(x_1,x_2)$ ($y \in [-1,1]$), as illustrated in Fig. 3.5.

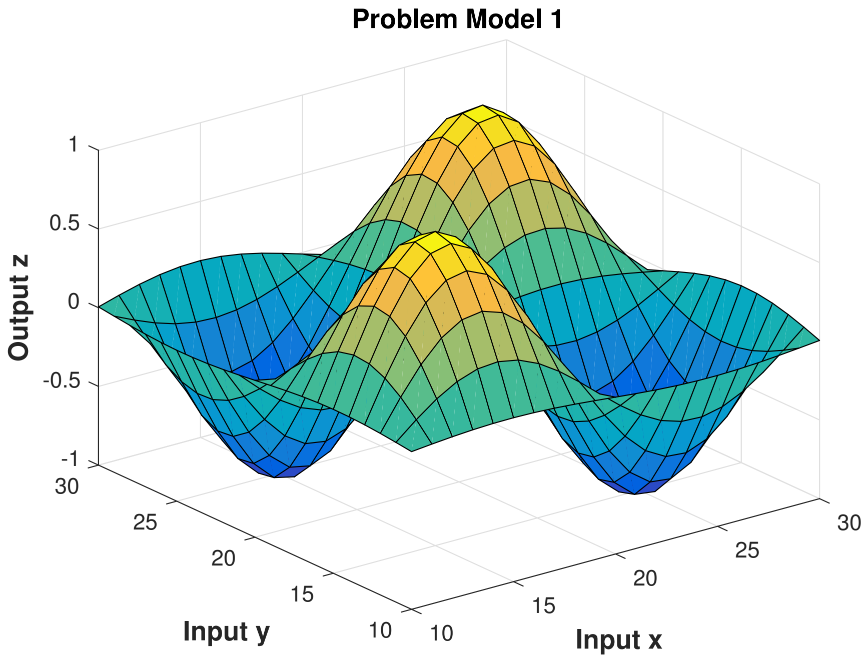


Fig. 3.5 Surface view of Eq. 3.17 in Experiment 1

3.4.1.1 Rule Base Initialisation

In order to demonstrate the proposed TSK+ rule base generation approach, a sparse training dataset $\mathbb{T}$ was manually generated from Eq. 3.17. The dataset was composed of 300 data points sparsely distributed within the $[10,30] \times [10,30]$ domain covering 57% of the input domain. The distribution of this training dataset is illustrated in a 2-dimensional plane in Fig. 3.6. The key steps of the TSK rule base initialisation using the training dataset are summarised below.

Step 1 *Dense sub-datasets generation*: The sparse training dataset $\mathbb{T}$ was first divided into a number of dense sub-datasets by applying the sparse K-Means clustering algorithm. In particular, the number of sub-datasets was determined by the Elbow method, as discussed in the Chapter 3.3.1. The performance improvements (PIs) against the incremented number clusters are listed in Table 3.1 and shown in Fig. 3.7. It is clear from the figure that the performance improvement decreased rapidly when k increased from 2 to 6, before it flattened out after 6. Although there is a fluctuation after $k = 6$, 6 was also taken as the number of clusters in order to minimise the generated number of rules. The application of the sparse K-Means led to six dense sub-datasets, as demonstrated in Fig. 3.8.

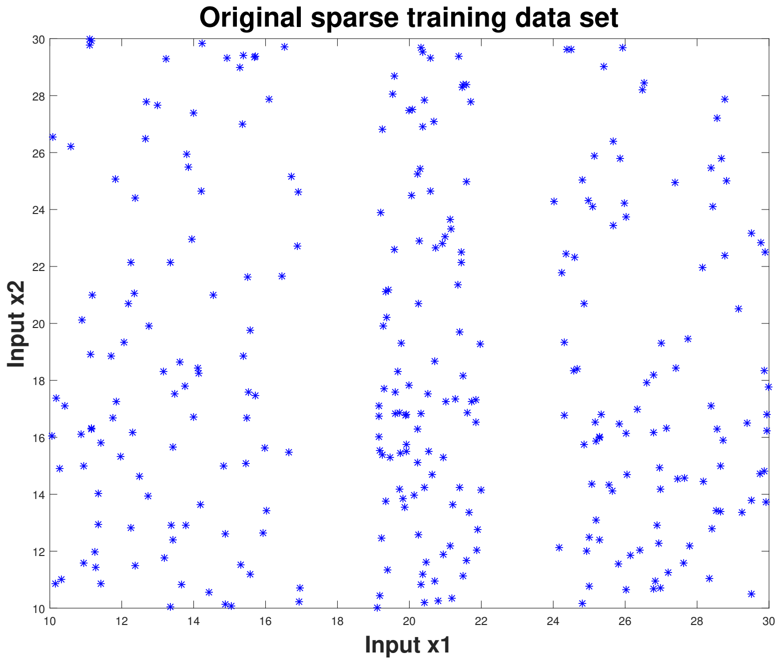


Fig. 3.6 Training dataset distribution in Experiment 1

Table 3.1 SSE and performance improvement

No. of k	SSE	PI
1	2961.91	
2	2248.98	712.93
3	1763.74	485.24
4	1478.32	285.42
5	1286.41	191.91
6	1180.48	105.93
7	1079.48	101.00
8	997.626	81.85
9	946.238	51.39
10	880.43	65.81
11	840.865	39.57
12	808.51	32.36
13	767.61	40.90
14	709.997	57.61
15	689.301	20.70

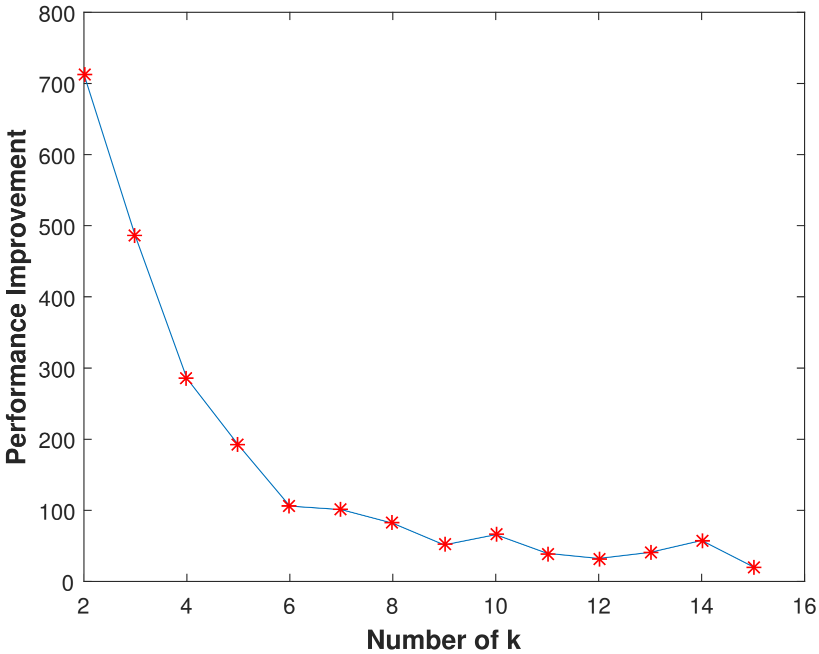


Fig. 3.7 Performance improvement against incremented k. The performance improvement decreased rapidly when k increased from 2 to 6, before it flattened out after 6. Although there is a fluctuation after $k = 6$, 6 was also taken as the number of clusters in order to minimise the generated number of rules.

Step 2 *Rule cluster generation*: Once the sparse training dataset $\mathbb{T}$ was divided into 6 dense sub-datasets ($T_1, T_2, ..., T_6$), the standard K-Means clustering algorithm was employed on each determined sub-dataset to group similar data points into rule clusters. The application of the Elbow approach led to a set of SSEs and PIs, as listed in Table 3.2, which in turn determined the value of k for each sub-dataset, as listed in Table 3.3. According to Table 3.3, the total number of rules should be 28 ($5+5+3+4+6+5 = 28$).

Table 3.2 SSE and PI for each sub-dataset ($k = \{1, 2, \cdots, 10\}$)

No. of	T_1		T_2		T_3		T_4		T_5		T_6	
k	**SSE**	**PI**	**SSE**	**PI**	**SSE**	**PI**	**SSE**	**PI**	**SSE**	**PI**	**SSE**	**PI**
1	159.9		271.6		128.3		222.28		133.98		259.65	
2	109.42	50.48	187.65	83.95	68.61	59.69	140.16	82.12	96.06	37.92	187.95	71.70
3	81.44	27.98	149.88	37.78	56.66	11.95	114.23	25.93	66.42	29.64	147.02	40.93
4	66.85	14.86	126.35	23.53	46.16	10.50	99.48	14.75	52.46	13.96	123.73	23.29
5	56.45	10.13	109.73	16.62	39.87	6.29	85.06	14.42	42.79	9.67	109.39	14.34
6	48.04	8.41	94.14	15.59	33.04	6.83	77.22	7.84	37.69	5.10	95.25	14.104
7	42.01	6.03	83.67	10.47	28.33	4.71	69.90	7.32	33.52	4.17	85.26	9.99
8	36.22	5.79	77.63	6.04	26.26	2.07	61.87	8.03	29.18	4.34	80.06	5.2
9	32.4	3.82	74.63	8.00	24.03	2.23	56.17	5.70	25.01	4.17	74.62	5.44
10	28.77	3.63	67.11	7.52	22.69	1.34	52.39	3.78	24.29	0.72	68.56	6.06

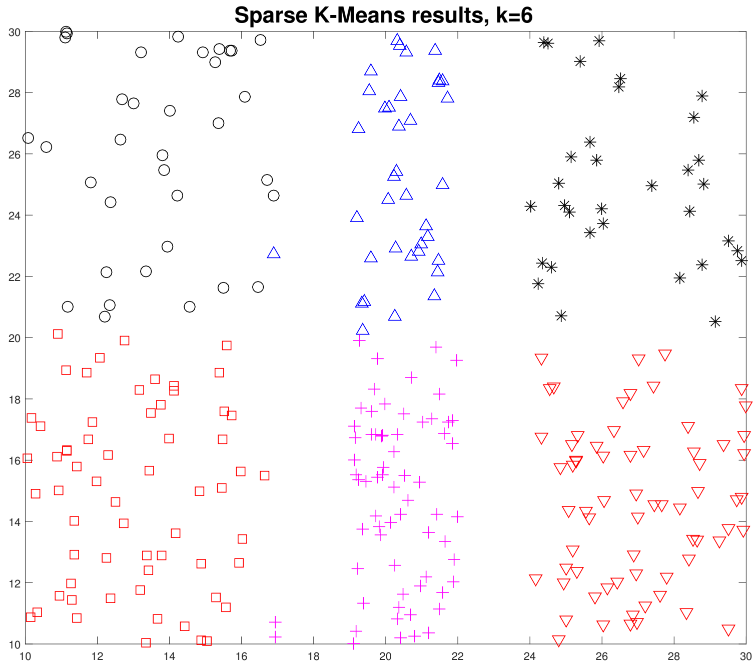


Fig. 3.8 Result of sparse K-Means where k=6

Table 3.3 The value of k for each sub-dataset

	T_1	T_2	T_3	T_4	T_5	T_6
Determined Number of k	5	5	3	4	6	5

Step 3 *Rule base initialisation*: A rule was extracted from each cluster, and all the extracted rules jointly initialised the rule base. Suppose a TSK fuzzy rule R_r $(1 \leq r \leq n)$, where n is the total number of generated fuzzy rules, can be expressed as:

$$\begin{aligned} R_r : \textbf{IF } x_1 \text{ is } A_{1r} \text{ and } x_2 \text{ is } A_{2r} \\ \textbf{THEN } y = \beta_{0r} + \beta_{1r}x_1 + \beta_{2r}x_2, \end{aligned} \tag{3.18}$$

where A_{1r} and A_{2r} are triangular fuzzy sets, which can be represented as $A_{1r} = (a^r_{11}, a^r_{12}, a^r_{13})$ and $A_{2r} = (a^r_{21}, a^r_{22}, a^r_{23})$. The generated 28 TSK fuzzy rules are detailed in Table 3.4.

Table 3.4 Generated raw TSK rule base

No.	**Input 1**			**Input 2**			**Output**		
(r)	a^r_{11}	a^r_{12}	a^r_{13}	a^r_{21}	a^r_{22}	a^r_{23}	β_{0r}	β_{1r}	β_{2r}
1	10.08	12.72	15.36	26.22	27.00	27.78	-0.14	0.06	-0.38
2	13.22	14.87	16.52	27.87	28.84	29.82	-0.00	0.28	-8.24
3	11.82	13.02	14.22	22.97	24.46	25.96	-0.15	-0.02	1.72
4	10.90	13.68	16.45	18.85	20.50	22.16	-0.04	-0.20	4.47
5	10.06	11.18	12.30	14.89	16.91	18.93	0.25	-0.06	-1.49
6	13.16	14.57	15.99	15.63	17.69	19.76	0.00	-0.21	4.30
7	19.24	20.48	21.71	26.83	28.25	29.68	0.14	-0.06	-1.21
8	19.59	20.52	21.45	22.15	22.89	23.64	0.24	0.08	-6.50
9	19.20	20.39	21.58	23.90	24.66	25.42	0.30	0.00	-6.00
10	10.16	11.26	12.37	10.85	12.44	14.02	0.12	0.09	-2.22
11	13.36	15.15	16.95	10.05	10.78	11.51	-0.01	0.30	-2.85
12	12.25	14.14	16.03	11.75	13.42	15.09	0.05	0.10	-2.21
13	20.51	21.24	21.97	15.28	17.48	19.69	-0.17	0.12	1.32
14	19.27	20.31	21.35	19.30	20.33	21.35	0.13	-0.00	-2.49
15	16.64	18.59	20.54	15.11	16.71	18.32	-0.27	-0.00	5.31
16	20.95	21.42	21.89	11.14	12.25	13.35	-0.17	-0.13	4.92
17	19.23	20.61	21.98	12.46	13.57	14.69	-0.28	-0.01	5.71
18	19.10	20.14	21.19	10.00	10.81	11.62	-0.06	-0.03	1.52
19	24.39	25.46	26.53	28.19	28.94	29.68	-0.00	-0.27	8.02
20	24.02	25.03	26.04	21.77	24.08	26.39	-0.01	0.05	-0.13
21	27.38	28.63	29.89	21.95	24.91	27.88	-0.27	-0.00	8.17
22	24.32	26.04	27.75	17.93	19.32	20.71	0.03	0.28	-6.23
23	24.31	25.73	27.15	15.75	16.37	16.98	0.09	0.15	-5.72
24	28.40	29.19	29.99	15.89	18.20	20.52	0.23	0.03	-7.43
25	26.03	27.77	29.52	10.50	11.39	12.29	0.07	-0.18	-0.01
26	24.15	25.29	26.42	10.16	11.62	13.08	0.03	-0.26	1.80
27	28.40	29.16	29.92	12.79	13.89	14.99	0.29	-0.02	-8.38
28	25.07	26.62	28.17	12.92	13.92	14.92	0.18	-0.06	-4.66

3.4.1.2 TSK Fuzzy Interpolation

Given any input, overlapped with any rule antecedent or not, the proposed TSK+ inference engine is able to generate an output. For instance, a randomly generated testing data point was ($A^*_1 = (27.37, 27.37, 27.37), A^*_2 = (13.56, 13.56, 13.56)$). The proposed TSK+ inference engine firstly calculated the similarity degrees between the given input and the antecedents of every rule ($S(A^*_1, A_{r1}), S(A^*_2, A_{r2}), r = \{1, 2, ..., 28\}$) using Eq. 3.7, with the results listed in Table 3.5.

Table 3.5 Sub result from each rule and its calculation details

p	$S(A_1^*, A_{p1})$	$S(A_2^*, A_{p2})$	FD_p	Consequence
1	0.0053	0.059	0.053	-0.014
2	0.020	0.003	0.003	-0.012
3	0.004	0.010	0.004	-0.014
4	0.001	0.836	0.001	0.006
5	0.007	0.455	0.007	0.004
6	0.016	0.780	0.016	0.023
7	0.467	0.157	0.157	0.170
8	0.461	0.004	0.004	0.008
9	0.449	0.052	0.052	0.118
10	0.012	0.954	0.012	0.018
11	0.024	0.870	0.024	0.024
12	0.002	0.939	0.002	0.005
13	0.211	0.848	0.211	-0.430
14	0.567	0.812	0.567	-0.996
15	0.440	0.480	0.440	0.114
16	0.591	0.941	0.591	-0.913
17	0.480	0.969	0.480	-1.042
18	0.414	0.871	0.414	-0.232
19	0.926	0.009	0.009	0.005
20	0.932	0.004	0.004	0.014
21	0.925	0.077	0.077	0.005
22	0.927	0.868	0.868	-0.937
23	0.918	0.736	0.736	-0.471
24	0.932	0.616	0.616	-1.030
25	0.948	0.902	0.902	-0.622
26	0.947	0.966	0.947	-0.547
27	0.906	0.913	0.906	-0.849
28	0.920	0.964	0.920	-0.589

From the calculated similarity degrees, the firing strength of each rule FS_r was computed using Eq. 3.2, and the sub-consequence from each rule was calculated using Eq. 3.4, which are also shown in Table 3.5. From this, the final output of the given input was calculated using Eq. 3.5 as $y = -0.902$.

3.4.1.3 Rule Base Optimisation

In order to achieve the optimal performance, the generated raw rule base was fine-tuned using the GA algorithm, as detailed in Chapter 3.3.2. The GA parameters used in this experiment are listed in Table 3.6. The population was initialised as the individuals representing the raw rule base and its random variations. The performance against the number of iterations is shown in Fig. 3.9, which clearly demonstrates the performance improvements led by the GA.

Table 3.6 GA parameters

Parameters	**Values**
Population Size	100
max value in fitness function	2
Crossover rate	0.85
Mutation rate	0.05
Maximum Iteration	10,000
Termination threshold	0.01

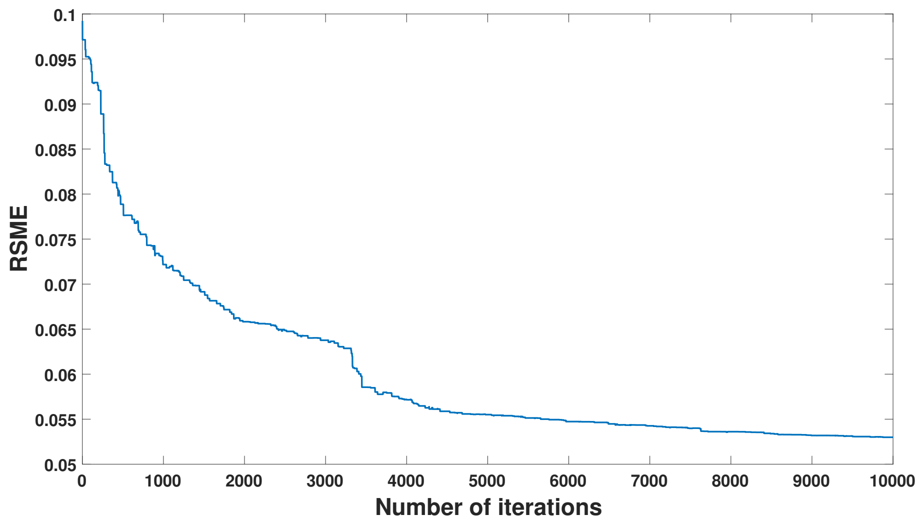


Fig. 3.9 RMSE values decrease over time during optimisation

3.4.1.4 Results Comparison

In order to enable a direct comparative study with the support of the approaches proposed in [11] and [178], the suggested approach was also applied to 36 randomly generated testing data points. The sum of errors for the 36 testing data led by the proposed approach is shown in Table 3.7, in addition to those led by the compared approaches. The proposed TSK+ outperformed the approaches proposed in [11] and [178], although fewer rules have been used. Another advantage of the suggested approach is that the number of rules led by the proposed system was determined automatically by the system without the requirement of any human inputs. In addition, noticeably, the optimisation process significantly improved the system performance by reducing the sum of errors from 3.38 to 1.78.

Table 3.7 Results led by approaches in experiment 1

Approaches	No. of rules	Sum of error	RSME
[11]	36	5.8	0.161
[178]	36	3.1	0.086
TSK+ without GA	28	3.38	0.094
TSK+ with GA	28	1.78	**0.067**

3.4.2 Experiment 2

The proposed TSK+ inference approach is not only able to deal with sparse or unevenly distributed datasets, but also able to work with dense datasets. In order to evaluate its ability in handling dense datasets, a two input and a single output non-linear mathematical model, expressed in Eq. 3.19, was employed as a test bed, given that it has been used by [62] and [158].

$$f(x,y) = \frac{sin(x)}{x} \cdot \frac{sin(y)}{y}. \tag{3.19}$$

In this case, a randomly generated dense training dataset was used, including 1,681 data points distributed within the range of $[-10,10] \times [-10,10]$. By employing the proposed approach, 14 TSK fuzzy rules in total were generated. In the work presented in [62] and [158], the mean-squares error (MSE) was used as the measurement of models. In order to allow a direct comparison, MSE is then used as the the measurement method in the experiment 2. The detailed calculations of this experiment are omitted here. The MSE values led by different approaches with the specified number of rules are listed in Table 3.8. The results demonstrated that the proposed system TSK+ performed competitively with only 14 TSK fuzzy rules.

Table 3.8 Results led by fuzzy models in experiment 2

Approaches	No. of rules	MSE
[62]	49	0.001
[158]	13	0.0004
[158]	14	0.0002
TSK+ with GA	14	0.0002

3.4.3 Discussion

The proposed TSK+ fuzzy inference approach and data-driven rule base generation method has been evaluated by applying two mathematical models. For each experiment, the performance of the proposed TSK+ inference approach was compared with the works presented in the literature. The results, which are given in Table 3.7 and 3.8, confirmed that the TSK+ inference approach contributed the competitive performances.

For the experiment 1, 28 TSK fuzzy rules have been extracted from a simulated sparse dataset, thus to construct a sparse TSK rule base to support the TSK+ fuzzy reasoning. The TSK+ inference approach contributed the best performance in the ability of dealing with sparse rule bases. In particular, the system performance was significantly improved after the parameters optimisation by adopting the GA.

For the experiment 2, a dense dataset, which contains 14 TSK fuzzy rules, has been created from a simulated dense dataset. The TSK+ inference approach also performed competitively, thus to demonstrate the power of handling with dense rule bases.

The proposed approach is the first attempt to extend the idea of the fuzzy interpolation to TSK fuzzy inference, and the experiments results show that the proposed approach demonstrated the great performances in handling both sparse and dense rule bases. This will, therefore, provide an additional alternative solution for those existing applications of FRI, such as [112], and at the same time an opportunity to enjoy the advantages of TSK fuzzy inference. This will also enable extensions of existing FRI, such as the experience-based rule base generation and adaptation approach [112], to work with TSK inferences targeting a wider range of applications.

The rule base for the traditional TSK fuzzy model, which was used in this initial work, was generated by the linear regression algorithm, based on a randomly generated dataset. Note that a recent development on sparse rule base updating and generating has been reported [178]. Although this approach was implemented on the Mamdani inference, the underpinning principle can be used to generate a sparse TSK rule base. In particular, given a training dataset, a sparse TSK rule base can be generated directly from data by strategically locating the important regions for fuzzy modelling [178], thus to boost the applicability of the proposed approach. In addition, due to the limitation of the proposed measurement of the similarity degree, which is not applicable for dealing with the Gaussian membership function, the proposed approach is not suitable for rule bases that contain the Gaussian membership functions. However, how the proposed system could deal with the Gaussian membership functions will remain as the future works.

3.5 Summary

This chapter proposed a fuzzy inference system TSK+, which extended the conventional TSK inference system, such that it is also applicable to sparse rule bases and unevenly distributed rule bases. This is achieved by allowing the interpolation and extrapolation of the output from all rules, even if the given input does not overlap with any rule antecedents. The chapter also proposed a novel data-driven rule base generation approach, which is workable with sparse, dense, and unevenly distributed datasets. The system was validated and evaluated by applying two benchmark problems. The experimental results demonstrated the wide applicability of the proposed system with compact rule bases and competitive performances.

Chapter 4

Interval Type-2 TSK+ Fuzzy Inference System

Type-2 fuzzy sets generalise the standard type-1 fuzzy sets by representing the membership of each element as a standard type-1 fuzzy set, while IT2 fuzzy sets represent the membership as a crisp interval bounded by [0, 1]. Such extension can handle rule uncertainties in a more flexible and effective way, but generally requires more computational power at the same time. Regardless of the type of fuzzy sets used, a complete dense rule base which covers the entire input domains, is always required by either Mamdani or TSK fuzzy inference approaches; otherwise, no rule can be fired and consequently no result can be generated when a given input does not overlap with any rule antecedent in the rule base.

In this chapter, the proposed TSK+ fuzzy interpolation approach, which was introduced in the previous chapter, is extended to allow the utilisation of interval type-2 (IT2) TSK fuzzy rule bases. The extended fuzzy inference system TSK+ is able to work with: 1) sparse rule bases, 2) dense rule bases, 3) type-1 fuzzy sets, and 4) interval type-2 fuzzy sets. The proposed system has been applied to two problems that have been considered in the literature. One illustrative case based on an example problem from the literature demonstrates the working of the proposed system, and the application on the cart centring problem reveals the power of the proposed method. The experimental investigation confirmed that the proposed approach is able to perform fuzzy inferences using either dense or sparse interval type-2 TSK rule bases, with promising results generated.

The rest of this chapter is structured as follows. Chapter 4.1 reviews (interval) type-2 fuzzy sets. Chapter 4.2 details the proposed IT2 TSK+. Chapter 4.3 reports the experimentation and analyses the experimental results. Chapter 4.4 concludes the chapter and suggests probable future developments.

4.1 Type-2 Fuzzy Sets

The membership grades of each element in type-1 fuzzy systems are crisp values, which minimises the effect of uncertainty handling. Type-2 fuzzy systems can handle more uncertainty using type-2 fuzzy sets. An type-2 fuzzy set, denoted as $\tilde{A}$, can be represented as:

$$\begin{aligned}\tilde{A} =&\{((x,u),\mu_{\tilde{A}}(x,u))|\forall x \in X, \forall u \in J_x \subseteq [0,1],\\ &\mu_{\tilde{A}}(x,u) \in [0,1]\}\ ,\\ =&\int_{x\in X}\int_{u\in J_x}\mu_{\tilde{A}}(x,u)/(x,u)\ ,\end{aligned} \tag{4.1}$$

where X is the primary domain, J_x is the primary membership for a given element x, and $\mu_{\tilde{A}}(x,u)$ denotes the secondary membership. When $\mu_{\tilde{A}}(x,u) = 1$, $\tilde{A}$ is deteriorated as an IT2 fuzzy set, which can be expressed as:

$$\tilde{A} = \int_{x\in X}\int_{u\in J_x} 1/(x,u),\ \ J_x \subseteq [0,1]. \tag{4.2}$$

An example trapezoidal IT2 fuzzy set $\tilde{A}$ is illustrated in Fig. 4.1, which can be represented by a lower membership function (LMF), $\underline{\tilde{A}} = (\underline{a_1},\underline{a_2},\underline{a_3},\underline{a_4},\underline{w})$, and an upper membership function (UMF), $\overline{\tilde{A}} = (\overline{a_1},\overline{a_2},\overline{a_3},\overline{a_4},\overline{w})$. In this case, $\tilde{A} =< \underline{\tilde{A}},\overline{\tilde{A}} >$, where $(\underline{a_1},\underline{a_2},\underline{a_3},\underline{a_4})$ and $(\overline{a_1},\overline{a_2},\overline{a_3},\overline{a_4})$ are respectively the four odd points of the LMF and UMF, and $\underline{w}$ and $\overline{w}$ denote respectively the degrees of confidence for $\underline{\tilde{A}}$ and $\overline{\tilde{A}}$, with $0 < \underline{w} \leq \overline{w} = 1$. The area between the LMF and UMF, illustrated in grey in Fig. 4.1, thus denotes the footprint of uncertainty (FOU), which represents the uncertainty of the fuzzy set $\tilde{A}$. Obviously, a larger FOU area implies a higher level of uncertainty, and the IT2 fuzzy set degenerates to a type-1 fuzzy set when $\underline{\tilde{A}}$ coincides with $\overline{\tilde{A}}$ (i.e., the area of $FOU(\tilde{A})$ is 0).

Using the concept of FOU, Eq.n 4.2 can also be rewritten as [130]:

$$\tilde{A} = 1/FOU(\tilde{A}). \tag{4.3}$$

4.2 Interval Type-2 TSK+

Many type-1 fuzzy systems have been extended to support IT2 fuzzy systems, including the TSK approach [60, 19]. Generally speaking, the inputs and all the fuzzy sets in the rule antecedents can but not necessarily be IT2 fuzzy sets in an IT2 fuzzy systems; and the consequence of IT2 TSK rules are zero or the first order of polynomial functions, where the

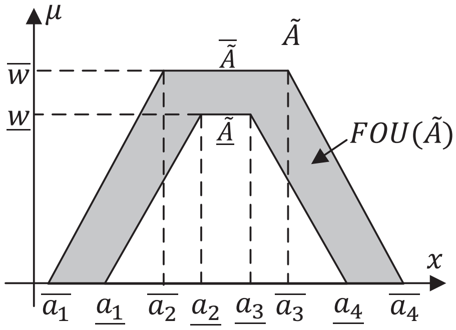


Fig. 4.1 LMF and UMF of a trapezoidal IT2 fuzzy set

parameters can be either crisp values or a crisp interval. Assume that an IT2 sparse TSK rule base is comprised of n rules as:

$$
\begin{aligned}
R_1 &: \textbf{IF } x_1 \text{ is } \tilde{A}_1^1 \text{ and } \ldots \text{ and } x_k \text{ is } \tilde{A}_k^1 \\
&\quad \textbf{THEN } y = \tilde{p}_0^1 + \tilde{p}_1^1 x_1^1 + \cdots + \tilde{p}_k^1 x_k^1 \\
&\ldots \\
R_i &: \textbf{IF } x_1 \text{ is } \tilde{A}_1^i \text{ and } \ldots \text{ and } x_k \text{ is } \tilde{A}_k^i \\
&\quad \textbf{THEN } y = \tilde{p}_0^i + \tilde{p}_1^i x_1^i + \cdots + \tilde{p}_k^i x_k^i \\
&\ldots \\
R_n &: \textbf{IF } x_1 \text{ is } \tilde{A}_1^n \text{ and } \ldots \text{ and } x_k \text{ is } \tilde{A}_k^n \\
&\quad \textbf{THEN } y = \tilde{p}_0^n + \tilde{p}_1^i x_1^n + \cdots + \tilde{p}_k^n x_k^n,
\end{aligned}
\tag{4.4}
$$

where $\tilde{A}_j^i, (j \in \{1,\ldots,k\}, i \in \{1,\ldots,n\})$, is an IT2 fuzzy set regarding input variable x_j in the i^{th} rule. The consequence is a crisp polynomial function $y = f_i(x_1, \ldots, x_k) = \tilde{p}_0^i + \tilde{p}_1^i x_1^i + \cdots + \tilde{p}_k^i x_k^i$, where $\tilde{p}_j^i$ are parameters usually being crisp intervals, with crisp singleton numbers being the special case. For a given input $O(\tilde{A}_1^*, \cdots, \tilde{A}_k^*)$, the steps introduced in Chapter 3.2.2 can generally be used for the generation of the output, but all the operations on type-1 fuzzy sets involved in these steps need to be upgraded to those based on IT2 fuzzy sets. Fortunately, such upgrading has been extensively studied in the literature [130], and all the operations on IT2 fuzzy sets used in this work can be simplified by the calculation of the *FOU* of the resultant IT2 fuzzy sets based on Eq. 4.3, which in turn can be simplified by the calculation of the LMF and UMF. Note that crisp numbers, sets and type-1 fuzzy sets are special cases of IT2 fuzzy sets, and thus the discussion below is all based on the general case of IT2 fuzzy sets, unless only a simplified version can be the case.

4.2.1 Firing Strength

The firing strength and similarity measure based on type-1 fuzzy sets, which was introduced in Chapter 3.2.1, is extended in this subsection to support the development of the IT2-TSK+. Assume that two weighted convex trapezoidal fuzzy sets in a normalised variable domain are given as $A = (a_1, a_2, a_3, a_4, w)$ and $A^{'} = (a_1^{'}, a_2^{'}, a_3^{'}, a_4^{'}, w^{'})$, where $0 < w \leq 1$ and $0 < w^{'} \leq 1$ represent the degrees of confidence for fuzzy sets A and $A^{'}$, respectively. Note that the weighted fuzzy set A will deteriorate to a normal fuzzy set when $w = 1$, which is usually simply denoted as $A = (a_1, a_2, a_3, a_4)$. The similarity degree $s(A, A^{'})$ between A and $A^{'}$ can be calculated by the following equation:

$$s(A,A') = \left(1 - \frac{\sum_{i=1}^{4} |a_i - a_i'|}{4}\right) \cdot d \cdot \frac{min(w,w^{'})}{max(w,w^{'})}, \tag{4.5}$$

where d represents $distance\ factor$, which has been detailed in Chapter 3.2.1, and it is able to be obtained by Eq. 3.8. As each IT2 fuzzy set can be repressed by a set of embedded type-1 fuzzy sets, the matching degree between a crisp value and an IT2 fuzzy sets can be calculated as the complete set of matching degrees between the crisp value and every embedded type-1 fuzzy set within the calculated IT2 fuzzy set using Eq. 4.5, and the calculated matching degree is thus a crisp interval.

Without losing generalisation, given a crisp singleton IT2 observation item $\tilde{A}_j$ and the corresponding antecedent value $\tilde{A}^i_j$ of rule R_i regarding the same antecedent variable x_j, their matching degree S is calculated as:

$$\tilde{s}(\tilde{A}^i_j, \tilde{A}_j) = [s(\underline{\tilde{A}^i_j}, \tilde{A}_j), s(\overline{\tilde{A}^i_j}, \tilde{A}_j)], \tag{4.6}$$

where $\overline{\tilde{A}^i_j}$ and $\underline{\tilde{A}^i_j}$ respectively indicate the UMF and LMF of $\tilde{A}^i_j$, and $s(\cdot,\cdot)$ represents the similarity degree between two type-1 fuzzy sets (or specifically one crisp singleton and one type-1 fuzzy set), as defined in Eq. 4.5.

Once the similarity degrees between the observed values and the rule antecedent values of rule R_i regarding every antecedent variable $x_1, \ldots, x_n$ are obtained, they are then integrated as the firing strength $\tilde{\alpha}_i$ of rule R_i. The integration is implemented by a t-norm operator when the similarity degrees are crisp values, as in type-1 fuzzy systems, but this needs to be extended to the meet $\sqcap$ operation when intervals are used [205]:

$$
\begin{aligned}
\tilde{\alpha}_i =&[\underline{\tilde{\alpha}_i}, \overline{\tilde{\alpha}_i}] \\
=&\sqcap_{j=1}^{k} \tilde{s}(\tilde{A}^i_j, \tilde{A}_j) \\
=&[\min(A_1), \max(A_1)] \sqcap \ldots \sqcap [\min(A_n), \max(A_n)] \\
=&[\min(\min(A_1), \ldots, \min(A_n)), \\
&\min(\max(A_1), \ldots, \max(A_n))] \\
=&[\tilde{s}(\tilde{A}^i_1, \tilde{A}_1) \sqcap \cdots \sqcap \tilde{s}(\tilde{A}^i_k, \tilde{A}_k)] \\
=&[[s(\underline{\tilde{A}^i_1}, \underline{\tilde{A}_1}), s(\overline{\tilde{A}^i_1}, \overline{\tilde{A}_1})], \cdots, \\
&[s(\underline{\tilde{A}^i_k}, \underline{\tilde{A}_k}), s(\overline{\tilde{A}^i_k}, \overline{\tilde{A}_k})]] \\
=&[s(\underline{\tilde{A}^i_1}, \underline{\tilde{A}_1}) \wedge \cdots \wedge s(\underline{\tilde{A}^i_k}, \underline{\tilde{A}_k}), \\
&s(\overline{\tilde{A}^i_1}, \overline{\tilde{A}_1}) \wedge \cdots \wedge s(\overline{\tilde{A}^i_k}, \overline{\tilde{A}_k})],
\end{aligned}
\tag{4.7}
$$

where $\wedge$ represents a t-norm operation implemented as a minimum operator in this work.

4.2.2 Intermediate Result from Individual Rule

As detailed in Chapter 3.2.2, the TSK+ inference approach integrates the intermediate results from every individual rules in the rule base to form the final output. Given an observation $O(\tilde{A}_1, \cdots, \tilde{A}_k)$, the intermediate result $\tilde{c}^i$ led by rule R_i needs to be calculated first. As intervals are usually used as the parameters of the polynomial function in the consequence of IT2 TSK rules and the domains of input variables are normalised, each sub-consequence $\tilde{c}^i$ from rule R_i in the IT2 TSK+ system is therefore a crisp interval [19, 60]. The minimum and maximum values of $\tilde{c}^i$ can be obtained based on the given observation and the corresponding IT2 polynomial function of the rule consequence:

$$
\begin{aligned}
\tilde{c}^i =&\tilde{p}^i_0 + \tilde{p}^i_1 x^i_1 + \cdots + \tilde{p}^i_k x^i_k \\
=&[\underline{\tilde{p}^i_0} + \underline{\tilde{p}^i_1} x_1 + \cdots + \underline{\tilde{p}^i_k} x_k\ , \\
&\overline{\tilde{p}^i_0} + \overline{\tilde{p}^i_1} x_1 + \cdots + \overline{\tilde{p}^i_k} x_k],
\end{aligned}
\tag{4.8}
$$

where $\underline{\tilde{p}^i_j}$ and $\overline{\tilde{p}^i_j}$, $(j \in \{0, 1, \cdots, k\})$, denote the minimum and maximum values of crisp interval $\tilde{p}^i_j$, respectively.

4.2.3 Final Output Generation

The final output of the TSK+ system is a fuzzified weighted average of the sub-consequences from all rules. In particular, based on the obtained firing strength $\tilde{\alpha}^i$ and corresponding sub-consequence $\tilde{c}^i$, the final interval output $\tilde{c}$ can be calculated by:

$$\begin{aligned}\tilde{c} =& [\underline{\tilde{c}}, \overline{\tilde{c}}] \\ =& \int_{\tilde{c}^1 \in [\underline{\tilde{c}^1}, \overline{\tilde{c}^1}]} \cdots \int_{\tilde{c}^n \in [\underline{\tilde{c}^n}, \overline{\tilde{c}^n}]} \int_{\tilde{\alpha}^1 \in [\underline{\tilde{\alpha}^1}, \overline{\tilde{\alpha}^1}]} \cdots \int_{\tilde{\alpha}^n \in [\underline{\tilde{\alpha}^n}, \overline{\tilde{\alpha}^n}]} \\ & 1 \Bigg/ \frac{\sum_{i=1}^{n} \tilde{\alpha}^i \cdot \tilde{c}^i}{\sum_{i=1}^{n} \tilde{\alpha}^i} .\end{aligned} \tag{4.9}$$

This equation can be practically implemented by computing the two extreme values of the crisp interval, minimum $\underline{\tilde{c}}$ and maximum $\overline{\tilde{c}}$, separately:

$$\begin{cases} \underline{\tilde{c}} = \dfrac{\sum\limits_{i=1}^{L} \overline{\tilde{\alpha}^i} \underline{\tilde{c}^i} + \sum\limits_{j=L+1}^{n} \underline{\tilde{\alpha}^i \tilde{c}^i}}{\sum\limits_{i=1}^{L} \overline{\alpha}_i + \sum\limits_{j=L+1}^{n} \underline{\alpha}_j} \\ \overline{\tilde{c}} = \dfrac{\sum\limits_{i=1}^{R} \overline{\tilde{\alpha}^i \tilde{c}^i} + \sum\limits_{j=R+1}^{n} \underline{\tilde{\alpha}^i} \overline{\tilde{c}^i}}{\sum\limits_{i=1}^{R} \overline{\alpha}_i + \sum\limits_{j=R+1}^{n} \underline{\alpha}_j}, \end{cases} \tag{4.10}$$

where L and R are the *switch points* that are used to make sure $\underline{\tilde{c}}$ is minimised and $\overline{\tilde{c}}$ is maximised, which can be obtained by an iterative procedure. A number of implementations on such a problem have been proposed in the literature and widely used in the real world, such as Karnik-Mendel (KM) algorithms, enhanced Karnik-Mendel algorithms (EKMA), an iterative algorithm with a stop condition (ISAC), and enhanced ISAC [196]. In particular, the Karnik-Mendel (KM) algorithm is adapted in this work due to its efficiency, and the details of this approach are omitted here as this is beyond the focus of this chapter.

Once the output interval or special IT2 fuzzy set is generated, type reduction or defuzzification needs to be applied. This can be readily implemented by applying a simple average operation:

$$c = \frac{\underline{\tilde{c}} + \overline{\tilde{c}}}{2}. \tag{4.11}$$

IT2 fuzzy sets are extensions of type-1 fuzzy sets. The above proposed IT2 fuzzy TSK+ approach deteriorates into type-1 TSK+ when all the IT2 TSK fuzzy rules degenerate into type-1 ones. There are variations of IT2 rule bases. For instances, the parameters in the rule consequences can be all crisp numbers, instead of crisp intervals. In this special case, the intermediate result led by Equation 4.8 is a singleton number. In addition, if the $\tilde{p}_1^i = \tilde{p}_2^i = \cdots = \tilde{p}_k^i = 0$ for all rules in the rule base in Eq. 4.4, the TSK+ model will become to a 0-order one, and Eq. 4.8 can then be rewritten as $\tilde{c}^i = [\underline{\tilde{p}_0^i}, \overline{\tilde{p}_0^i}]$. Therefore, the proposed approach is able to be applied to perform the inference with multiple varied TSK rule bases.

Note that rough-fuzzy set-based interpolation approaches may also provide similar approximation functionality practically, such as in [31, 38]. However, there is an obvious theoretical difference between the two as different underpinning approximation approaches are utilised. In addition, the rough-fuzzy set-based interpolation approaches are analogy-based and consider the representative values and the shapes of upper and lower membership functions separately during the reasoning processes, while the proposed approach considers the IT2 fuzzy set as a single component. The proposed approach utilises the similarity degree value between the given inputs and rule antecedents as a rule firing strength in aggregating the final result, which follows the conventional TSK inference principle.

4.3 Experimentation

The proposed IT2 TSK+ approach is validated and evaluated in this chapter by applying it to two cases used in the literature. In particular, the first example uses a mathematical model to illustrate the working procedure of the proposed approach, and the second example evaluates the proposed approach by considering a well-known cart centring problem.

4.3.1 Illustrative Example

A two inputs and signal output fuzzy inference problem, which has been used as an illustration example in the work of [195], is re-considered here. In particular, the domains of the two inputs are each fuzzy partitioned and represented by two trapezoidal IT2 fuzzy sets, and thus the rule base consists of four rules which covers the entire problem domain. The consequence of each rule is a crisp interval. The complete rule base is listed in Table 4.1, where each IT2 fuzzy set is represented as $\tilde{A} = (\overline{a}_1, \overline{a}_2, \overline{a}_3, \overline{a}_4, \overline{w}; \underline{a}_1, \underline{a}_2, \underline{a}_3, \underline{a}_4, \underline{w})$, and $(\overline{a}_1, \overline{a}_2, \overline{a}_3, \overline{a}_4, \overline{w})$ denotes the UMF of the IT2 fuzzy set with degree of confidence $\overline{w}$, and $(\underline{a}_1, \underline{a}_2, \underline{a}_3, \underline{a}_4, \underline{w})$

indicates the LMF with $\underline{w}$ being the confidence level. Given an input vector $O = (-0.3, 0.6)$, the calculation of the final crisp inference output is detailed below:

Table 4.1 Rule base for the illustrative example

No.	**Inputs**		**Output**
	x_1	x_2	y
R_1	(-1.5,-1.5,-0.5,1.5,1;-1.5,-1.5,-1.5,0.5,1)	(-1.5,-1.5,-0.5,1.5,1;-1.5,-1.5,-1.5,0.5,1)	$\tilde{p}_0^1 = [-1, -0.9]$
R_2	(-1.5,-1.5,-0.5,1.5,1;-1.5,-1.5,-1.5,0.5,1)	(-1.5, 0.5, 1.5, 1,5,1;-0.5, 1.5, 1.5, 1.5,1)	$\tilde{p}_0^2 = [-0.6, -0.4]$
R_3	(-1.5, 0.5, 1.5, 1,5,1;-0.5, 1.5, 1.5, 1.5,1)	(-1.5,-1.5,-0.5,1.5,1;-1.5,-1.5,-1.5,0.5,1)	$\tilde{p}_0^3 = [0.4, 0.6]$
R_4	(-1.5, 0.5, 1.5, 1,5,1;-0.5, 1.5, 1.5, 1.5,1)	(-1.5, 0.5, 1.5, 1,5,1;-0.5, 1.5, 1.5, 1.5,1)	$\tilde{p}_0^4 = [0.9, 1]$

The matching degree between each given input item and each antecedent item of every rule is calculated using Eq. 4.6, with the results shown in the second and third columns in Table 4.2. Note that Eq. 4.6 is an extension of Eq. 4.5, and the sensitive factor is set to 8, which is determined empirically. Having computed the matching degrees, the firing strength of each rule is obtained using Eq. 4.7, and the results are listed in the fourth column in Table 4.2.

Table 4.2 Firing strength for experimentation 1

i	x_1 $\tilde{s}(\tilde{A}_1^i, O)$	x_2 $\tilde{s}(\tilde{A}_2^i, O)$	Firing Strength $\tilde{\alpha}^i$
1	[0.3975, 0.6259]	[0.0443, 0.4195]	[0.0443, 0.4195]
2	[0.3975, 0.6259]	[0.5063, 0.6608]	[0.3975, 0.6259]
3	[0.1115, 0.5008]	[0.0443, 0.4195]	[0.0443, 0.4195]
4	[0.1115, 0.5008]	[0.5063, 0.6608]	[0.1115, 0.5008]

The intermediate result led by each rule is the corresponding rule consequence, as the given rule base only consists of 0-order TSK fuzzy rules. By applying the KM algorithm, the switching points $L = 1$ and $R = 3$ can be calculated. From this, the final inference output interval $\tilde{c} = [\underline{\tilde{c}}, \overline{\tilde{c}}]$ can be computed as:

$$\begin{aligned}\underline{\tilde{c}} &= \frac{\overline{\tilde{\alpha}^1}\underline{\tilde{p}_0^1} + \underline{\tilde{\alpha}^2}\underline{\tilde{p}_0^2} + \underline{\tilde{\alpha}^3}\underline{\tilde{p}_0^3} + \underline{\tilde{\alpha}^4}\underline{\tilde{p}_0^4}}{\overline{\tilde{\alpha}^1} + \underline{\tilde{\alpha}^2} + \underline{\tilde{\alpha}^3} + \underline{\tilde{\alpha}^4}} \\ &= \frac{0.42 \cdot (-1) + 0.40 \cdot (-0.6) + 0.04 \cdot 0.4 + 0.11 \cdot 0.9}{0.42 + 0.40 + 0.04 + 0.11} \\ &= -0.5636\end{aligned} \tag{4.12}$$

$$
\begin{aligned}
\overline{\tilde{c}} &= \frac{\underline{\tilde{\alpha}}^1 \overline{\tilde{p}_0^1} + \underline{\tilde{\alpha}}^2 \overline{\tilde{p}_0^2} + \underline{\tilde{\alpha}}^3 \overline{\tilde{p}_0^3} + \overline{\tilde{\alpha}}^4 \overline{\tilde{p}_0^4}}{\underline{\tilde{\alpha}}^1 + \underline{\tilde{\alpha}}^2 + \underline{\tilde{\alpha}}^3 + \overline{\tilde{\alpha}}^4} \\
&= \frac{0.04 \cdot (-0.9) + 0.40 \cdot (-0.4) + 0.04 \cdot 0.6 + 0.50 \cdot 1}{0.04 + 0.40 + 0.04 + 0.50} \\
&= 0.4064 \ .
\end{aligned} \tag{4.13}
$$

The final system output can be derived by applying a fuzzy type reduction method such as Eq. 4.11:

$$
\begin{aligned}
c &= \frac{\underline{\tilde{c}} + \overline{\tilde{c}}}{2} \\
&= \frac{-0.5636 + 0.4064}{2} \\
&= -0.0786 \ .
\end{aligned} \tag{4.14}
$$

This example demonstrates the working of the proposed TSK+ system with an IT2 fuzzy rule base. Note that the results generated by the conventional IT2 TSK inference approach, as reported in the work of [195], are $\underline{\tilde{c}} = -0.6316$, $\overline{\tilde{c}} = 0.4897$ and $c = -0.0710$. The final output led by the proposed IT2 TSK+ approach is very similar to the final result reported in [195], but the generated output interval is only a subset of the one reported in [195].

4.3.2 Cart Centring Application

The proposed IT2 TSK+ approach is the first attempt to involve the idea of the fuzzy interpolation to solve the sparse interval type-2 TSK fuzzy rule base problem. There is no similar approaches or applications existing that can be referred for comparison purpose. Therefore, in this experiment, the well-known cart centring problem, which has been considered as a dense IT2 TSK fuzzy model in [60], was used for evaluation purpose. In this particular problem, a cart can only move along a line on a frictionless plane, and the goal is to drive and keep the cart to the central position of this line from a given initial position, which forms a typical control problem, as illustrated in Fig. 4.2. The inputs of the controller for this problem are the current position coordinates of cart x and the current velocity of cart v, and the output of this fuzzy model is force F that should be applied on the cart. In [60], the domain of cart position x was restricted from $-0.75m$ to $0.75m$; the range of cart velocity v was restricted from $-0.75m/s$ to $0.75m/s$; the output force F was defined between $-0.18m/s$ and $0.18m/s$; and the sampling time used was $t = 0.1s$. This set of parameters and constraints were also utilised in this experiment, reported herein.

Based on the problem described above, a 0-order IT2 TSK fuzzy model has been designed and created in [60]. In particular, five linguistic values, represented as IT2 fuzzy sets, were

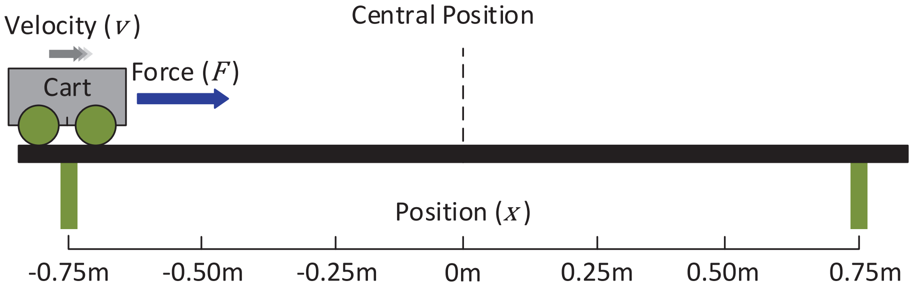


Fig. 4.2 The cart centring problem

used to cover every domain of input variables x and v, which are negative large (NL), negative small (NS), zero (0), positive small (PS) and positive large (PL), as shown in Fig. 4.3. Five crisp interval values were used as the output, which are labelled as NL, NS, 0, PS and PL, as listed in Table 4.3. A dense rule base for this cart centring problem was generated in [60], as shown in Table 4.4.

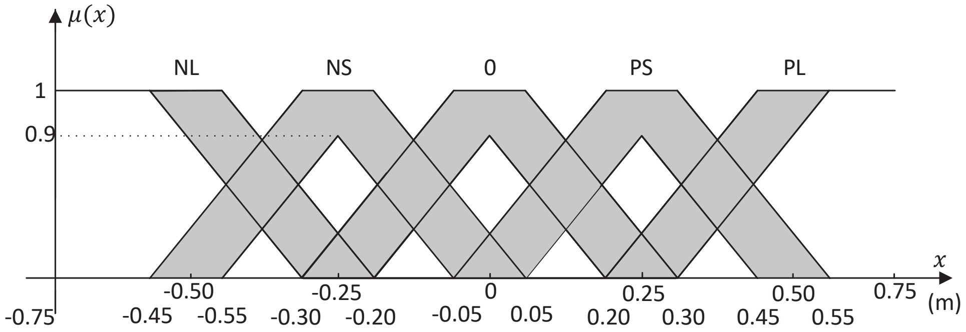
Table 4.3 Fuzzy partition of output domain

Output label	**Value**	**Linguistic value**
NL	[-0.18 -0.14]	NL
NS	[-0.10 -0.06]	NS
0	[-0.02 0.02]	0
PS	[0.06 0.10]	PS
PL	[0.14 0.18]	PL

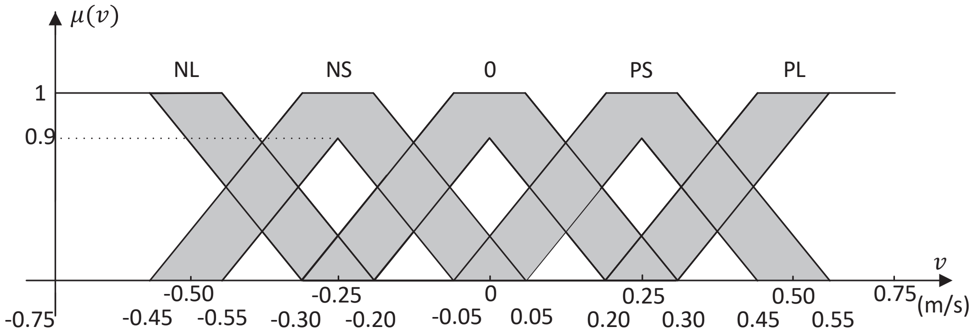
Table 4.4 Dense rule base with 25 rules used in [60]

		Position (x)				
		NL	**NS**	**0**	**PS**	**PL**
	NL	PL	PL	PL	PS	0
	NS	PL	PL	PS	0	NS
Velocity (v)	**0**	PL	PS	0	NS	NL
	PS	PS	0	NS	NL	NL
	PL	0	NS	NL	NL	NL

In order to evaluate the proposed IT2 TSK+ approach working with a sparse rule base, two fuzzy sets from each input domain, as shown in Fig. 4.3 were manually removed to simulate a lack of information, and the result is shown in Fig. 4.4. Consequently, from the incomplete information, a sparse rule base with only 9 rules was generated, as listed in Table 4.5.



(a) IT2 fuzzy partition for the domain of Position



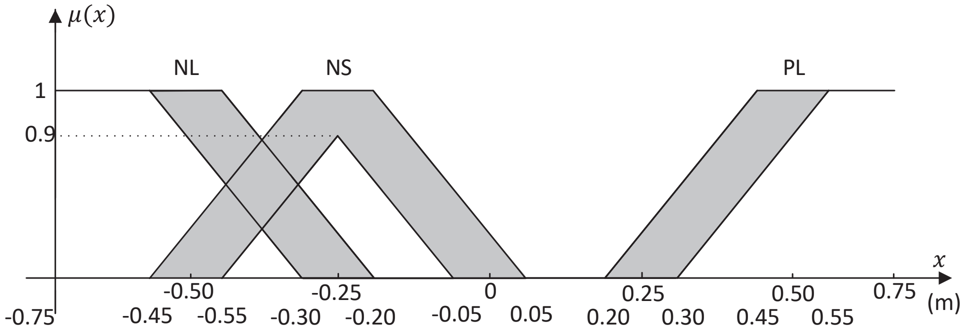
(b) IT2 membership function for the domain of Velocity

Fig. 4.3 Fuzzy partition on input domain

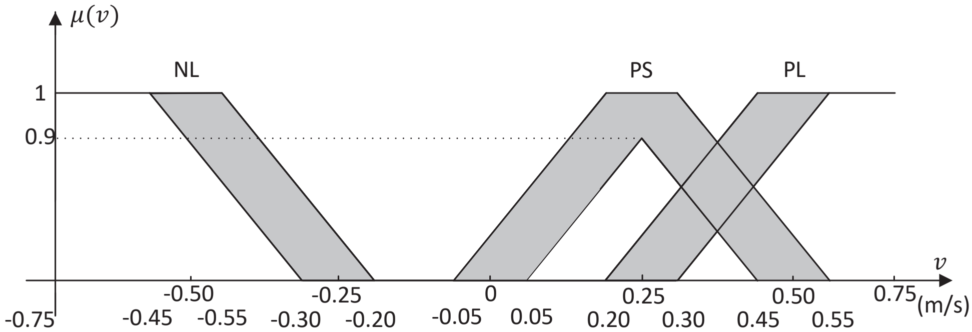
Table 4.5 Modified sparse rule base with 9 rules

		Position (x)		
		NL	**NS**	**PL**
	NL	PL	PL	0
Velocity (v)	**PS**	PS	0	NL
	PL	0	NS	NL

Given the initial state of the cart, $x_0 = 0.5m$ and $v_0 = 0.5m/s$, the conventional IT2 TSK approach and the proposed IT2 TSK+ were both applied in this experiment using the dense and sparse rule bases, if applicable, to drive the cart to the central position of the plane, with the results demonstrated in Fig. 4.5. In particular, the results led by the conventional IT2 TSK, of course, based on the dense rule base, are shown in Figs. 4.5(a) and 4.5(b); the results led by the proposed IT2 TSK+ based on the dense rule base are shown in Figs. 4.5(c) and 4.5(d);



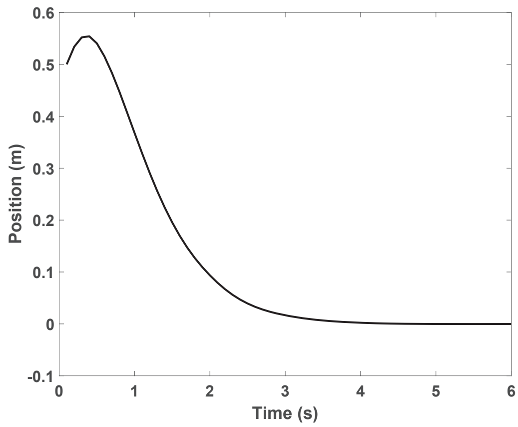
(a) Reduced number of IT2 linguistic values for position



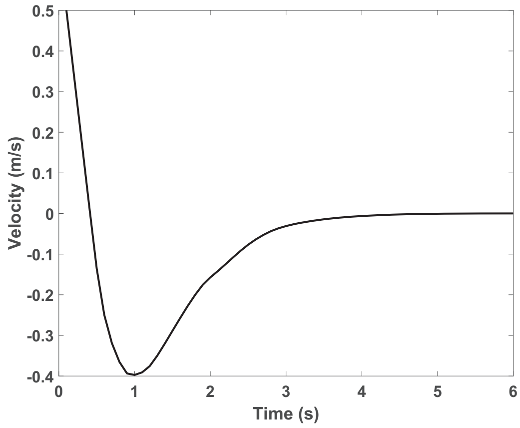
(b) Reduced number of IT2 linguistic values for velocity

Fig. 4.4 Reduced number of IT2 linguistic values for input domain

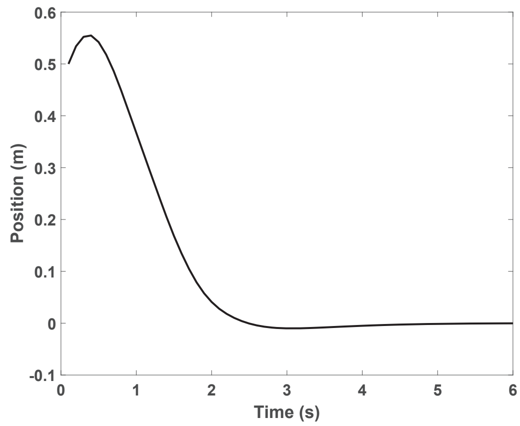
and the results generated by the proposed IT2 TSK+ approach using the sparse rule base are illustrated in Figs. 4.5(e) and 4.5(f). Note that only the cart's position and velocity are both become to 0 at the same time can indicate the cart reach the final goal, which is the cart has been moved and kept at the centre position of the plane.



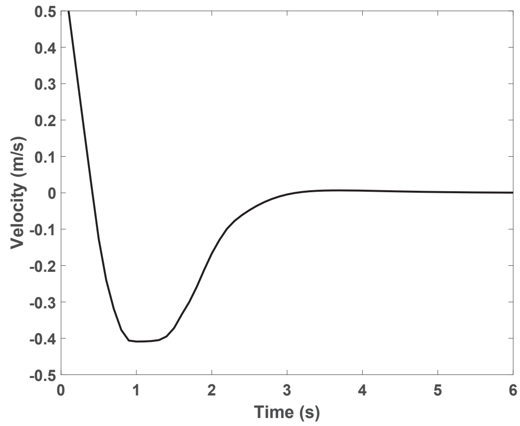
(a) Cart position by conventional TSK with dense rule base [60]



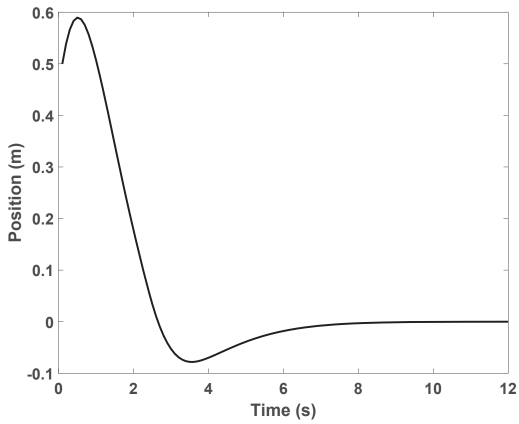
(b) Cart velocity by conventional TSK with dense rule base [60]



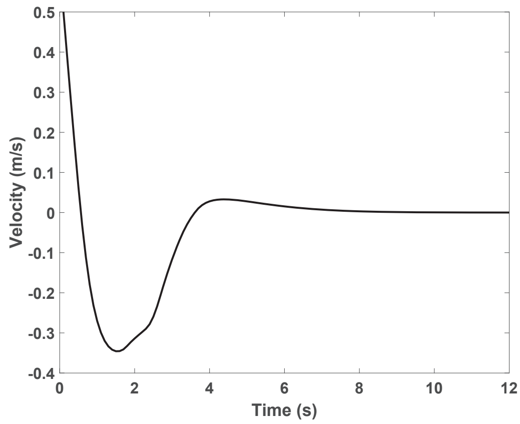
(c) Cart position by IT2 TSK+ with dense rule base



(d) Cart velocity by IT2 TSK+ with dense rule base



(e) Cart position by IT2 TSK+ with sparse rule base



(f) Cart velocity by ITS TSK+ with sparse rule base

Fig. 4.5 Performance comparison between proposed IT2 TSK+ approach with dense and sparse rule base and conventional TSK with dense rule base in [60]

4.3.3 Discussions

In experiment 1, an illustration example has been carried out to illustrate the procedure of the proposed IT2 TSK+ inference approach in details through a mathematical operation. And then, a cart centring problem has been simulated for the system evaluation in the second experiment. This experiment reveals that the proposed IT2 TSK+ approach is able to generate the outputs to drive the cart to the central position using either a dense or sparse rule base. From Fig. 4.5, it is clear that the proposed IT2 TSK+ with the dense rule base took less time to drive the cart from the initial position to the goal position with relatively smooth control, compared to the performance from the conventional TSK approach based on, of course, the dense rule base. This indicates that the proposed IT2 TSK+ system outperforms the conventional IT2 TSK method when the dense rule base is used. Also, interestingly, the proposed approach took longer to change the moving direction, which might be useful in real-world control for better dynamic stability.

The IT2 TSK+ also successfully drove the cart to the goal position with a relatively smooth curve, although the convergence time of the proposed IT2 TSK+ with a sparse rule base was longer than those with dense rule bases of either approach. However, if the size of utilised rule bases is taken into account, the proposed approach can solve the same control problem with only 9 rules, whilst a dense rule with 25 rules is required by the conventional approach. This clearly demonstrates the power of the proposed system in system complexity reduction.

The sparse rule base used in the second experiment was generated by manually removing some linguistic values from each variable domain rather arbitrarily, and thus the sparse rule base and correspondingly the inferred results may not be optimal. Therefore, better performance is expected from an optimal sparse rule base. Note that developments on sparse rule base generation have been reported in the literature [178, 115]. Although these approaches only targeted type-1 fuzzy models, the underpinning principle can be readily extended to generate sparse IT2 TSK rule bases, and this remains an active future work. Also, the cart is limited in its movement along a straight line only in this experiment. Note that fuzzy controllers have been applied to mobile robot control with no restriction on the cart movement [225]. Such complex control problems may better reveal the capability of the proposed approach.

Although many fuzzy interpolation approaches have been proposed to enable fuzzy inference with sparse rule bases, the majority of them were developed based on Mamdani rule bases, with some being extended to support IT2 fuzzy sets. The proposed system is the first attempt to extend the TSK fuzzy system with wider applicability, supporting either type-1 or IT2 fuzzy rule bases, which are either dense or sparse. This will significantly improve the performance of the widely applied TSK fuzzy inference systems in real-world applications, with better uncertainty management. For instance, a wall-following mobile robot controller has been proposed in [87]. In this system, an IT2 TSK fuzzy model is designed for mobile robot control, and the reinforcement method Q-learning is adopted to learn the IT2 TSK fuzzy rule base. The proposed IT2 TSK+ approach can be readily applied to the mobile robot system to make the best guess of the next action rather than simply a random one, when the rule base is extremely sparse. Consequently, the total number of trials in the learning process is expected to be reduced. The proposed approach may also be used for other real-world applications, which were developed based on type-1 fuzzy sets, such as [225], in an effort to boost the system performance. The implementation and the evaluation of such applications remain as pieces of future work. The system will, at the same time, also provide an effective form of system complexity reduction, especially in the era of big data.

4.4 Summary

This chapter extended the TSK+ fuzzy inference approach, which was introduced in the previous chapter, by allowing the utilisation of sparse IT2 TSK rule bases, as well as dense ones. Thanks to the extensive research carried out in the field of IT2 fuzzy sets and the corresponding computing approaches, this work also presented a practically feasible computing approach for real-world applications. IT2 TSK+ is, therefore, able to perform inferences with dense, sparse, type-1, or IT2 fuzzy rule bases. Two experiments adapted from the literature have been used for system validation and evaluation, with the first one illustrating the working of the system and the second one demonstrating the power of the proposed fuzzy inference system in mobile cart control.

Chapter 5

Experience-based Rule Base Generation and Adaptation

Fuzzy inference is one of the most advanced technologies in the control field. It has been widely applied to solve real-world problems due to its simplicity and effectiveness in representing and reasoning on human natural language. Examples of such applications include the subway control system in the city of Sendai [215], and the home heating control system [189], amongst others. Traditional fuzzy modelling requires either complete experts' knowledge or large data sets to generate rule bases such that the input spaces can be fully covered. Although fuzzy rule interpolation (FRI) relaxes this requirement by approximating rules using their neighbouring ones, it is still difficult for some real-world applications to obtain sufficient experts' knowledge and/or data to generate a reasonable sparse rule base to support FRI. Also, the generated rule bases are usually fixed and therefore cannot support dynamic situations. In order to address these limitations, this chapter presents a novel rule base generation and adaptation system to allow the creation of rule bases with minimal a priori knowledge. This is implemented by adding accurate interpolated rules into the rule base guided by a performance index from the feedback mechanism, and also by considering the rules' previous experience information as a weight factor in the process of rule selection for FRI. In particular, the selection of rules for interpolation in this work is based on a combined metric of the weight factors and the distances between the rules and a given observation, rather than being simply based on the distances. Two digitally simulated scenarios are employed to demonstrate the working of the proposed system, with promising results generated for both rule base generation and adaptation.

The rest of the chapter is structured as follows: Chapter 5.1 presents the proposed approach. Chapter 5.2 details a digital experimentation for demonstration and validation. Chapter 5.3 concludes the chapter by summarising the work.

5.1 Rule Base Generation and Adaptation

The proposed rule base generation and adaptation system for FRI is introduced in this section, and the system framework is outlined in Fig. 5.1, which is comprised of mainly four parts: rule base initialisation, rule selection for interpolation, transaction-based FRI and rule base revision. Firstly, an initial set of rules is generated from limited a priori knowledge. For a given observation, the system then selects the 'best' two rules from the current rule base for interpolation, based on a particular set of metrics, including the usage frequency/experience information, the previous *performance index* and the distances between the given observation and the rule antecedents. From this, a new rule is interpolated based on the selected 'best' rules from the given observation. Afterwards, the *performance index* will be utilised to support the rule base updating, whenever it is available from the feedback system.

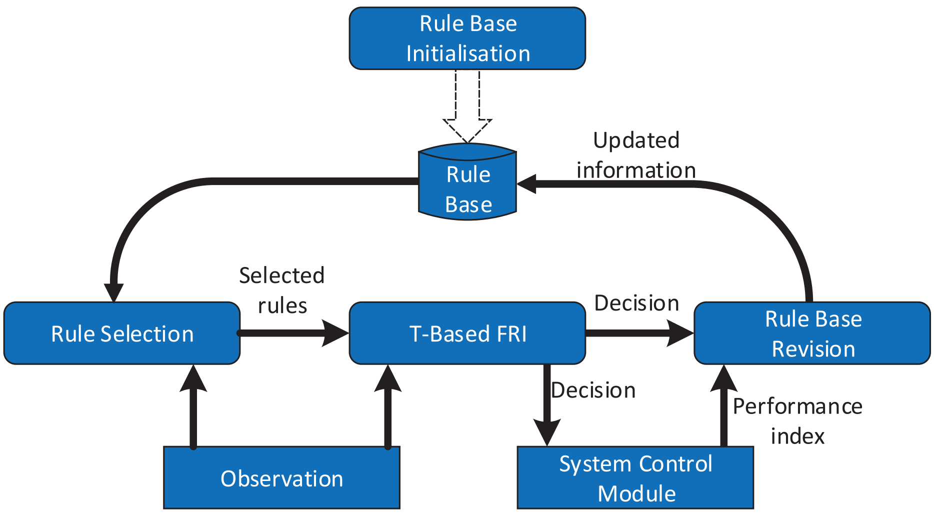


Fig. 5.1 The framework of the proposed system

5.1.1 Rule Base Initialisation

Traditionally, fuzzy rule bases can be generated from either human expertise or historic data; yet neither of them may be sufficiently available during the process of fuzzy modelling. For instance, most existing smart heater control systems require usage data for the rule base generation, as it is very difficult, if not impossible, for human experts to accurately learn the control patterns for different users regarding their living habits. However, collecting such data usually takes a long time, and the control systems cannot make intelligent decisions before this initial modelling progress has been done. This chapter addresses such problems to enable fuzzy modelling with very limited a priori knowledge. For simplicity, in this initial investigation, it is assumed that all the rules describing the boundary of the problem space are always available, although fundamentally, the idea underlying the proposed approach can be extended to support the situations where the rule base contains at least two rules for any single step of inference (even random ones), which remains for future work.

A weight is assigned to each individual rule in the rule base when the rule is created, which provides a measure to help select the 'best' rules for interpolation in FRI, and will be discussed in Section 5.1.4. In this work, only rules with single antecedents are considered. Without losing generality, suppose the following two fuzzy rules R_i and R_j are transformed from either human expertise or historic data:

$$\begin{aligned} R_i : \textbf{IF}\ \ x\ is\ A_i\ ,\quad \textbf{THEN}\ \ y\ is\ B_i\ (w_i,\ EF_i,\ CD_i) \\ R_j : \textbf{IF}\ \ x\ is\ A_j\ ,\quad \textbf{THEN}\ \ y\ is\ B_j\ (w_j,\ EF_j,\ CD_j), \end{aligned} \tag{5.1}$$

where w represents the weight of the rule, which is introduced in details later, and EF stands for *experience factor* and represents the usage and effectiveness of information of the particular rule in the previous FRI progresses. It can be increased or decreased during the system running, based on the employability of the rule and the effectiveness of the interpolated results. The rule with a greater EF indicates that it is of more experience and is more likely to generate an accurate result. CD stands for the *cooling down factor* and represents the times that the concerned rule has not been selected continuously so far. The introduction of the CD allows the control system to identify rules that are less likely to be selected for interpolation.

In the progress of the rule base initialisation, the EF and the CD are also initialised. The CD is always reconfigured as 0 once the corresponding rule has been selected to perform interpolation, based on its physical meaning. In this initial work, for simplicity, the EF is initialised as 50 based on initial investigation through experimentation. Briefly, a larger EF

usually leads to have a longer convergence time, and a smaller EF may result in rule removal unexpectedly. A further study of the initialisation of this factor remains for future work. These two factors jointly decide the importance or weight of rule R_i in the rule base as:

$$w_i = \left(\frac{2}{1+e^{-\frac{EF_i}{n}}} - 1\right)\left(1 - \frac{1}{1+5e^{-\frac{CD_i}{a}+b}}\right), \tag{5.2}$$

where a, b, n are sensitivity factors ($b > 3$, $a, n \neq 0$). A smaller value of b and a greater value of a and n make the system less sensitive to the rule weight, and vice versa. The values of these factors need to be determined based on the specific problems, but some early stage experimentation generally suggests $4 \leq b \leq 10$, $1 \leq a \leq 100$, and $1 \leq n \leq 200$.

5.1.2 Rule Selection

Fuzzy rule interpolation is utilised in this work to perform fuzzy inferences. In order to enable the utilisation of FRI, two rules need to be selected for interpolation. Dissimilar from traditional FRI approaches that select the two closest rules for interpolation (for a given distance metric), the proposed system selects rules for interpolation based on an importance factor (IF) with regard to a given input, which is a combined metric of rule weights and the distance between the given input and rule antecedents. The rule with the greatest IF value on each side of the observation is selected and used in the process of fuzzy rule interpolation.

Given an input 'x is A^*', suppose that there are n rules in the rule base '**IF** x is A_i, **THEN** y is B_i (w_i, EF_i, CD_i)', $i \in \{1, 2, ..., n\}$, then the importance factor for each rule can be calculated as follows:

$$IF_i = \sqrt{\lambda_i^{'}}\, w_i\ , \tag{5.3}$$

where w_i is computed using Eq. 5.2, and $\lambda_i^{'}$ is the *inverse distance weighting factor* (IDWF) $\lambda_i^{'}$ for each rule in the rule base using the following equation:

$$\lambda_i^{'} = \frac{\frac{1}{d_i}}{\sum_{l=1}^{n} \frac{1}{d_l}}. \tag{5.4}$$

In this equation, d_i is the distance between the given observation A^* and the rule antecedent A_i, which is calculated as the Euclidean distance between their *representative values* using Eq. 2.12.

5.1.3 Rule Interpolation

Once the rules for interpolation have been selected, rule interpolation can be performed. As introduced in Chapter 2.1, there are two main types of FRI approaches, including the resolution principle-based reasoning and analogy-based reasoning. The system proposed herein can readily work with any analogical-based reasoning approach, although it can be potentially extended to work with the direct interpolation approach, which is beyond the scope of this chapter and remains for future work. The scale and move transformation-based FRI approach is able to handle both interpolation and extrapolation, and also guarantees the uniqueness as well as the normality and convexity of the resulting interpolated fuzzy sets. Therefore, this approach is employed in this work, which has been introduced in Chapter 2.1.2, and thus the details are omitted here.

5.1.4 Rule Base Revision

A feedback mechanism is typically included in an intelligent control system to represent the system performance, which indicates the difference between the actual and desired outputs. Furthermore, a quantitative measure of the performance of a system, generally called a *performance index* (*PI*), is always considered by optimum control systems during the process of parameter configuration or adjustment. Noticing the difficulty in retrieving the accurate desired results in control systems, the feedback system is capable of indicating if the control decision works or not. This feedback is used effectively in this work to support the rule base generation and adaptation. The working progress of rule base revision is outlined in Fig. 5.2, where n denotes the number of rules in the rule base. Each of its components is elaborated on in the rest of this section.

5.1.4.1 Adding Rules

High quality interpolated rules can be reused in the future, and thus an interpolated rule that has successfully made a decision result will be added into the current rule base. In order to avoid redundant or duplicated rules, the similarity degree between the interpolated and existing rules are calculated. Suppose that the interpolated rule is R^* ('**IF** x is A^*, **THEN** y is B^* (w^*, EF^*, CD^*)'). Given a rule R_i ('**IF** x is A_i, **THEN** y is B_i (w_i, EF_i, CD_i)') in the rule base, then the degree of similarity S_i between the interpolated rule and R_i can be calculated as:

$$S_i = \frac{S(A_i, A^*) + S(B_i, B^*)}{2}, \tag{5.5}$$

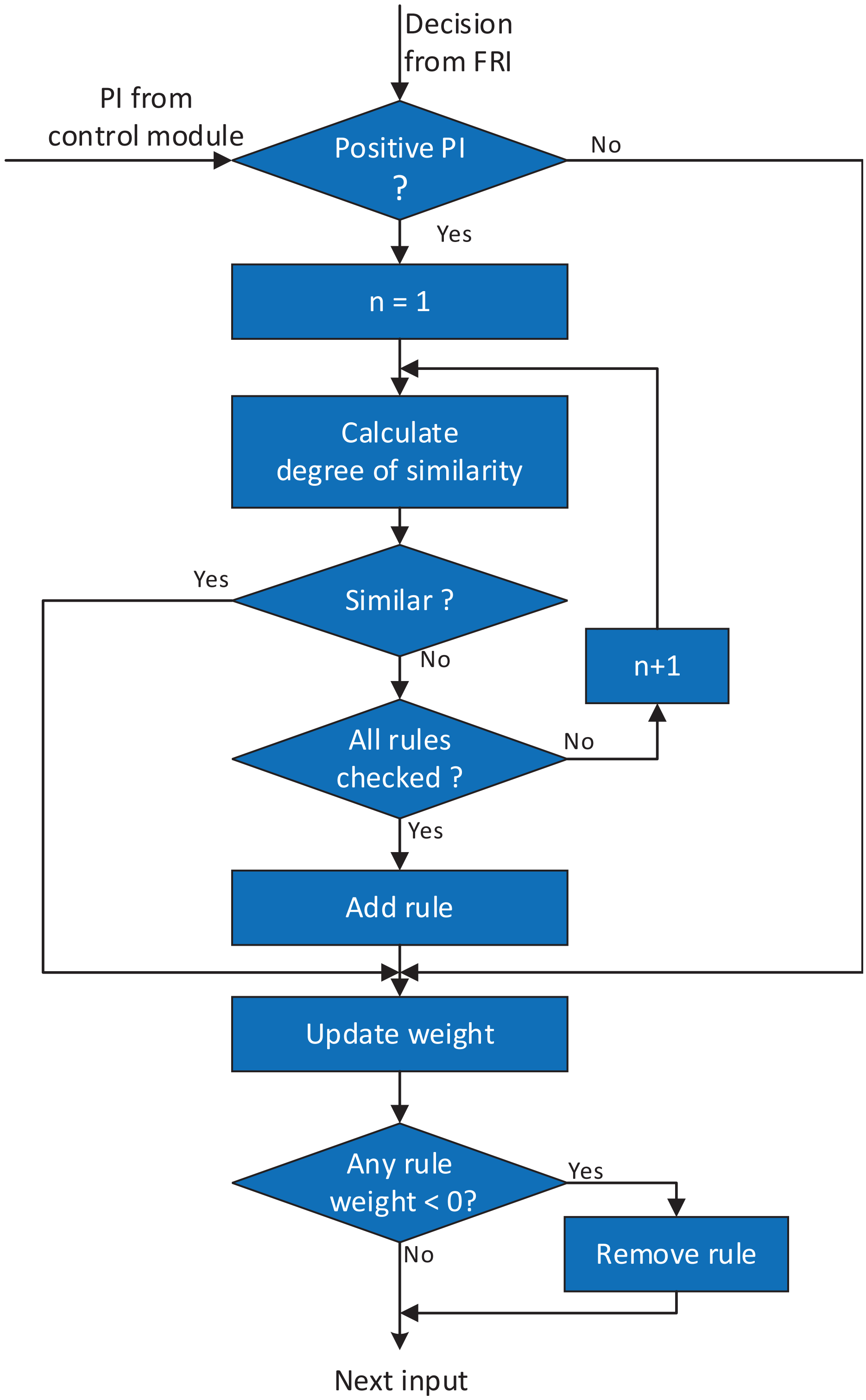


Fig. 5.2 The rule base revision procedure

where $S(A_i, A^*)$ and $S(B_i, B^*)$ are the similarity degrees between the antecedents and the consequences of the interpolated rule and R_i, respectively. The similarity degree between two fuzzy sets A_i and A^* in this work is defined as:

$$S(A_i, A^*) = 1 - \frac{|a_{i1} - a_1^*| + |a_{i2} - a_2^*| + |a_{i3} - a_3^*|}{3}. \tag{5.6}$$

The similarity between B_i and B^* can be calculated in the same way. Given a threshold, if the similarity degree between the interpolated rule and the existing rule reaches a certain threshold value, the system believes that the interpolated rule is redundant, and it will be ignored; otherwise, the interpolated rule will be added into the rule base.

Suppose that the neighbouring rules, which have been utilised to interpolate rule R^*, are R_i and R_j, ($1 \leq i, j \leq n$ and $i \neq j$), the *experience factor* EF of the new rule is computed as:

$$EF = (1 - \frac{d_i}{d})EF_i + (1 - \frac{d_j}{d})EF_j, \tag{5.7}$$

where d_i and d_j represents the distances between the observation A^* and the antecedents A_i, A_j respectively, d represents the distance between the two antecedents A_i and A_j, and EF_i and EF_j represent the current *experience factors* of rules R_i and R_j, respectively. The value of CD_i is initialised as 0.

5.1.4.2 Updating Weights

Once a decision is inferred or an interpolated rule is generated, the values of EF and CD for each rule will be updated. In particular, if a rule R_i is not employed for FRI during this step of interpolation performance, the value of CD_i will be increased by 1, and EF_i will remain the same. If the rule has been selected to perform FRI and the generated decision supports the system positively based on the *performance index*, the EF_i value will be increased by 1 for this rule and theCD_i value will be reset to 0; otherwise, if a negative *performance index* is returned, the value of EF_i will be decreased by 1 and the value of CD_i will be reset to 0. These updating operations are summarised in the table.

Table 5.1 The updating operations

Situation		**Operation**
Rule R_i Employed	Positive PI	$EF_i + 1, CD_i = 0$
	Negative PI	$EF_i - 1, CD_i = 0$
Rule R_i Not Employed		$CD_i + 1$

Based on the updated EF and CD values, the weight factor (w) of each rule will also be accordingly updated using Eq. 5.2. Given rule R_i, if the updated CD_i is equal to 0, the *cooling down factor* CD_i does not have any effect, according to Eq. 5.2. Then, the original weight factor of this particular rule will be:

$$w_i = \frac{2}{1 + e^{-\frac{EF_i}{n}}} - 1. \tag{5.8}$$

If the updated CD_i is greater than 0, which means that rule R_i has not been selected to perform interpolation for a while, the weight of this rule will be reduced by a certain percentage according to the value of the parameter CD_i. The final revised weight w_i' can be computed by:

$$w_i^{'} = w_i(1 - \frac{1}{1 + 5e^{-\frac{CD_i}{a} + b}}). \tag{5.9}$$

5.1.4.3 Removing Rules

The rule base adaptation mechanism provides a function allowing redundant or out-of-date rules to be removed from the current rule base. The judgment for the latter situation is made based on their weight factors. In particular, if the weight of rule R_i is reduced to less than 0 ($w_i < 0$), then R_i will be removed from the rule base immediately. As mentioned before, two situations may lead to the reduction of the weight of rule R_i: 1) R_i has been used, but it resulted in an incorrect decision, which is indicated by a negative *performance index*, and 2) the rule has not been used for a while, resulting in its weight fading out over time.

The interpolated result may not be accurate enough in the beginning of the deployment of the system, as the initiated rule base may not be sufficiently accurate and of sufficient rules to support FRI. Despite this, the system is still able to generate results and make decisions. However, the patterns will be adaptively learned by the system along with the performance of fuzzy interpolation, and thus the generated results will increasingly better reflect the real situation. If the current situation has changed, the decision-making system will gently adapt to the new situation by removing incorrect rules and adding new high quality interpolated rules in the rule base. Although this may take a while, the system is able to generate a new rule base to reflect the new situation in time.

5.2 Experimentation

In order to validate and evaluate the proposed automatic rule base generation and adaptation approach, a non-linear function is employed in this section to demonstrate the rule base generation functionality. Another similar yet different function is utilised to illustrate the adaptation ability of the proposed approach.

5.2.1 Rule Base Generation

Suppose the problem pattern to be modelled can be represented by the smooth curve of Eq. 5.10, shown in Fig. 5.3, where x represents the input domain, and y represents the output domain. Assume that the system inputs are vague values that are simulated by randomly generated fuzzy sets within the input domain of x. As the system inputs are usually linguistic values, in order to preserve the comprehensibility and for easy interpretation, the generated fuzzy sets for simulation are always normal and convex.

$$y = 2\sin(\frac{2x}{3}) + 5, x \in [2, 12]. \tag{5.10}$$

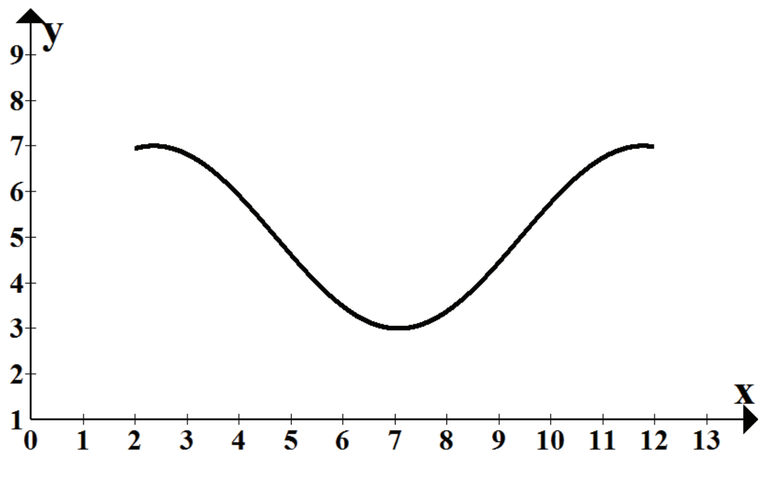


Fig. 5.3 The scenario curve to be modelled (Eq. 5.10)

The very first step to utilise the proposed modelling approach is to build an initial rule base, which is usually implemented by a very limited number of the most general rules. Theses rules may be provided by domain experts or simply be good guesses of the most typical situations and thus the initialised rules may not be very accurate. In this initial work, for simplicity, suppose 3 initial rules are available in the format of:

$$R_i: \textbf{ IF } x \textit{ is } A_i \text{ ,}\textbf{THEN } y \textit{ is } B_i \ (w_i, EF_i,\ CD_i) \ (i = \{1,2,3\}) \ . \tag{5.11}$$

where A_i and B_i are rule antecedent and consequent, which both are convex and normal triangular fuzzy sets that can be represented as $A_i = (a_{i1}, a_{i2}, a_{i3})$ and $B_i = (b_{i1}, b_{i2}, b_{i3})$. The corresponding parameters are listed in Table 5.2. In particular, EF_i, $i = \{1,2,3\}$, is initially

Table 5.2 The initialised rule base

i	A_i (a_{i1}, a_{i2}, a_{i3})	B_i (b_{i1}, b_{i2}, b_{i3})	w_i	EF_i	CD_i
1	$(2.00, 2.50, 3.00)$	$(6.50, 7.00, 7.50)$	0.12	50	0
2	$(6.50, 7.00, 7.50)$	$(2.50, 3.00, 3.50)$	0.12	50	0
3	$(11.00, 11.50, 12.00)$	$(6.50, 7.00, 7.50)$	0.12	50	0

configured as 50 (based on brute force trying), and then the weight of each rule is accordingly initialised as $w_1 = w_2 = w_3 = 0.12$, as shown in Table 5.2. In the implementation of this work, the weight factor calculation function is as follows:

$$w_i = (\frac{2}{1+e^{-\frac{EF_i}{200}}} - 1)(1 - \frac{1}{1+5e^{-\frac{CD_i}{100}+5}}). \tag{5.12}$$

For better illustration, the generated rule bases are visualised. The Fig. 5.4 visualised the initialised rule base, where x and y indicate the input and output domain, respectively. And each dot, shows as $+$ in the figure, represents one fuzzy rule in the rule base. In particular, the x coordinate of the dot is the *representative value* of its rule antecedent, which can be obtained by Eq. 2.12, and the y coordinate of the dot indicates the *representative value* of the corresponding consequent of the same rule. Taken R_1, listed in the Table 5.2, as an example, the rule antecedent is $A_1 = (2.00, 2.50, 3.00)$, its *representative value* can then be computed by Eq. 2.12, which $Rep_{A_1} = (2.00 + 2.50 + 3.00)/3 = 2.5$. The consequent of R_1 is $B_1 = (6.50, 7.00, 7.50)$, and its *representative value* is $Rep_{B_1} = (6.50 + 7.00 + 7.50)/3 = 7.0$. Therefore, the R_1 can be represented by a dot $(+)$ in a 2-D plane, which its x coordinate is 2.50 and its y coordinate is 7.0, as shown in Fig. 5.4. Note that this representation method is also applied to the rest of rule base visualisations in this section.

Once the rule base is initialised, the system is able to take randomly generated inputs in the input domain to generate the outputs. After performing 50 interpolation inferences, the

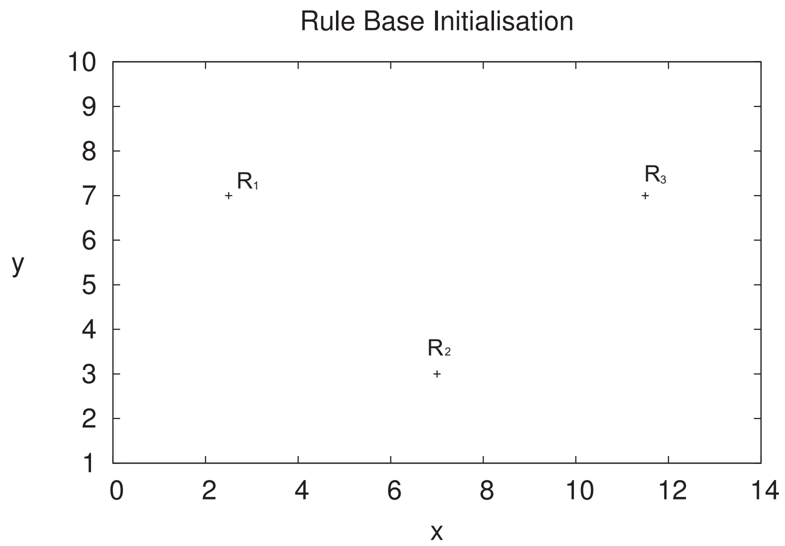


Fig. 5.4 The initial rule base

rule base includes eight rules (denoted as $R_i, i \in \{1, 2, ..., 8\}$), as shown in Fig. 5.5, and listed in Table 5.3. From this, the next step of rule generation is elaborated on as follows:

Table 5.3 The evolved rule base used in the example

i	A_i (a_{i1}, a_{i2}, a_{i3})	B_i (b_{i1}, b_{i2}, b_{i3})	w_i	EF_i	CD_i
1	$(2.00, 2.50, 3.00)$	$(6.50, 7.00, 7.50)$	0.057	23	10
2	$(3.98, 4.38, 5.29)$	$(4.73, 5.13, 5.74)$	0.049	19	41
3	$(4.53, 4.87, 5.06)$	$(4.68, 4.88, 5.02)$	0.033	13	20
4	$(4.97, 5.58, 7.39)$	$(2.83, 3.37, 4.30)$	0.050	20	20
5	$(6.50, 7.00, 7.50)$	$(2.50, 3.00, 3.50)$	0.129	53	0
6	$(9.23, 10.11, 10.54)$	$(4.81, 5.32, 5.59)$	0.061	24	41
7	$(10.50, 11.00, 11.50)$	$(5.90, 6.50, 6.70)$	0.105	43	0
8	$(11.00, 11.50, 12.00)$	$(6.50, 7.00, 7.50)$	0.087	35	10

Step 1 System input: For each individual step of inference performance, the system starts from taking an observation as system input. In this example, assume that an observation $A^* = (7.00, 7.30, 9.00)$ is given as system input.

Step 2 Rule selection: Based on the given observation and the current rule base, the system calculates the importance factor of each rule regarding the given input by Eq. 5.3. The details of the intermediate and final results of the calculation are shown in Table 5.4. According to the calculated result, the rule with the greatest importance factor on each side of the given observation will be selected to perform FRI. In this particular example, rules

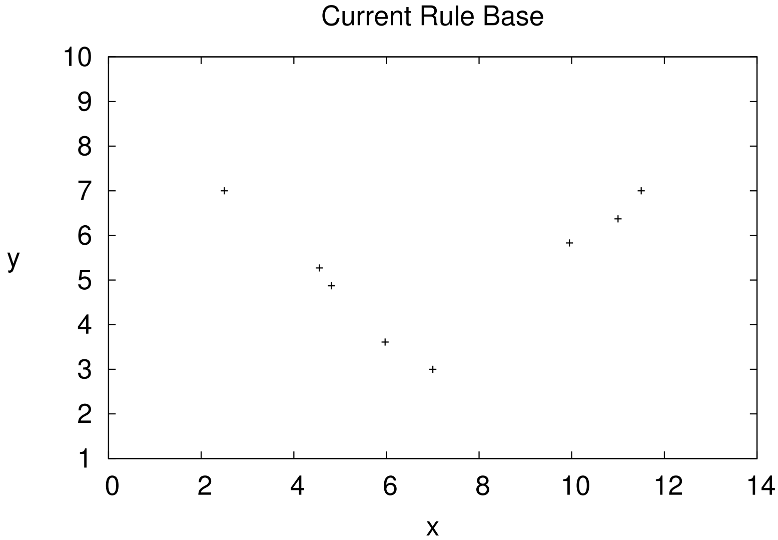


Fig. 5.5 The evolved rule base used in the example

R_5 and R_7 are selected by the proposed approach rather than the closest rules R_5 and R_6, according to the existing FRI approaches.

Table 5.4 The calculation for rule selection

i	d_i	$\lambda_i^{'}$	w_i	EF_i	CD_i	IF_i	Left rule or Right rule?
1	5.27	0.05	0.057	23	10	0.0129	L
2	3.22	0.08	0.049	19	41	0.0142	L
3	2.95	0.09	0.033	13	20	0.0100	L
4	1.79	0.15	0.050	20	20	0.0194	L
5	0.77	0.35	0.129	53	0	0.0764	L
6	2.19	0.12	0.061	24	41	0.0212	R
7	3.23	0.08	0.105	43	0	0.0303	R
8	3.73	0.07	0.087	35	10	0.0234	R

Step 3 Transformation-based FRI: When the two rules for interpolation are determined with regard to the given observation, the HS approach [79, 80], which is a transformation-based fuzzy rule interpolation method, is employed to generate the inference result. Firstly, based on the values of rule antecedents and the given observation, the relative placement factors (λ) is calculated, which is $\lambda = 0.07$. Secondly, the scale rate is obtained as $s = 0.5$. Then, the move ratio is calculated: $m = 0.7$. Finally, the interpolation result is achieved as

$B^* = (2.76, 3.26, 3.74)$. After the defuzzification using the centre of gravity principle, the crisp result $B^* = 3.20$ is generated as system output.

Step 4 Performance index: Based on the given observation ($A^* = (7.00, 7.30, 9.00)$, $A^* = 7.7$ after defuzzification) and the simulated model, as entailed in Eq. 5.10, the desired result should be 3.20. As the system output is equal to the desired output, a positive *performance index* is returned. Note that the desired result is usually not available or obtainable in most of the control systems at any stage, but it is common that a *performance index* is available after the interpolated result has been utilised. The *performance index* clearly indicates if the interpolated result was acceptable or not. Next, the rule base is then updated according to the value of the returned *performance index*.

Step 5 Rule base updating: The rules R_5 and R_7 were used to generate the interpolated result, which supports the system running correctly, and as a result, the *experience factor* and *cooling down factor* of both rules will be accordingly updated. Although the rest of the rules in the rule base were not selected to perform this particular FRI, their *CD* will also be updated. The updating operations of the current rule base is shown in Table 5.5.

Table 5.5 The operations of rule base updating

i	A_i	B_i	w_i	EF_i	CD_i
1	$(2.00, 2.50, 3.00)$	$(6.50, 7.00, 7.50)$	$0.057 \rightarrow 0.057$	23	$10 \rightarrow 11$
2	$(3.98, 4.38, 5.29)$	$(4.73, 5.13, 5.74)$	$0.049 \rightarrow 0.045$	19	$41 \rightarrow 42$
3	$(4.53, 4.87, 5.06)$	$(4.68, 4.88, 5.02)$	$0.033 \rightarrow 0.032$	13	$20 \rightarrow 21$
4	$(4.97, 5.58, 7.39)$	$(2.83, 3.37, 4.30)$	$0.050 \rightarrow 0.049$	20	$20 \rightarrow 21$
5	$(6.50, 7.00, 7.50)$	$(2.50, 3.00, 3.50)$	0.129	$53 \rightarrow 54$	0
6	$(9.23, 10.11, 10.54)$	$(4.81, 5.32, 5.59)$	$0.061 \rightarrow 0.056$	24	$41 \rightarrow 42$
7	$(10.50, 11.00, 11.50)$	$(5.90, 6.50, 6.70)$	0.105	$43 \rightarrow 44$	0
8	$(11.00, 11.50, 12.00)$	$(6.50, 7.00, 7.50)$	$0.087 \rightarrow 0.087$	35	$10 \rightarrow 11$

Due to the positive performance of the interpolation inference, the interpolated rule will be added into the rule base as a new rule for future use, unless a similar rule already exists in the current rule base. The degree of similarity between each of the existing rules and the interpolated rule can be calculated using Eq. 5.5 and 5.6, and the results are summarised in Table 5.6. A negative similarity degree indicates that neither the antecedents of two compared rules nor the consequences of them overlap with each other. Define the similarity threshold as 0.7 in this work. It is clear from this table, that no similar rule exists in the current rule base regarding the interpolated rule; that is, the degree of similarity between every rule and the interpolated rule is less than 0.7. Therefore, the interpolated rule is added into the current rule base as the R_9. The *experience factor* of this new rule EF_9 is calculated using Eq. 5.7, and the cooling down value CD_9 is set as 0. The details of this rule are shown in Eq. 5.13.

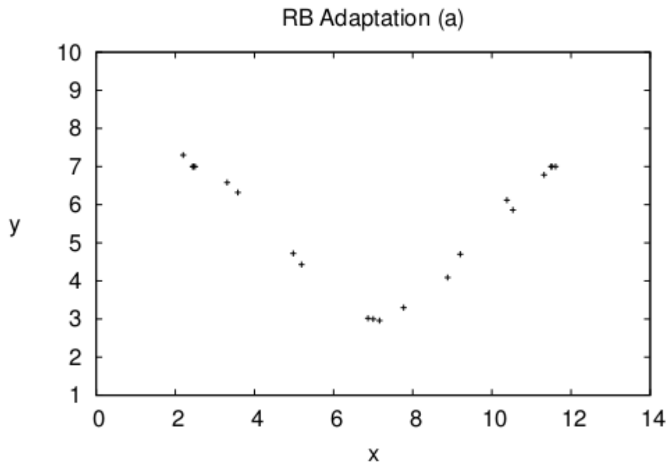
$$\begin{aligned} \mathbf{R_9:IF} \quad & x = A^* = (7.00, 7.30, 9.00) \\ \mathbf{THEN} \quad & y = B^* = (2.76, 3.26, 3.74)(0.324, 52, 0) \end{aligned} \tag{5.13}$$

Table 5.6 Similarity degree between existing rules and interpolated rule

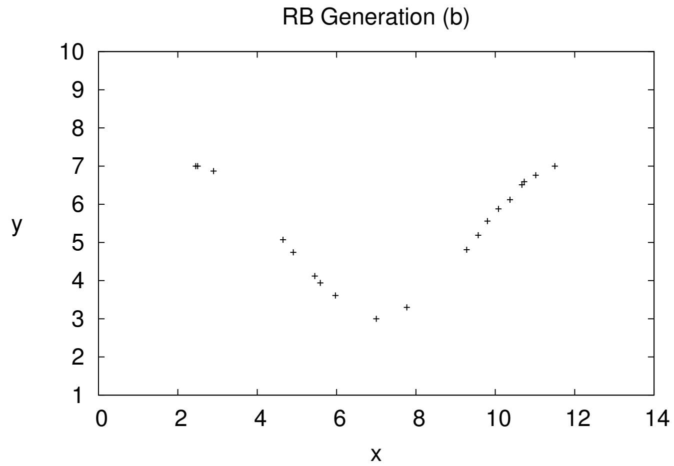
	Rule1	Rule2	Rule3	Rule4	Rule5	Rule6	Rule7	Rule8
Similarity	-2.74	-1.01	0.61	0.65	0.67	-1.54	-2.11	-2.74

The above exemplar interpolation step demonstrates the situation where a satisfied result is generated by FRI and the interpolated rule is added into the rule base. When a negative *performance index* is returned, the system will work differently. For instance, the next observation is $A^* = (6.50, 8.10, 9.30)$. Then, rules R_5 and R_7 are selected to perform FRI instead of the closest rules R_5 and R_6. From this, the generated result $B^* = (3.44, 3.97, 4.39)$ is interpolated, which results in $B^* = 3.93$ after defuzzification. This system output is quite different to the desired value which is 3.36; thus, a negative *performance index* will be returned. Consequently, this interpolated rule will be ignored, and the *experience factors* of these two rules will be decreased by 1, as punishment. Of course, the *experience factor*s and *cooling down factors* of all other rules will also be updated based on Table 5.1, with the details omitted here to save space.

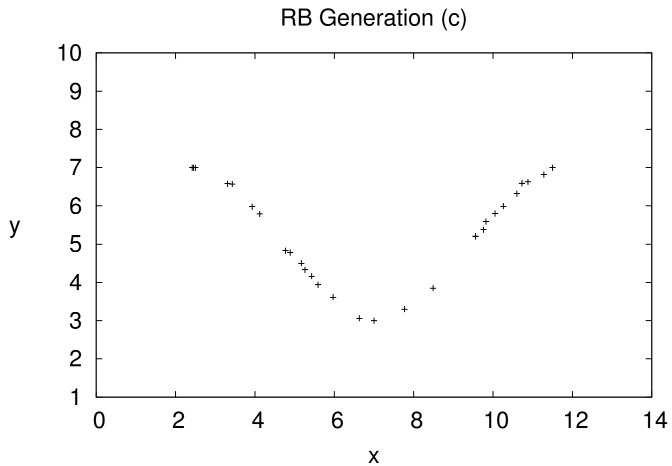
The system repeats the above process for every new input, and it will be stabilised after a number of performance iterations. Particularly for the given example, the system rule base becomes stable after 3,000 inference performances, resulting in a rule base with 36 rules. The evolvement of the rule base for the running example is illustrated in Fig. 5.6.



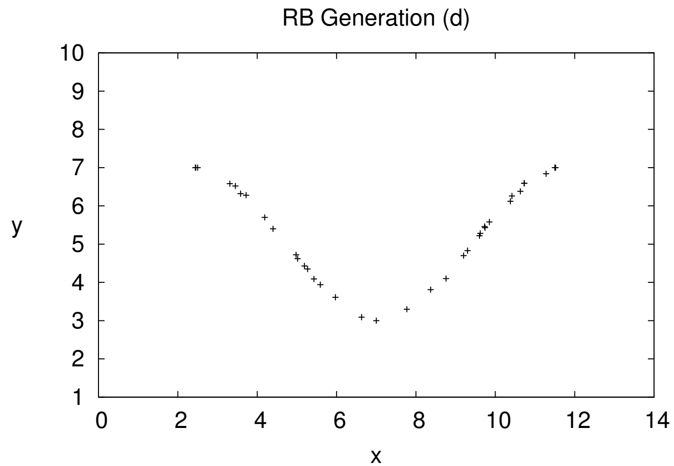
(a) Rule base after 1000 interpolation performance



(b) Rule base after 2000 interpolation performance



(c) Rule base after 2600 interpolation performance



(d) Rule base after 3000 interpolation performance

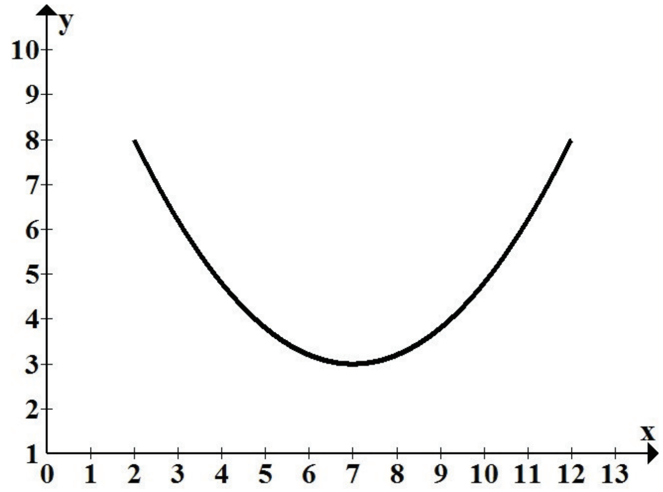
Fig. 5.6 The processes of rule base generation

5.2.2 Rule Base Adaptation

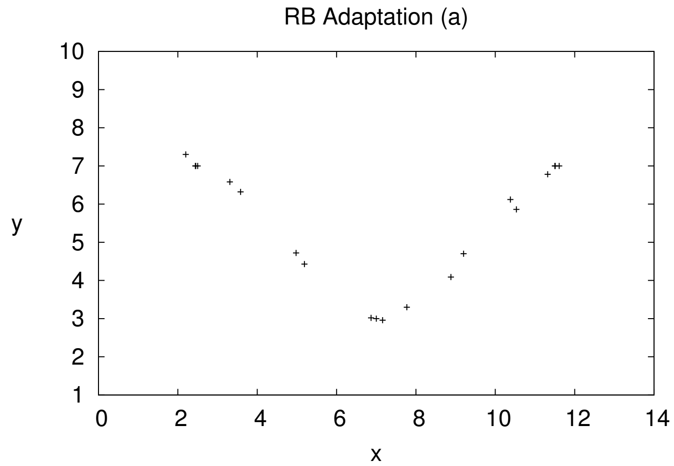
The proposed automatic rule base generation and adaptation approach is not only able to learn the model pattern whilst performing interpolation inference, but is also able to adapt the current rule base to a changed model pattern. In order to demonstrate how the proposed system handles the changes of the underlying model pattern, assume the ground truth model pattern has changed from the pattern shown in Fig. 5.3 to that shown in Fig. 5.7(a) (corresponding to Eq, 5.14), after 3,000 interpolation performances by which the rule base is illustrated in Fig. 5.6(d).

$$y = \frac{(x-7)^2}{5} + 3, x \in [2, 12]. \tag{5.14}$$

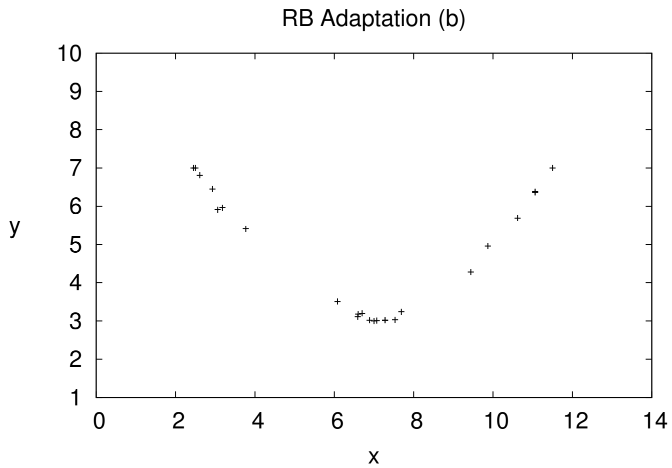
Due to the change of the underlying pattern to be modelled, the previous rule base will not be able to generate satisfying results from time to time. Therefore, the weight factor of some of the existing rules will be dramatically decreased and they will be gradually removed along with the performance of interpolation inferences. Of course, if a positive *performance index* is returned regarding a certain step of interpolation inference, this particular rule will be added into the rule base. These operations are exactly the same as the ones introduced above, and thus the details are omitted here. The overall evolvement progress of the rule base is illustrated in Fig. 5.7. Note that the rule base adaptation process will keep running and not be terminated itself, although the generated rule base is able to accurately reflect the given problem. The reason is that the system has to keep itself active in order to detect the changing situation. As the consequence, the rules, which can correctly represent the problem, may be removed if they have not been involved in fuzzy interpolative for a long time period. It is a limitation of the proposed system. Therefore, the termination condition of the rule base adaptation progress has to be considered, which will remain as the future work.



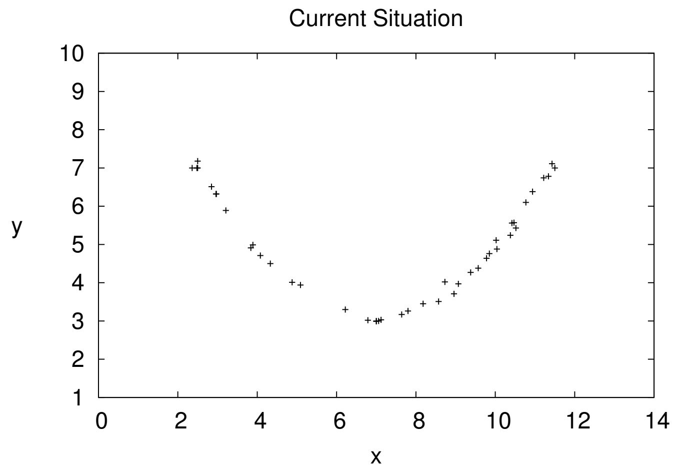
(a) A changed pattern to be modeled



(b) Rule base adaptation after 3500 of interpolation inferences



(c) Rule base adaptation after 4200 of interpolation inferences



(d) Rule base adaptation after 5000 of interpolation inferences

Fig. 5.7 The processes of rule base adaptation

5.2.3 Discussion

Two non-linear mathematical models have been employed for the proposed system validation. In particular, the first model demonstrated the functionality of the rule base generation, and the second model validated the ability of the adaptation of the proposed system.

In the first experiment, the system started from only three initialised rules. The 'best' two rules regarding to the given input were selected from the current rule base for interpolation. The rule bases, which were generated at the different stages while the performing, e.g. 50, 1000, 2000, 2600, and 3000 iterations, were visualised. Those visualised rule bases confirmed that the proposed system is able to automatically generate a rule base from very limited a priori knowledge to support a specific problem.

In the second experiment, the system started from the rule base, which has been generated from the first experiment. However, the problem model has been changed. The generated rule bases at the different stages while the performing were also visualised, particularly, at 500, 1200, and 2000 iterations. The visualised rule bases clearly illustrated how the proposed system led to the change of the rule base to fit the changed situation. That confirmed the ability of the adaptation of the proposed method.

The experimental results show that the proposed system is able to adaptively generate the rule base and revise it whenever the underlying model has changed. Therefore, this system may provide solutions for some real-world problems, such as smart home control system. FRI has been successfully employed to the smart home heating management systems, such as [14], where the rule base was predefined based on the historic data of a particular property and residents and thus it is only able to deal with fixed per-defined situations. Two benefits will take place if the proposed system is applied. Firstly, only the most general/common a priori knowledge is required to initialise the system, thus the heating management system can be mass-produced (commercialised). Secondly, the system is able to handle changing situation such as change of radiators, residents, or their living styles, thus the model is highly adaptable.

5.3 Summary

This chapter presented a novel rule base generation and adaptation approach for FRI, which is able to adaptively generate and revise the rule base with limited training data and/or expert knowledge for control problems, as long as a *performance index* is available to indicate if the inference result is acceptable or not. In particular, the system initialises the rule base with very limited rules first. Then, based on the existing rules' usage frequency information, historic performance data and distances between observation and existing rules, the best two rules are selected for FRI, rather than the two closest neighbouring rules as implemented in existing FRI approaches. Finally, the rule base is adaptively generated and revised, which is guided by the *performance index* of the interpolated result and the performance experiences of rules. The simulation experimentation suggests that the proposed system is able to automatically generate and adapt rule bases to enhance FRI.

Chapter 6

Intelligent Home Heating Control by Fuzzy Rule Interpolation

The advance of smart home appliance control plays a key role in improving current environmental issues by reducing unexpected domestic energy wastes, which have greatly contributed to excessive carbon emissions. Typical domestic energy wastes include heating unoccupied homes, washing very few clothes for too long, unnecessary low temperatures for fridges and freezers, and lighting unoccupied rooms [70]. Noticing that home heating uses more energy than any other residential energy expenditure, including air conditioning, water heating, and appliances [58], research on the reduction of heating unoccupied homes is of great importance. Human motion sensors have been commonly used in home heating control to detect if a home is occupied or not. If the home is not occupied, the heating system will usually be turned off or kept at a minimum temperature, and otherwise the system will be on.

Two important drawbacks have limited the wide application of sensor-based home heating controllers. Firstly, residents may suffer from low temperature if heating systems are simply controlled by motion sensors. This is because it takes time to heat the home to a certain comfortable temperature and it can be very cold when the residents have just arrived home. Another important limitation of the sensor-based systems is that the sensors can only be triggered by human activities, which means they can only deal with situations where users are within the sensor's coverage. In order to address this, a number of programmable controllers for central heating systems have been proposed in the literature [73, 120, 153, 167, 189], which are usually developed based on the assumption that residents in a property have a fixed and simple living pattern. These systems have gained different levels of success by providing some intelligence to control home heating systems to minimise the wastes of heating unoccupied homes.

Fundamentally, the existing programmable heating controllers can be grouped in to two classes, which are schedule-based and learning-based [89]. Schedule-based controllers require pre-configured timetables made by home users [153, 154]. The heating system follows exactly the scheduled time, which is independent from the user's current situation. As a consequence, the system will waste domestic energy by heating an unoccupied home when no resident is there or close to getting home. Also, the residents may suffer if they arrive home earlier than scheduled. In this case, they may manually turn the heating system on, but they will still suffer for the period of time before the home is properly heated as it takes time to heat the home to a comfortable temperature. Learning-based controllers are able to automatically make a heating schedule by learning the user's habits, such as a satisfying room temperature and a regular daily routine for a period of time [120, 167]. Different from the schedule-based controllers, this type of controller requires a number of sensors to detect human activities, and thus this system can be seen as a combination of a sensor-based and a schedule-based controller, which enjoys the advantages of both. However, it still cannot solve the problem of preheating homes for the situations of getting home earlier than the schedule.

This chapter proposes a novel smart home heating control system by efficiently utilising the personal data captured in smart portable devices. It is able to successfully preheat the home by predicting when the residents will arrive there and thus to address the limitation of the existing heating controllers by using it as a complementary part of the existing ones. In particular, the knowledge-driven approach is adopted first to generate a rule base based on the Mamdani fuzzy model. And then, a fuzzy interpolation technique is selected here to reduce the complexity of the generated fuzzy model by omitting those rules in the fuzzy rule base which can be approximated by their neighbours. The proposed system has been applied to a real-world case and a promising result has been generated.

The rest of the chapter is structured as follows. Chapter 6.1 presents the proposed intelligent home heating system in detail. Chapter 6.2 applies the proposed system to a real-world case for demonstration and validation. Chapter 6.3 concludes the chapter.

6.1 Smart Home Heating Controller

Most UK houses have a central heating system that uses a boiler to heat the water supply for heating. The boiler does not usually provide a user-friendly interface to adjust the output power, and thus the easiest way to adjust the room's temperature is to control the time duration of the boiler burning. The proposed smart home heating controller in this work only concerns this type of house heating systems by deciding if the boiler should be on or off, though it can be readily extended for other types.

6.1.1 The Framework

A number of smart home heating controllers [73, 120, 153, 167, 189] have been developed to deal with the situations when the resident is at home; this work, therefore, only focuses on the development of decision-making when a home user is away. This is achieved by predicting if the heater should be turned on to preheat the home such that the home temperature can reach a certain comfortable level when the resident arrives home. Whether the resident is at home or not can be detected by checking if the resident's smart portable device is connected to the home Wi-Fi. The decision-making procedure is triggered once the resident's portable device is disconnected from the home Wi-Fi, and it terminates when the resident arrives home. The flow chart of the decision-making procedure for the proposed home heating controller is illustrated in Fig. 6.1.

The controller first extracts the resident's location and moving information. There are four types of residents' location and moving information that need to be considered: At Home, Way Back Home, Leaving Home, and Static (i.e. at Special Location). The user's current location and moving states are obtained effectively using the GPS information provided by the user's portable devices. From this, if the resident's current state is At Home, the algorithm terminates; and if the resident's current state is Leaving Home, that is, the resident is moving away from home, the boiler is off and the system will check the resident's location and moving information again in a certain period of time. Otherwise, the time until arriving home (denoted as T_{AH}) is predicted and the time to preheat the home to a comfortable temperature (denoted as T_{PH}) is also calculated, based on the resident's current situation and the current environment around the home. If T_{AH} is not greater than T_{PH}, the boiler will be turned on and the system will check this again in a certain period of time. The details of the important steps in this procedure are explained below.

6.1.2 Location Information Processing

As stated earlier, there are four different situations that a user is normally in, based on the location data. In particular, the situation of static represents that the user is either stuck in traffic or at some special locations for particular activities, such as shopping in a supermarket or dinning in a restaurant. In order to obtain the user's current state, a comparison algorithm, as shown in Fig. 6.2, was developed to verify which situation the user currently belongs

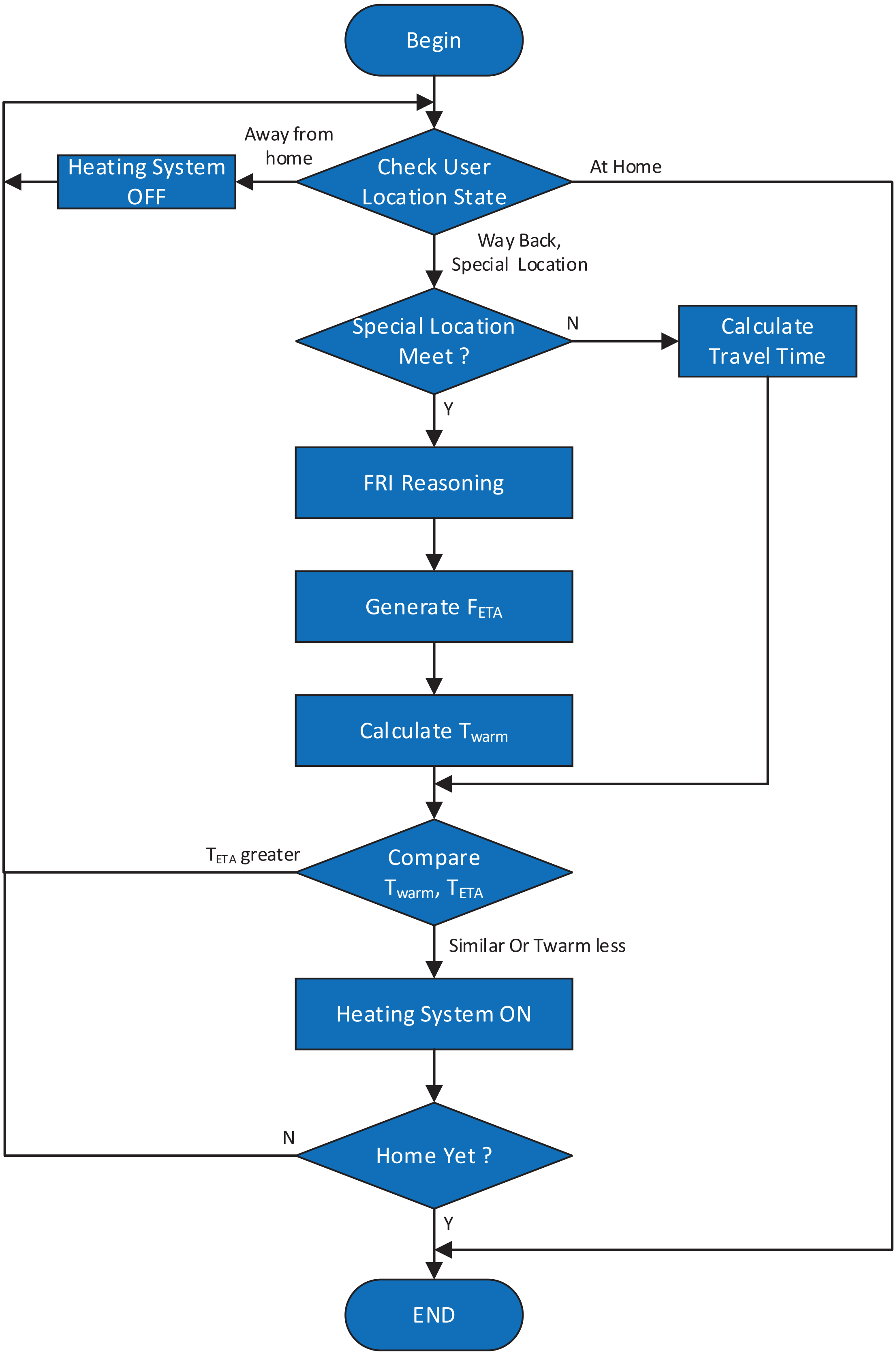


Fig. 6.1 The overall work flow

to. In the algorithm, the *Google Distance Matrix API* was employed here to calculate the real travel distance and time duration between a user's current location and home. As different travel modes, including driving, walking and bicycling, will result in different travel distances and times, the naive Bayes classifier [66] is applied to the current and past location information to obtain the user's current travel modes. Then the detected travel mode is used as a parameter that passed to *Google Distance Matrix API* to estimate travel distance and time. The distance and duration will be recorded continuously based on an adjustable time interval to determine the user's current state. In particular, when the current distance from home is continuously φ times greater than the previous ones, where $\varphi \in \mathbb{N}$ is a adjustable parameter can be adjusted based on the different situations, the user state will be classed as Away From Home. Similarly, if the current distance from home is continuously three times less than the previous ones, it is believed the resident is on their Way Back Home. Otherwise, if the past four captured distances are roughly equal to each other, the system will then acknowledge that the user is under a special event. The At Home state can be verified based on the condition that either the distance is zero or the user's mobile device is connected to the home Wi-Fi.

The times spent on different locations can vary significantly, and also different residents usually spend different amounts of time at the same special location as people have their own living styles or patterns. In order to predict the time that a particular resident is most likely to spend at a special location, the resident's historical GPS data can help. The historical GPS data can either be stored on smart portable devices, such as a mobile phone, or it can be deliberately captured for the proposed system. Once the historical GPS data is obtained, data mining techniques can be used to extract the time spending information for different types of locations. For simplicity, this work assumes that the resident only goes shopping on their way back home, and all residents spend the same amount of time at the same types of stores. *Google Place API* is applied herein to achieve the home user's current location. A systematic study of the time spent in different types of locations based on the demographic classification may greatly improve the performance of the controller, which remains as future work.

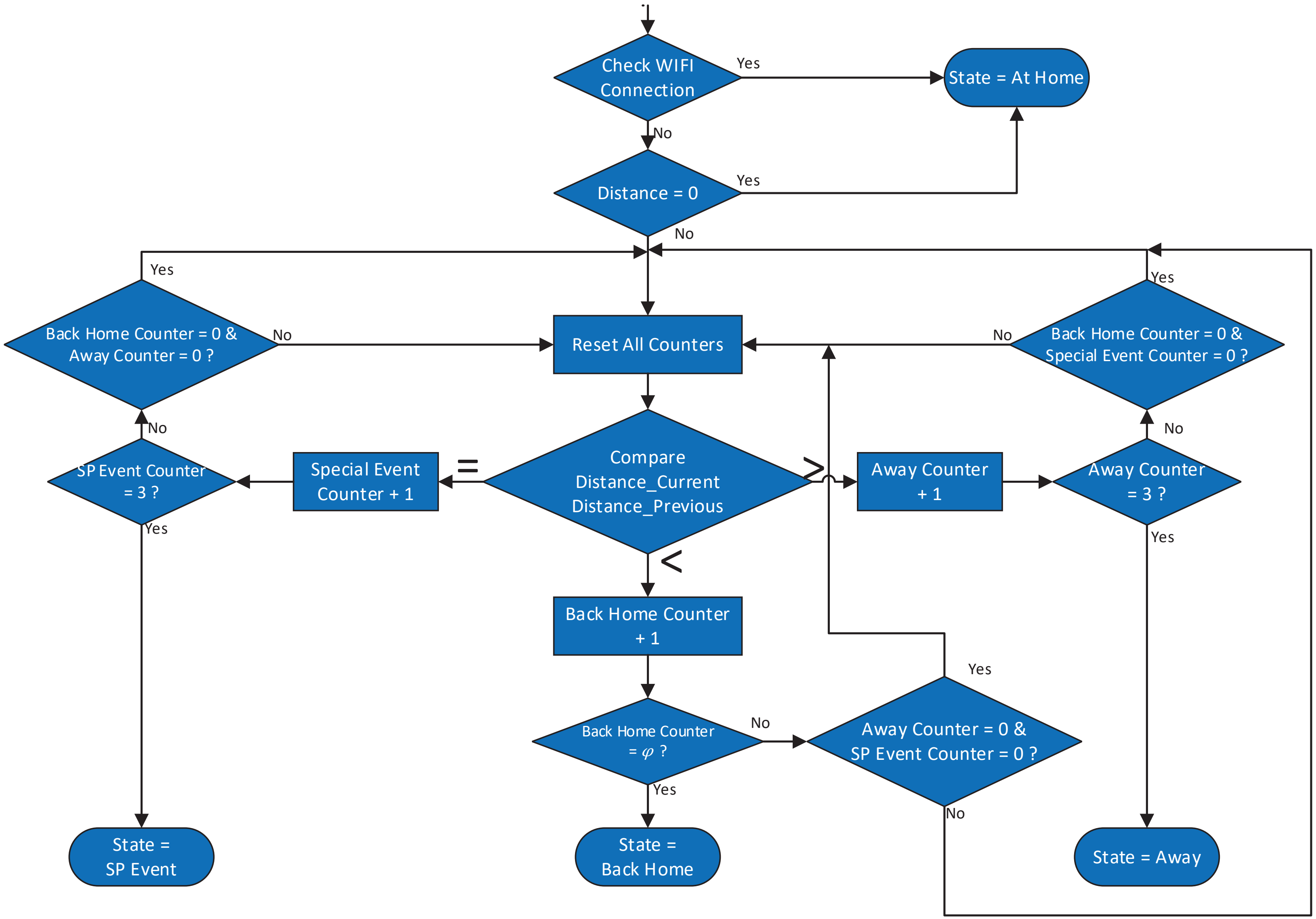


Fig. 6.2 Comparison algorithm

6.1.3 Time to Home Estimation

Due to the difficulty in modelling the resident's behaviour using traditional mathematical modelling led by uncertainty and complexity, fuzzy inference systems are employed in this work to predict the time duration before a resident gets home. By using fuzzy inference systems, a resident's behaviour pattern can be modelled as fuzzy rules, which can be readily transferred from natural language description. The fuzzy inference engine takes five fuzzy inputs and produces one fuzzy output which is the estimate of the time to get home.

$x_{Location}$: The time spent in stores will directly affect the time to get home. In this work, the Tesco store system is used to represent different types of store: Superstore, Extra, Metro, and Express, which are represented as triangular fuzzy sets. In other words, the four types of Tesco stores are used in this system to represent the domain of variable location.

x_{Days} and x_{Time}: Time spent at the same location will normally be different during different times of the day, and different days of the week. For example, people usually spend longer at a supermarket at the weekends than on week-days. Seven fuzzy sets have been pre defined to partition the domain of the variable x_{Days} to represent seven days a week (Monday to Sunday), and 13 fuzzy sets are defined for the domain of the variable x_{Time} to represent a day.

x_{Spent}: The amount of time that has already been spent in the location is another factor to affect the time of travel to home. In this work, 13 triangle fuzzy sets are designed for this input variable to represent real-time values between 0 and 60 minutes. It is simply fuzzified from crisp data, which are obtained by personal portable devices.

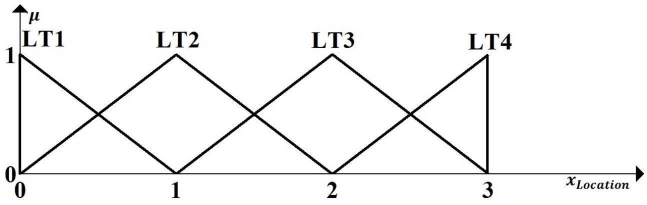
x_{Travel}: This input variable is used to identify the travel time between a user's current location and home based on the current traffic situation. It is fuzzified from crisp data, which is returned by the *Google Distance Matrix API*. 13 fuzzy sets are designed for this fuzzy input variable to represent between 0 and 60 minutes, as most of the properties can be preheated within an hour.

x_{AH}: The proposed fuzzy inference engine is to predict the time span from the moment the prediction is made to the resident getting home. This variable domain is partitioned by 13 different linguistic values which represent the time period of 0 to 120 minutes. During defuzzification, the linguistic values (T_{AH}) are converted to a crisp value using the centre of gravity. The domain partition of input and output variables are shown in Fig. 6.3.

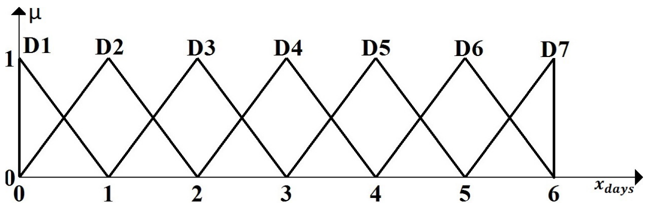
Based on the above description, 61,516 rules are needed to fully cover all the situations. In order to simplify the system and to preserve the transparency of fuzzy inference systems, fuzzy rule interpolation is employed in this work thanks to its ability to reduce system complexity by omitting those rules which can be represented by their neighbours. Therefore, 72 of the most important rules have been selected for fuzzy rule interpolation, which are listed in Table 6.1. In this table, *LT* represents Location Types; *D* represents Days; *TD* represents Times of Day; *TSP* represents Time Spent; *TTV* represents Travel Time;, and *AH* represents the Final Decision AH. The numbers in the table represent the corresponding fuzzy variables, as illustrated in Fig. 6.3. Given the robustness and generality of the scale and move transformation-based fuzzy rule interpolation approach, it is employed in this work to predict the time of getting home.

Table 6.1 FRI rules

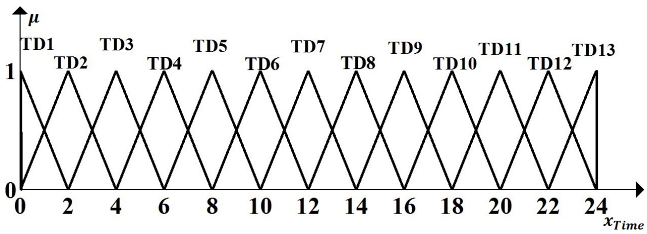
No.	I F					THEN	No.	I F					THEN
	LT	D	TD	TSP	TTV	AH		LT	D	TD	TSP	TTV	AH
1	1	1	1	1	1	**2**	**37**	4	1	1	1	1	**5**
2	1	1	1	1	13	**8**	**38**	4	1	1	1	13	**13**
3	1	1	1	13	1	**1**	**39**	4	1	1	13	1	**1**
4	1	1	1	13	13	**7**	**40**	4	1	1	13	13	**7**
5	1	1	10	1	1	**3**	**41**	4	1	10	1	1	**6**
6	1	1	10	1	13	**9**	**42**	4	1	10	1	13	**12**
7	1	1	10	13	1	**1**	**43**	4	1	10	13	1	**1**
8	1	1	10	13	13	**7**	**44**	4	1	10	13	13	**7**
9	1	1	13	1	1	**2**	**45**	4	1	13	1	1	**5**
10	1	1	13	1	13	**8**	**46**	4	1	13	1	13	**13**
11	1	1	13	13	1	**1**	**47**	4	1	13	13	1	**1**
12	1	1	13	13	13	**7**	**48**	4	1	13	13	13	**7**
13	1	5	1	1	1	**3**	**49**	4	5	1	1	1	**5**
14	1	5	1	1	13	**9**	**50**	4	5	1	1	13	**13**
15	1	5	1	13	1	**1**	**51**	4	5	1	13	1	**1**
16	1	5	1	13	13	**7**	**52**	4	5	1	13	13	**7**
17	1	5	10	1	1	**4**	**53**	4	5	10	1	1	**7**
18	1	5	10	1	13	**10**	**54**	4	5	10	1	13	**13**
19	1	5	10	13	1	**1**	**55**	4	5	10	13	1	**1**
20	1	5	10	13	13	**7**	**56**	4	5	10	13	13	**7**
21	1	5	13	1	1	**3**	**57**	4	5	13	1	1	**5**
22	1	5	13	1	13	**9**	**58**	4	5	13	1	13	**13**
23	1	5	13	13	1	**1**	**59**	4	5	13	13	1	**1**
24	1	5	13	13	13	**7**	**60**	4	5	13	13	13	**7**
25	1	7	1	1	1	**2**	**61**	4	7	1	1	1	**5**
26	1	7	1	1	13	**8**	**62**	4	7	1	1	13	**13**
27	1	7	1	13	1	**1**	**63**	4	7	1	13	1	**1**
28	1	7	1	13	13	**7**	**64**	4	7	1	13	13	**7**
29	1	7	10	1	1	**3**	**65**	4	7	10	1	1	**6**
30	1	7	10	1	13	**9**	**66**	4	7	10	1	13	**12**
31	1	7	10	13	1	**1**	**67**	4	7	10	13	1	**1**
32	1	7	10	13	13	**7**	**68**	4	7	10	13	13	**7**
33	1	7	13	1	1	**2**	**69**	4	7	13	1	1	**5**
34	1	7	13	1	13	**8**	**70**	4	7	13	1	13	**13**
35	1	7	13	13	1	**1**	**71**	4	7	13	13	1	**1**
36	1	7	13	13	13	**7**	**72**	4	7	13	13	13	**7**



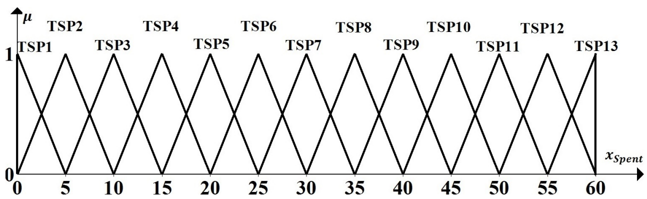
(a) Input-Location types



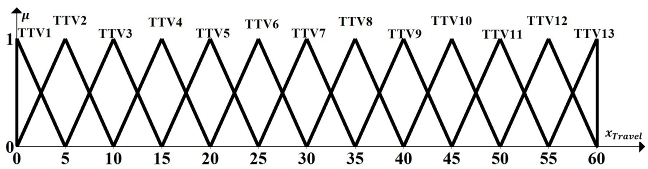
(b) Input-Days



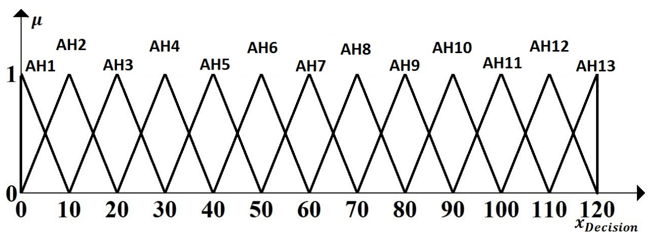
(c) Input-Times of day



(d) Input-Time spent



(e) Input-Travel times



(f) Output-Decision AH

Fig. 6.3 Fuzzy variables

6.1.4 Preheat Time Calculation

The rate of home temperature increasing (i.e. the time it takes for the house to heat up by 1°C) is not linear and several factors can affect it, such as the weather, the outside temperature, the efficiency of a radiator, the insulation of walls, and the output power of the home boiler. A higher current inside temperature will lead to a much longer increasing rate. For simplicity, this work assumes that the boiler, radiators, and the wall insulation are selected and installed based on the UK standard, and other factors such as the weathers and the outside environment will not be considered in this work. This assumption has been commonly used in the literature. For example, a 'heating gain table' has been created based on the collected temperature data in a house over three days in the work of [167]. This work adopts a similar approach and creates a heating gain table based on a four-bedrooms detached house with a total floor area of 100 m^2 and a ceiling height of 2.4 metres. The heating gain table, as shown in Table 6.2, was created based on the collected data over three days by employing the average principle.

Table 6.2 Heating gain table

Temperature Range	**Time Required**
Boiler initialise / pre heat	5 Minutes
15°C-16°C	8 Minutes
16°C-17°C	8 Minutes
17°C-18°C	8 Minutes
18°C-19°C	12 Minutes
19°C-20°C	17 Minutes
20°C-21°C	28 Minutes

6.1.5 Decision Making

A final ON/OFF decision for a boiler can be made by comparing the time to get home (T_{AH}) with the time needed for preheating a home (T_{PH}). In particular, when $T_{PH} \geq T_{AH}$, the boiler will be on based on the current situation. The system will continue checking the user's and the home's states and making decisions for the home heating system in a certain frequency until the At Home state is triggered. From this, all the GPS location checking will be temporarily stopped until the state of the user's mobile device is disconnected from the home Wi-Fi network again. Surely, if the user is far away from home, the FRI system will keep the home heating system in the OFF position to save energy usage.

6.2 Experimentation

The proposed intelligent heating management system has been applied to a real-world situation for the purposes of validation and evaluation. Suppose that the resident is on their way back home by driving from location A at 12:00 PM on Wednesday, as shown in Fig. 6.4. As location A is far away from home, the current heating is OFF. During the travelling, the user's GPS location data will be obtained and sent back to the heating control system by the mobile device every two minutes. Based on returned information, the user's travelling time and estimated time of arrival could be obtained by the designed algorithm, as shown in Fig. 6.1. The testing house environment is a newly built four-bedroom detached house, and the total heating area is about 100 m^2 with a 2.4 metres ceiling height. The output power of the heating boiler is 15 KW. The home temperature is 16°C, and the user's satisfied temperature has been set to 20°C. Based on Table 6.2, T_{PH} can be calculated, which is $T_{PH} = 50$ $minutes$. Five different time points during the travelling have been selected to demonstrate the working progress of the system.

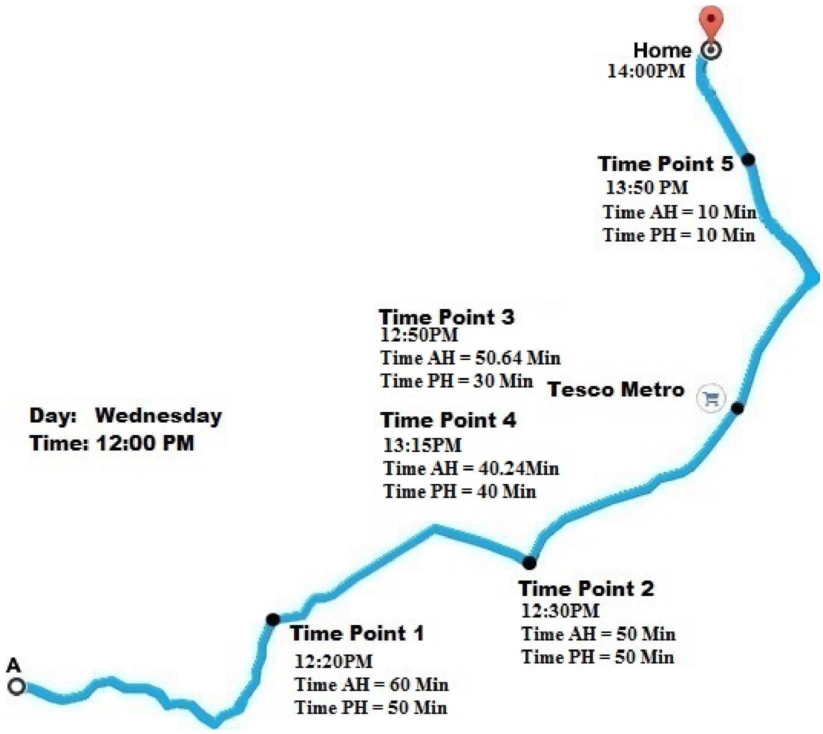


Fig. 6.4 The map used in the experimentation

6.2.1 Time Point 1

Based on the GPS data in the user's portable device, the comparison algorithm shown in Fig. 6.2 is used to detect the home user's state, who is currently on the way back home. Then, the returned strings from *Google Place API* are checked to see if the user has gone shopping, and in this case the user does not go to any shops. As no special location event occurred, the time to get home (T_{AH}) is the same as the travel time, which is provided by *Google Distance Matrix API*. In this case, $T_{AH} = 60\ minutes$. As mentioned above, the T_{PH} has already been calculated based on the current situation of the home and its environment, which is $T_{PH} = 50\ Minutes$. It is obvious in this case that $T_{AH} > T_{PH}$. Therefore, the system turned the home heating system off and waited for the next cycle of location checking and decision-making. The decision-making process is summarised as follows:

1	User State	$\rightarrow$	Way Back Home
2	Special Loction ?	$\rightarrow$	No
3	Calculate T_{AH}	$\rightarrow$	$T_{AH} = 60Min$
4	Calculate T_{PH}	$\rightarrow$	$T_{PH} = 50Min$
5	Compare T_{AH}, T_{PH}	$\rightarrow$	$T_{AH} > T_{PH}$
6	Heating	$\rightarrow$	Turn OFF.

6.2.2 Time Point 2

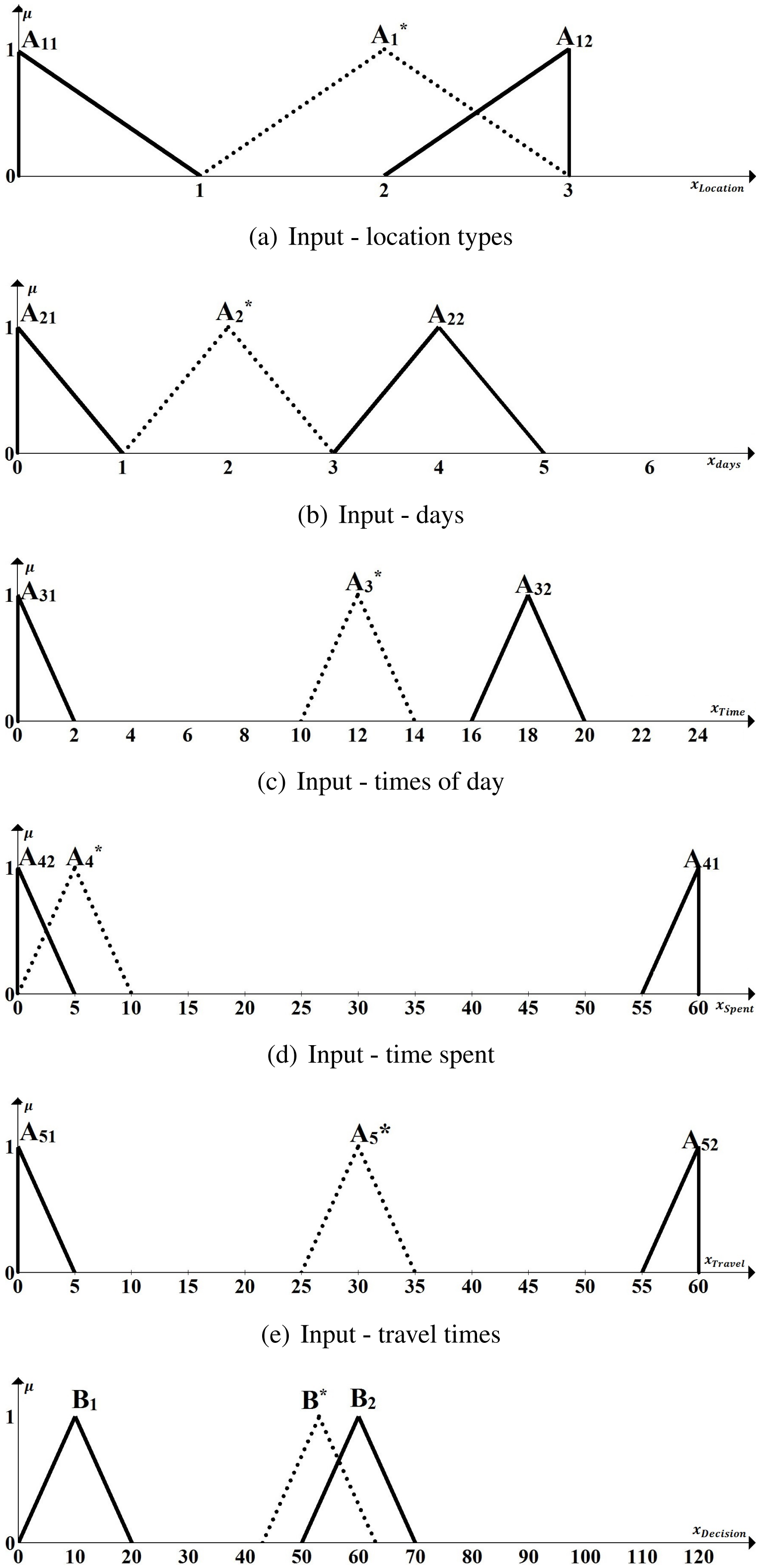
During the travelling, the user was getting closer to home. At time point 2, the returned $T_{AH} = 50\ Minutes$, which is equal to T_{PH}. The system will then decide to turn the home heating system on to preheat the room, as the progress shows below:

1	User State	$\rightarrow$	Way Back
2	Special Location ?	$\rightarrow$	No
3	Calculate T_{AH}	$\rightarrow$	$T_{AH} = 50Min$
4	Calculate T_{PH}	$\rightarrow$	$T_{PH} = 50Min$
5	Compare T_{AH}, T_{PH}	$\rightarrow$	$T_{AH} = T_{PH}$
6	Heating	$\rightarrow$	Turn ON
7	Home Yet ?	$\rightarrow$	No.

6.2.3 Time Point 3

The system keeps running to check the user's location data during the travelling, and in a particular time point of 3, the system detected that the user went to a shop equivalent to Tesco Metro, based on the returned location information by *Google Place API*. The system timer shows the user has stayed in for 5 minutes. In this particular case, the transformation-based FRI was employed to estimate the home time. In particular, the observation is ($A_1^* = (1,2,3)$, $A_2^* = (1,2,3)$, $A_3^* = (10,12,14)$, $A_4^* = (0,5,10)$, $A_5^* = (25,30,35)$), which does not overlap with any rule antecedent. The two closest neighbouring rules used for interpolation are $A_{11} \wedge A_{21} \wedge A_{31} \wedge A_{41} \wedge A_{51} \Rightarrow B_1$ and $A_{12} \wedge A_{22} \wedge A_{32} \wedge A_{42} \wedge A_{52} \Rightarrow B_2$, as shown in Fig. 6.5. In this case, based on Eq. 2.13, the relative placement factors can be calculated: $\lambda_1 = 2.5$, $\lambda_2 = 0.83$, $\lambda_3 = 1.89$, $\lambda_4 = 15.5$, $\lambda_5 = 1$, and the average $\lambda_{ave} = 4.34$ was used to calculate the intermediate rule result B'. Then, based on the calculated $\lambda_i, i = (1,2,3,4,5)$, the obtained scale rate for each variables (Eq. 2.16) are $S_1 = 0.5$, $S_2 = 0.92$, $S_3 = 1.57$, $S_4 = 1.67$, and $S_5 = 0.46$. The average scale rate $S_{ave} = 1.02$, works together with the combined move ratio, which is 0.87 (the average of the five move ratios (0.70, 0.27, 1.40, 0.62, 1.37)), is employed to achieve the final result B^*, which is $B^* = (40.64, 50.64, 60.64)$. The centre of gravity principle was used to defuzzify the generated result, and $T_{AH} = 50.64$ *minutes* has been generated. This value is then compared with T_{PH}. Because the home heating system has been turned on at Time Point 2 for a while, T_{PH} was re-calculated, which is currently $T_{PH} = 30$, and thus $T_{PH} < T_{AH}$. The system then turned the home heating system off immediately and returned to the beginning to prepare for the next prediction cycle.

1	User State	$\rightarrow$	Way Back
2	Special Loction ?	$\rightarrow$	Yes
3	FRI Reasoning	$\rightarrow$	$T_{AH} = 50.64 Min$
4	Calculate T_{PH}	$\rightarrow$	$T_{PH} = 30 Min$
5	Compare T_{AH}, T_{PH}	$\rightarrow$	$T_{AH} > T_{PH}$
6	Heating	$\rightarrow$	Turn OFF.



(a) Input - location types

(b) Input - days

(c) Input - times of day

(d) Input - time spent

(e) Input - travel times

(f) Output - Decision AH

Fig. 6.5 Example of fuzzy rule interpolation

6.2.4 Time Point 4

At Time Point 4, 25 minutes ($T_{Spend} = 25\ Minutes$) has been spent in the shop, and T_{PH} has changed to 40 minutes as the heating did not turn on for $T_{Spent} = 25\ Minutes$. At this time point, the generated result from FRI reasoning was 40.24 minutes, which is less than the T_{PH}. The system therefore turned the heating system back on. Table 6.3 summarises the results of FRI reasoning. As the user was still in the shop, the system will check the user's state until they return home.

1	User State	$\rightarrow$	Way Back
2	Special Location ?	$\rightarrow$	Yes
3	FRI Reasoning	$\rightarrow$	$T_{AH} = 40.24Min$
4	Calculate T_{PH}	$\rightarrow$	$T_{PH} = 40Min$
5	Compare T_{AH}, T_{PH}	$\rightarrow$	$T_{AH} \sim= T_{PH}$
6	Heating	$\rightarrow$	Turn ON
7	Home Yet ?	$\rightarrow$	No.

Table 6.3 FRI reasoning for time point 4

Antecedents		**Observation**
$A_{11} = (0,0,1)$	$A_{12} = (2,3,3)$	$A_1^* = (1,2,3)$
$A_{21} = (0,0,1)$	$A_{22} = (3,4,5)$	$A_2^* = (1,2,3)$
$A_{31} = (0,0,2)$	$A_{32} = (16,18,20)$	$A_3^* = (10,12,14)$
$A_{41} = (0,0,5)$	$A_{42} = (55,60,60)$	$A_4^* = (20,25,30)$
$A_{51} = (0,0,5)$	$A_{52} = (55,60,60)$	$A_5^* = (25,30,35)$
$B_1 = (10,20,30)$	$B_2 = (50,60,70)$	
Results		$B^* = (30.24,40.24,50.24)$

6.2.5 Time Point 5

The home users travelled again after the shopping. At Time Point 5, the system detected that the user is on the road travelling back home, and T_{AH} is 10 minutes. T_{PH} is also updated to 10 minutes in this case. Therefore, after the comparison, the home heating system is still on and the user's current state does not meet the at home state yet. The system continued making decisions until the users returned home.

1	User State	$\rightarrow$	Way Back
2	Special Loction ?	$\rightarrow$	No
3	Calculate T_{AH}	$\rightarrow$	$T_{AH} = 10Min$
4	Calculate T_{PH}	$\rightarrow$	$T_{PH} = 10Min$
5	Compare T_{AH}, T_{PH}	$\rightarrow$	$T_{AH} = T_{PH}$
6	Heating	$\rightarrow$	Turn ON
7	Home Yet ?	$\rightarrow$	No.

At 14:00 PM, the user finally reaches the At Home state as the mobile device was connected to the home Wi-Fi system. In the above case, the users came back home about 40 minutes later than their normal schedule because of shopping. Although the home heating system has been turned on at Time Point 2 for about 20 minutes, it was turned off for about 15 minutes immediately after a special location event was detected at Time Point 3. When the users arrived home, the house temperature had just been preheated to the desired temperature, which is 20°C. The system let the user benefit from a warm home but avoided energy waste.

6.2.6 Discussion

Most of the home heating controllers in the UK are schedule-based, and they do not have any learning method to acquire users' occupancy activity. Home heating systems are controlled by the scheduled time period. Once ON/OFF time periods have been scheduled, users do not normally re-configure them again, even if some scheduled times have slightly changed. For the environment for the experimentation, a schedule-based system is also used. In particular, the schedule-based controller has been set to turn the heating on between 12:30 PM and 13:20 PM every day to increase house temperature as the user normally comes back at 13:20. The different operations between the schedule-based controller and the proposed system are listed in Table 6.4.

The schedule-based heating controller is able to control the home heating system based on the pre scheduled time table. In the above situation, the home heating system has been turned on for 50 minutes in total by schedule-based controller. However, the home heating system was turned off 30 minutes before the user came back home and heated an unoccupied house for about 30 minutes. As a result, the home temperature did not reach the desired temperature when the user came back home as the heating was lost in the cold environment. The user has to manually turn the heating on for an extra 15-20 minutes to reach the satisfying temperature when they came back home.

Table 6.4 Comparison with schedule based controller

Time	Description	Schedule Based Operation	Proposed System Operation
12:00	Set Off	OFF	OFF
12:20	Time Point 1	OFF	OFF
12:30	Time Point 2	ON	ON
12:50	Time Point 3	ON	OFF
13:15	Time Point 4	ON	ON
13:20	Between TP4 and TP5	OFF	ON
13:50	Time Point 5	OFF	ON
14:00	At Home	OFF	OFF

Although the heating has been turned on between Time Points 2 and 3 for 20 minutes, the proposed system managed to turn it off immediately when a special event had been detected. The FRI system helps the proposed system to make the correct decision. Although the total heating time duration is 65 minutes, which is longer than the perfect situation, the home temperature can be just pre-heated to the desired temperature when the user comes back home and the total heating time is not longer than the schedule-based controller, which discussed above.

A number of learning-based smart home central heating system controllers have been proposed, which are usually combined with some novel features such as schedule learning, remote access, and occupancy sensing with auto-away mode. Although the proposed system does not provide any of the above three functionalities, it is able to successfully predict the user's home time based on the data captured through portable devices, and thus to effectively and efficiently pre-heat the home, which may not be possible by other approaches.

6.3 Summary

This chapter presented a smart home heating controller, which is able to control a heating system to preheat a property before home users getting home. The controller is developed by adapting fuzzy rule interpolation, supported by location information through portable devices. In particular, the system first predicts the time before home users get home, then the time to preheat the home is approximated. If the predicted time of getting home is not greater than the time to preheat the home, the heating system will be switched on. As shown in the demonstrative example, the proposed system is able to automatically provide a solution to preheat the home when there is a need, but not leave the heating system on all the time resulting in energy waste. Therefore, by applying such a system, home users can enjoy the benefit of energy saving, but without sacrificing the quality of life by suffering from a cold home during the home heating processes, as most properties with existing home heating controllers do.

Chapter 7

Network Intrusion Detection by TSK+

Cybersecurity has become an elevated risk that is amongst the most pressing issues affecting businesses, governments, other organisations, and even individuals. This issue is expected to become more important in time, as more devices, 'the internet of things', will be connected to the internet. Network security is one of the important challenges in the field of cybersecurity, knowing that networks provide the essential access to others which need to be protected in cyberspace, including the computer systems and data. Serious network attacks can lead to damages on computer systems, network paralysis, data loss or leakage. Network intrusion detection systems (NIDS) attempt to identify unauthorised, illicit, and anomalous behaviour based solely on network traffic to support decision making in network prevention actions by network administrators.

Several machine learning methods have been applied, such as [59, 110, 227, 226], which can be basically grouped into two types inspect suspicious traffic through either signatures or anomalies. The signature-based approaches attempt to classify a network connection based on an already known signature knowledge base [81, 179]. This group of methods are only applicable to the detection of already known types of threats. Differently, anomaly detection approaches, such as [4, 181], are introduced to detect unknown types of attacks by identifying the behaviour of network traffic that does not conform to any expected pattern in their knowledge bases. The accuracy of this type of approaches is limited by the lack of abnormal knowledge. Common to both types of approaches is that the requirement of a well-covered dataset or knowledge base, which are not always readily available or obtainable. In addition, the resultant system may be very complex if sufficient data is available. This chapter proposes a novel NIDS using previously proposed TSK+ fuzzy interpolation and its rule base generation approaches by which the above limitations can be overcome. In particular, a 0-order TSK rule base is generated from historical traffic data, which is guided by the proposed data-driven rule base generation method presented in Chapter 3.3. Once the rule base has been generated, the proposed TSK+ fuzzy interpolation approach then takes place to produce the output for hte given input. The proposed NIDS is not only able to create security alerts for known attack types, but also to detect potential unknown threats, as demonstrated in the experimentation.

The rest of the chapter is structured as follows. Chapter 7.1 introduces the background of the intrusion detection system. Chapter 7.2 presents the details of the proposed IDS. Chapter 7.2 demonstrates the proposed system by using a well-known benchmark dataset, NSL-CUP-99. Chapter 7.4 summarises the chapter.

7.1 Background of Intrusion Detection System

Data are sliced into a number of units during transmission over Ethernet. Each group of data, named a packet, is formed by adding an extra header section on top of the transmitting unit of data. Based on packet headers, important features regarding the corresponding data transaction can be identified, such as the source and destination IP addresses, the total length of the formatted data unit, and the source/destination ports. From this, the traffic pattern or signature of each data transmission in the network environment can be identified by analysing the information included in IP headers. These traffic patterns and signatures are commonly utilised by NIDS to identify potential threats in a network environment, thus to generate security alerts.

Soft computing algorithms have been widely employed for the development of NIDSes [148] to enable an intelligent agent in the system that is capable of disclosing the latent patterns in abnormal and normal connection audit records, and to generalise the patterns to new connection instances of the same class. The use of artificial immune systems in intrusion detection is appealing as it is very challenging to defend increasingly complex networks in a dynamic environment. The artificial immune system is inspired by a human immune system which protects the human body from invading germs and other micro-organisms. NIDSes have also been developed using artificial neural networks, each of which is composed of a number of neurons that are interconnected with each other with different weights. Genetic algorithm-based NIDSes utilise biological concepts of natural selection, that is, survival of the fittest. Fuzzy logic has been employed in either fuzzy inference-based IDSes or fuzzified other soft computing approaches to handle uncertainty.

Amongst the existing fuzzy logic-based NIDSes, D-FRI-Snort [135, 136] developed a dynamic fuzzy rule interpolation-based intrusion detection system. This system provides an additional level of intelligence to Snort, alongside the liability of creating a dynamic rule base, which enhances the prediction capability of the original Snort. In particular, D-FRI-Snort starts with network traffic data collection and analysis using three features: average packet time, the number of packets sent and that received. Threat alerts are generated to indicate an additional threat level if advised by the existing rule base. The system also stores interpolated results and dynamically promotes new rules based on the collected interpolated rules to enhance the original Snort according to the current network conditions.

7.2 Network Intrusion Detection

The overall system is outlined in Fig 7.1, which comprises of three main parts. The proposed system starts from dataset collection, which is used as a training dataset for the data-driven rule base generation method. The TSK rule base generation method, which is presented in Chapter 3.3, is employed to extract the fuzzy rules based on the collected training dataset. And the TSK+ fuzzy interpolation approach is used in this work as the inference engine to detect whether an attack is coming or not for incoming network traffic. Important procedures in the system modelling are detailed below. For simplicity, in this work, the modelling is assumed to take place under a noise-free network environment. Also, only normal and convex triangle fuzzy sets are used in fuzzy system modelling.

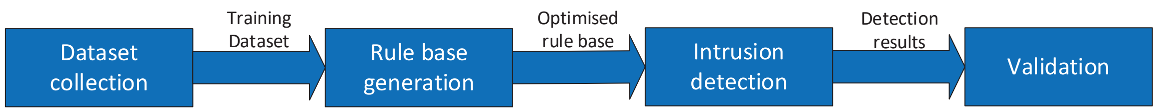


Fig. 7.1 The overall system

7.2.1 Data Collection and Feature Selection

Multiple general features can be readily monitored by networking tools for networking analysis during data packet transmission over the network. The Table 7.1 presents a complete listing of a set of features characterised that can be monitored [168]. Among with such features, most of them are irrelevant to intrusion detection [137]. Therefore, a well-thought-out feature selection by experts is often required for the task of network attack detection [71]. This common practice is also employed in this work.

Table 7.1 A complete list of features for traffic monitoring

No.	Feature	Description
1	Duration	Length (number of seconds) of the connection
2	Protocol_type	Type of the protocol
3	Service	Network service on the destination, e.g., http, telnet, etc.
4	Flag	Normal or error status of the connection
5	Src_bytes	The number of data bytes sent by source IP host
6	Dst_bytes	The number of data bytes sent by destination IP host
7	Land	1 if connection is from/to the same host/port; 0 otherwise
8	Wrong_fragment	Number of “wrong” fragments
9	Urgent	Number of urgent packets
10	Hot	Number of “hot” indicators
11	Num_failed_logins	Number of failed login attempts
12	Logged_in	1 if successfully logged in; 0 otherwise
13	Num_compromised	Number of “compromised” conditions
14	Root_shell	1 if root shell is obtained; 0 otherwise
15	Su_attempted	1 if “su root” command attempted; 0 otherwise
16	Num_root	Number of “root” accesses
17	Num_file_creations	Number of file creation operations
18	Num_shells	Number of shell prompts
19	Num_access_files	Number of operations on access control files
20	Num_outbound_cmds	Number of outbound commands in an ftp session
21	Is_hot_login	1 if the login belongs to the “hot” list; 0 otherwise
22	Is_guest_login	1 if the login is a “guest” login; 0 otherwise
23	count	number of connections to the same host as the current connection in the past two seconds
24	srv_count	number of connections to the same service as the current connection in the past two seconds
25	Serror_rate	% of connections that have “SYN” errors
26	Srv_serror_rate	% of connections that have “SYN” errors
27	Rerror_rate	% of connections that have “REJ” errors
28	Srv_rerror_rate	% of connections that have “REJ” errors
29	Same_srv_rate	% of connections to the same service
30	Diff_srv_rate	% of connections to different services
31	Srv_diff_host_rate	% of connections to different hosts
32	Dst_host_count	Count for destination host
33	Dst_host_srv_count	Srv_count for destination host
34	Dst_host_same_rate	% of connections that destination ports are same to the same destination IP in past 100 connections
35	Dst_host_diff_rate	% of connections that destination ports are different to the same destination IP in past 100 connections
36	Dst_host_same_src_port_rate	% of connections that source ports are same to the same destination IP in past 100 connections
37	Dst_host_diff_src_port_rate	% of connections that source ports are different to the same destination IP in past 100 connections
38	Dst_host_serror_rate	Serror_rate for destination host
39	Dst_host_srv_serror_rate	Srv_serror_rate for destination host
40	Dst_host_rerror_rate	Rerror_rate for destination host
41	Dst_host_srv_rerror_rate	Srv_serror_rate for destination host

In particular, four important features identified based on experts knowledge are selected as an IDS signature for the proposed system, which are listed in Table 7.2 [174, 207].

Table 7.2 Features for IDS

No.	**Feature**	**Description**
5	Source bytes	The number of data bytes sent by source IP host
6	Destination bytes	The number of data bytes sent by destination IP host
23	Count	The number of connections to the same host as the current connection in the past 2 seconds
35	Dst_host_diff_rate	% of connections that destination ports are different to the same destination IP in past 100 connections.

Once the features are determined, datasets for a given network in a certain environment need to be collected for model training. This is typically implemented in stages based on firstly an attack-free network and then different types of attacks that need to be identified. That is, data regarding normal network traffic are collected first from a threat-free condition network environment. Then, a number of attacks simulating the first type of attack is artificially launched such that this type of attack is sufficiently covered by the dataset. This process is repeated for every other type of attacks such that all the classes that need to be considered are fully covered by the dataset. The finally generated dataset is able to cover all the attack types and the attack-free situation.

7.2.2 Rule Base Generation

A TSK fuzzy rule base is able to be constructed by four main processes, which have been detailed in Chapter 3.3. Therefore, in this chapter, a well-known benchmark dataset, KDD Cup 99, is used as a training dataset to demonstrate how the proposed TSK rule base generation approach deals with a real-world problem.

7.2.2.1 The Dataset

The KDD Cup 99 dataset is a popular benchmark in the research field of intrusion detection, which includes legitimate connections and a wide variety of intrusions simulated in a military network environment [180]. This dataset contains almost 5 million data instances with 42 attributes, including the 'class' attribute, which indicates whether a given instance is a normal connection or one of the four types of attacks to be identified (i.e. Normal, Denial of Service Attacks, User to Root Attacks, Remote to User Attacks, and Probes). Knowing the inherent issues associated with the dataset, such as the high duplication rate of 78%, the KDD

Cup 99 dataset has been further processed to NSL-KDD-99 [180]. This processed dataset includes 125,937 data samples with all the features of the original dataset kept. Thanks to its convenience, this dataset has been utilised in a number pieces of recent research, such as [14], which is also employed for system validation and evaluation.

7.2.2.2 TSK+ Rule Base Construction

As the labels are symbolic values, 0-order TSK-style fuzzy rules were used. In order to construct a 0-order TSK rule base, the training dataset was divided into 5 sub-datasets based on the five symbolic labels, which are represented using five integer numbers. The sizes of the sub-datasets and their corresponding integer labels are listed in Table 7.3. The rule base generation process is summarised in four steps below.

Table 7.3 Data details regrading the types of connections

Datasets	**No. of instances**	**Classes**	**Rule consequence**
$\mathbb{T}_1$	53,874	**Normal Traffic**	1
$\mathbb{T}_2$	36,741	**DoS**	2
$\mathbb{T}_3$	42	**U2R**	3
$\mathbb{T}_4$	796	**R2U**	4
$\mathbb{T}_5$	9,325	**Probes**	5

Step 1 *Dense sub-dataset generation*: The sparse K-Means was applied on each sub-dataset $\mathbb{T}_j, 1 \leq j \leq 5$ to generate dense sub-datasets. Taking the second sub-dataset $\mathbb{T}_2$ as an example, the performance improvement led by the increment of k_2, $(k_2 \in \{1,2,\cdots,10\})$, is shown in Fig. 7.2. Following the Elbow approach, which has been details in Chapter 3.3.1.1, the elbow point on the drawn line chart indicated $k = 3$. Therefore, the 3 was selected as the number of clusters, i.e., $k_2 = 3$. The three generated dense sub-data-sets were denoted as (T_{21}, T_{22}, T_{23}). Note that the system performance improvement has also decreased to an interested value when $k = 4$ and $k = 5$. However, compared with performance improvement obtained when $k = 3$, the system did not improve the significant performance from $k = 4$. Therefore, according to the Elbow method, which should choose a number of clusters that adding another cluster does not give much better performance, the $k = 3$ supposes to be selected.

Step 2 *Rule cluster generation*: The standard K-Means clustering algorithm was applied on T_{ji} ($j \in \{1,2,\cdots,5\}$ and i ranging from 1 to the determined cluster numbers using the Elbow approach), to generate rule clusters each representing a rule. Again, take $\mathbb{T}_2$ as an

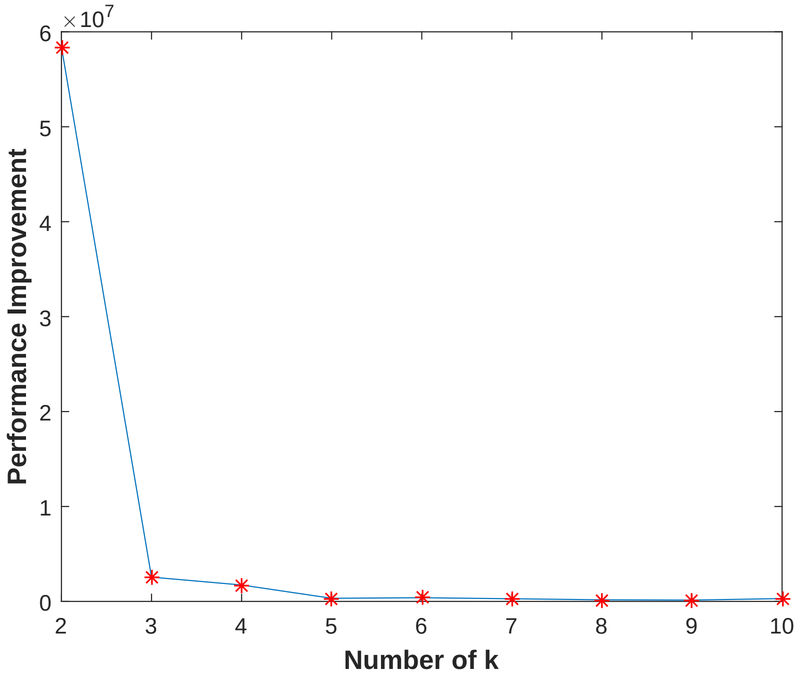


Fig. 7.2 Performance improvement regarding incremented k

example for illustration. The determined cluster numbers k_{2i} for each dense sub-dataset T_{2i} are shown in the third column of Table 7.4. Then, 13 rule clusters were generated from the sub-dataset $\mathbb{T}_2$. The rule cluster generation process for other dense sub-datasets is not detailed here, but the generated rule cluster or the given training dataset are summarised in Table 7.4.

Table 7.4 Rules and clusters for each set of data instances with the same type of connection

	Normal $\mathbb{T}_1$			**DoS** $\mathbb{T}_2$			**R2U** $\mathbb{T}_3$	**U2R** $\mathbb{T}_4$			**Probes** $\mathbb{T}_5$		
Dense dataset	T_{11}	T_{12}	T_{13}	T_{21}	T_{22}	T_{23}	T_{31}	T_{41}	T_{42}	T_{43}	T_{51}	T_{52}	T_{53}
Index i of rule cluster RC_i	1-4	5-7	8-11	12-14	15-18	19-24	25-30	31-35	36-38	39-41	42-44	45	46
Index i of rule R_i	1-4	5-7	8-11	12-14	15-18	19-24	25-30	31-35	36-38	39-41	42-44	45	46

Step 3 *Raw rule base generation*: A rule is extracted from each generated rule cluster. Taking rule cluster RC_{12} as an example, which is the first determined rule cluster in T_{21}, the corresponding rule R_{12} can be extracted. In particular, the consequence of rule R_{12} is the integer number representing the class of connections, and the rule antecedents are four triangle fuzzy sets ($A_{(12)1}$, $A_{(12)2}$, $A_{(12)3}$, $A_{(12)4}$) led by the approach discussed in

Section 3.3.1. The generated rule R_{12} is:

$$\begin{aligned} R_{12}: \textbf{IF} \quad & x_1 \text{ is } (0.588, 0.599, 0.601) \\ & \text{and } x_2 \text{ is } (0, 0.387, 0.414) \\ & \text{and } x_3 \text{ is } (0.05, 0.075, 0.1) \\ & \text{and } x_4 \text{ is } (0, 0, 0) \\ \textbf{THEN } & y = 2. \end{aligned} \tag{7.1}$$

According to Table 7.4, 46 rules in total were generated to initialise the rule base. The detailed initialised rule base can be found in Appendix A.

Step 4 *Rule base optimisation*: The GA was applied to optimise the membership functions of the fuzzy sets involved in the extracted fuzzy rules. The same GA parameters used in Chapter 3.4.1, as listed in Table 3.6, were also used in this example, but the number of iterations was increased to 20,000. The system performance against the number of iterations used in GA is shown in Fig. 7.3. The optimised rule base is attached in Appendix B.

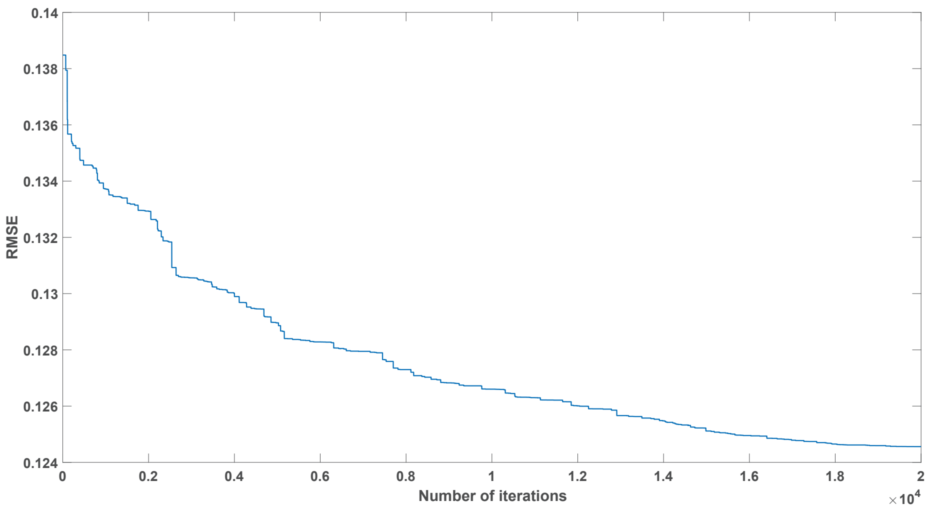


Fig. 7.3 RMSE decreasing over time

7.2.3 Intrusion Detection by TSK+

The TSK+ fuzzy interpolation approach is used to perform inferences in attack detection, as the generated TSK rule bases are usually sparse. In order to generate network intrusion alerts in real time, the system keeps capturing traffic data upon its deployment in a real network environment. The captured data in real time are then fed into the proposed system for system updating. Assume that the four features of current traffic for a given network are observed as $O = (x_1^*, x_2^*, x_3^*, x_4^*)$. From this input, the TSK+ approach first calculates the similarity degrees between the given input (O) and the antecedents of individual rules utilising Eq. 3.7. Then, the inference result of the TSK NIDS model is produced from Eq. 3.5. Finally, the numerical consequence values of the TSK-interpolation system is rounded to a whole number (symbolic value) which represents the category of the observed traffic.

7.3 Evaluation

The rule base and the inference engine TSK+ jointly formed the fuzzy model, which was validated and evaluated using a testing dataset. The testing dataset contained 22,544 data samples provided by [180]. The testing dataset was also extracted from the original KDD Cup 99 dataset but it does not share any data instance with the training dataset NSL-KDD-99.

Note that the testing dataset has been used in a number of projects with different classification approaches. In particular, the decision tree, the Naïve Bayes, the back-propagation neural network (BPNN), and the fuzzy clustering-artificial neural network (FC-ANN) have been employed in [190], and the modified optimum-path forest (MOPF) was applied in [15]. The accuracy of the classification results for each class of network traffic generated by different approaches including the proposed one with the initialised rule base and the optimised rule base, are listed in Table 7.5.

Table 7.5 Performance comparison between approaches [190]

	Normal Traffic	**DoS**	**U2R**	**R2U**	**Probes**
Decision Tree [190]	91.22	97.24	15.38	1.43	78.13
Naïve Bayes [190]	89.22	96.65	7.69	8.57	76.92
BPNN [190]	89.75	97.20	23.08	5.71	**88.75**
FC-ANN [190]	91.32	96.70	76.92	58.57	80.00
MOPF [15]	N/A	96.89	**77.98**	81.13	85.92
TSK+ without GA	77.10	94.07	57.69	55.29	78.71
TSK+ with GA	**93.10**	**97.84**	65.38	**84.65**	85.69

7.3.1 Discussion

The system evaluation has been carried out by applying the NSL-KDD-99 dataset. The experimental results have compared with 5 different approaches that were proposed in the literature, which have been illustrated in Table 7.5. The results show that the proposed TSK+

fuzzy inference approach outperformed overall all other approaches, and the proposed rule base optimisation method significantly improved the system performance by over 10% on average. In particular, the proposed system achieved better accuracies on the prediction of normal connections, DoS and R2U, than those of all other approaches, worse performance led by the proposed approach in the class of U2R compared to FC-ANN and MOPF, and a similar result was generated for class Probes with the existing best performance resulting from other approaches. In addition, such better system performance was achieved by integrating both the proposed work and the optimisation approach. Without the proposed rule base generation and TSK+ inference approaches, the optimisation approach, such as the GA in this work, will not be able to generate the rule base and perform the inference, based on the given dataset.

Although many fuzzy inference systems have been employed to help network administrators detect the illimitable network connections, they all require a dense fuzzy rule base to enable the performance of fuzzy inferences. However, in general, a dense rule base is difficult to be obtained to cover the entire input domain from collected training datasets, which restricts the effectiveness of traditional fuzzy inference approaches. As a result, intrusion detection systems developed from fuzzy inference systems with sparse knowledge bases may result in unexpected system outputs. Thanks to the relaxation on rule bases from TSK+ inference, the intrusion alerts can still be generated in cases when a given observation is not overlapped with any rule antecedents, which significantly improves the applicability of fuzzy inference systems in the field of network intrusion detection.

7.4 Summary

This chapter presents a data-driven network intrusion detection system, by employing the recently proposed TSK-interpolation approach. The sparse TSK rule base of the proposed inference system is extracted from a given training dataset collated from a particular network in a certain environment. The experiment results, using the benchmark dataset KDD-99 demonstrate that the proposed system is not only able to successfully generate security alerts for the known attack types, but is also able to detect the unknown types of threats with better success thanks to its good generalisation ability.

Chapter 8

Dynamic Quality of Service (QoS) Solution for Enterprise Network by TSK+

A multitude of applications require their data to be transmitted over networks, including real-time high quality voice and video data. The massive cyber traffic is getting busier thanks to the increasingly more connected devices, i.e., the 'internet of things'. The delay-sensitive data, such as real-time voice and video data, need to be transported with a limited delay. Quality of Services (QoS) is the measure or mechanism to ensure guaranteed high-quality performance regarding application data transmission over the network. The goal of QoS is to provide a preferential delivery service for the applications that need it by ensuring sufficient bandwidth, controlling latency and reducing data loss [193].

In general, there are two principal approaches to implement QoS in modern IP networks: Integrated Services (IntServ) and Differentiated Services (DiffServ). The IntServ model uses a specially designed protocol to request and reserve the network resources from network devices, whilst the DiffServ model tailors excess network traffic packets via the queueing strategies based on the value in the DSCP field of an IP header. Regardless of which mode is used, traditional QoS mechanisms are usually based on the static policy which allocates the priority for each IP packet based on expertise. A number of more efficient QoS mechanisms have been recently developed by employing AI algorithms, such as neuron networks and fuzzy logic [10, 68, 162, 223]. These approaches are successful in general, but they are not able to flexibly deal with dynamic priority requirements for different types of clients. To address such limitation, this chapter proposes a dynamic DiffServ-based QoS solution for enterprise networks, implemented using the proposed TSK+ approach.

This chapter presents a dynamic QoS solution based on the differentiated services (DiffServ) approach for enterprise networks, which is able to modify the priority level of a packet in real time by adjusting the value of the Differentiated Services Code Point (DSCP) in the Internet Protocol (IP) header of network packets. This is implemented by a 0-order TSK fuzzy model with a sparse rule base which is developed by considering the current network delay, application desired priority level and user current priority group. DSCP values are dynamically generated by the TSK fuzzy model and updated in real time. The proposed system has been evaluated in a real network environment with promising results generated.

The rest of the chapter is structured as follows: Chapter 8.1 introduces the theoretical underpinnings of the quality of services. Chapter 8.2 presents the proposed QoS system utilising the TSK+ approach. Chapter 8.3 discusses the experimentation to demonstrate the work of the proposed system. Finally, Chapter 8.4 concludes the chapter.

8.1 Quality of Services

QoS is often implemented in enterprise networks to measure and manage the transmission quality for the different types of applications, and is usually determined by the following three factors [176]:

Loss rate : the number of packets that were not received compared to the total number of packets for transmission. Loss is typically a function of network availability. In a highly available network, the loss should be essentially 0.

Delay : the finite amount of time it takes a packet to reach the final destination device after being transmitted from the sending endpoint device.

Jitter : also called delay variation, which is the difference in the end-to-end delay between packets, or the measuring time difference in packet inter-arrival time.

The attribute of delay somehow reflects the other two attributes. In particular, a high amount of delayed packets commonly suggests congestion in the network, which usually also leads to a high percentage of packet loss and unstable data transmission, and vice versa.

As introduced earlier, there are generally two types of QoS mechanisms in modern IP networks. DiffServ provides QoS by differentiating the traffic, whereas IntServ provides QoS by building a virtual circuit using the bandwidth reservation technique. IntServ requires the nodes in the network to remember the state information about the flow, while DiffServ does not as it operates on individual IP packets. Particularly in the DiffServ QoS approach, network devices use the queueing strategy to tailor performance to expectations. In general, manufacturers have their own queueing methods to manage the QoS, such as first-in-first-out (FIFO), weighted fair queueing (WFQ), and priority queueing (PQ).

Regardless of the queueing strategies, a priority value is assigned to each packet to indicate the different service level requirement for the particular application. The priority value is represented by the first six bits in the DSCP field of the IP header, as shown in Fig. 8.1. In theory, a six bit binary value can define up to 64 possible priority values. However, in the 64 possible values, only 21 of them are commonly used, which are listed in Table 8.1. Other fields of the header each have their purpose, but this is out of the scope of this chapter and thus omitted. According to the types of traffic, such as real-time voice data and bulk data, the QoS mechanism is able to allocate a suitable priority value from 21 common values upon the current QoS policies to the corresponding traffic packet, thus enabling the routers and switches to manage the quality of services.

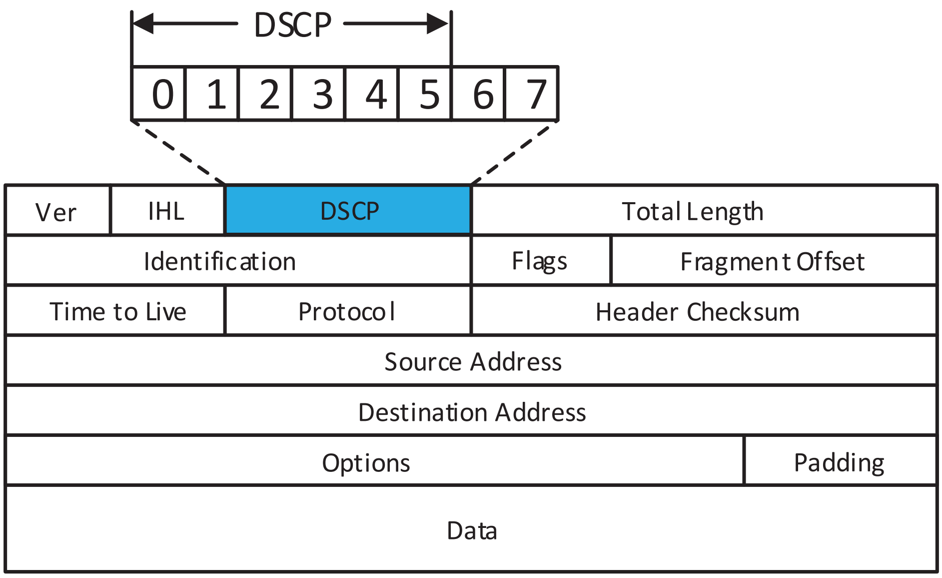


Fig. 8.1 IPv4 header and its DSCP field

Table 8.1 Commonly used DSCP values

No.	**DSCP Field Binary**	**DSCP Field Decimal**	**No.**	**DSCP Field Binary**	**DSCP Field Decimal**
1	000 000	0	11	011 010	26
2	001 000	8	12	011 100	28
3	001 010	10	13	011 110	30
4	001 100	12	14	100 000	32
5	001 110	14	15	100 010	34
6	010 000	16	16	100 100	36
7	010 010	18	17	100 110	38
8	010 100	20	18	101 000	40
9	010 110	22	19	101 110	46
10	011 000	24	20	110 000	48
			21	111 000	56

8.2 Dynamic QoS for Enterprise Networks

The priority level is usually determined by a static priority policy in a QoS-enabled network environment. This work focuses on the development of a decision-making system to determine the value of the DSCP in real time. This is achieved by developing a 0-order TSK fuzzy reasoning system, which considers both the current network congestion index and application/user desired priority. In particular, for the network packets from an application, the value of the corresponding the DSCP generated by the proposed system is fed into the field of DSCP in the IP header. Then, this packet is forwarded to the next hop device, such as a router or a switch, and joins its priority queue. Based on the value of the DSCP, the network device sends this packet out immediately, with a certain delay, or even drops off. The working procedure of the proposed system is illustrated in Fig. 8.2. Notice that this chapter focuses on the development of the decision-making system for the DSCP priority value. How the network devices (i.e., routers, switches, etc.) handle the network traffic based on the determined DSCP values varies from manufacturer to manufacturer, which is beyond the scope of this chapter.

8.2.1 Rule Base Generation

8.2.1.1 Rule Antecedent Features

Within the three factors considered in QoS (i.e. loss rate, delay and jitter), the attribute of delay somehow reflects the other two attributes. Therefore, the attribute of delay is taken as

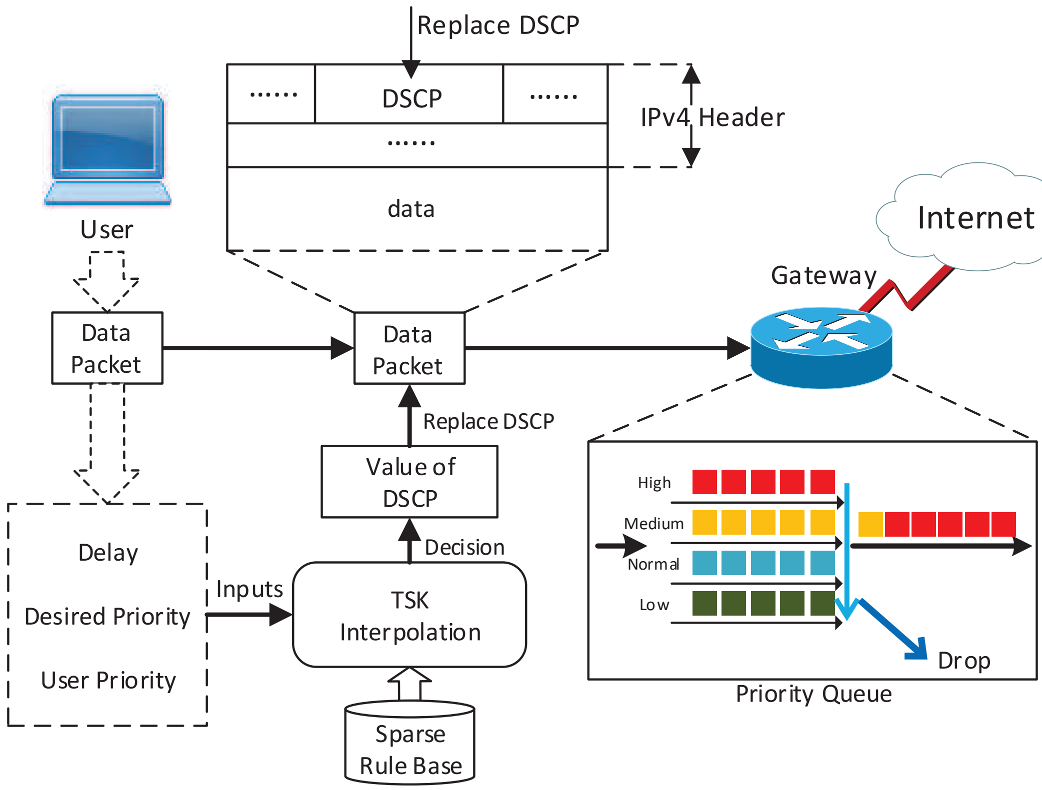


Fig. 8.2 The overview of dynamic QoS system

an input feature for TSK-interpolation in implementing the proposed QoS system. In order to allow the proposed system to deal with different situations in real time to reflect the dynamic priority requirements for different types of applications from different types of clients, two more attributes are also considered as system inputs: the application desired priority and the user priority group. Consequently, the proposed system takes three input features (i.e. the network delay, the application desired priority, and the user priority group), and it then produces one crisp output indicating the desired DSCP value. For simplicity, only normal and convex triangle fuzzy sets are used in rule base generation. The three input features and the fuzzy partition on these input domains are detailed below.

x_{delay}: The delay value is usually between 0 and 1,000 milliseconds. In this work, the variable domain is partitioned by 5 triangular fuzzy sets, as shown in Fig. 8.3(a), based on the knowledge of network engineers. Any delay value greater than 1,000 ms is capped at

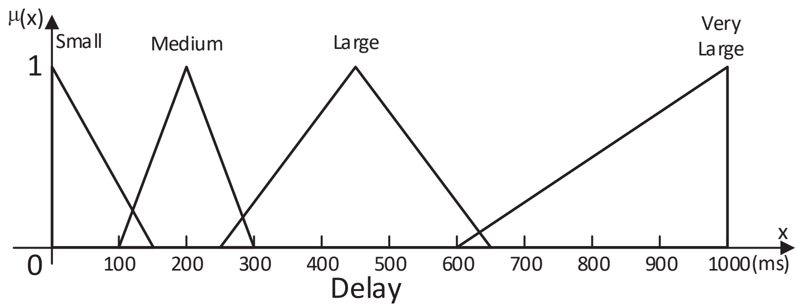
1,000 ms in this work, as the influence of a very large delay on network transmission is the same as 1,000 ms.

$x_{DesiredPrioirty}$: Every application has a unique traffic pattern and service level requirement, which usually requires a dedicated QoS priority to provide and guarantee the data transmission quality over networks. To help group multiple applications in the enterprise network, Cisco unifies the QoS baseline and provides standard-based recommendations, which defines 11 classes of applications based on their functionality [176]. The details of the Cisco QoS baseline classification and relevant priorities are listed in Table 8.2. These 11 linguistic classes outlined by Cisco are fuzzified by 11 fuzzy sets, as illustrated in Fig. 8.3(b). Notice that different groups of applications can be identified by network devices based on the source and destination IP address, the type of protocol and the source and destination port numbers in the IP header, which are beyond the scope of this chapter and not detailed here.

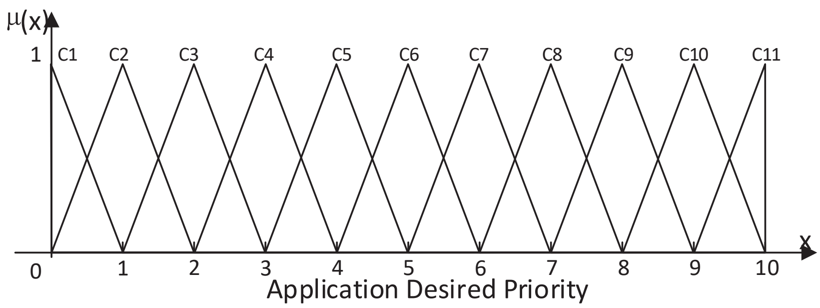
Table 8.2 Cisco QoS Applications classification

Application Class	**Description**	**Examples**	**Desired Priority Level**
IP Routing	Communication between network devices to ensure the network connectivity	OSPF, EIGRP	10
Voice	Voice data of VoIP telephony	VoIP traffic	9
Interactive Video	IP video conferencing	Skype video call, WebEx	8
Streaming Video	e-Learning application or multicast company meetings	Cisco DMS, IP/TV	7
Mission-Critical Data	Enterprise own defined critical applications that have the highest priority	N/A	6
Call-Signalling	Signalling traffic for VoIP	Dialling signal traffic from IP phone	5
Transactional Data	Interactive data applications	CRM	4
Network management	Network operations, administration, or management traffic	SSH, SNMP	3
Bulk Data	Non-interactive data applications, which means users are not awaiting a response	FTP/SFTP, Online backup	2
Best Effort	Default class non-classified applications should belong to	Web browsing	1
Scavenger	Non -business related	Gaming, P2P	0

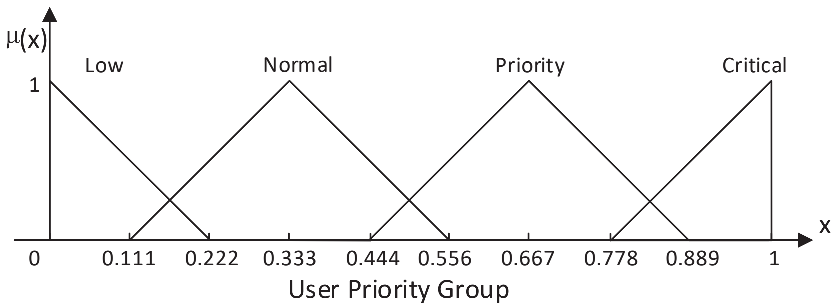
$x_{UserPriorityGroup}$: This input feature defines the priority of the end user to use the network resources. For instance, VoIP devices always require a higher service level than that for email servers. Also, the required service levels for different departments may vary. For example, staff in a sales department often requires a higher priority to use the internet resources than the shop floor staff, as smooth on-line payments should always be guaranteed. In this work, four different priorities are assigned to end users by following the work of [166]: low, normal,



(a) Input - Network delay



(b) Input - Application desired priority



(c) Input - User priority group

Fig. 8.3 Input fuzzy variables

priority, and critical. The fuzzy representation of this input domain partition is shown in Fig. 8.3(c), and the assignment of priority categories to different departments in a typical company is listed in Table 8.3. Note that flexible priority for different user groups may be required to maximise the efficiency of enterprise networks to support various activities. In this case, manual modification of a priority table is required.

Table 8.3 End user priority mapping

End Users	Class
VoIP Devices	Critical
Enterprise Servers	Priority
Department Managers	Priority
IT Department	Priority
Accounting Department	Normal
Purchasing Department	Normal
Others	Low

8.2.2.2 Rule Consequent Feature

As discussed in Chapter 8.1, the first six bits of the DSCP in the differentiated services field of an IP header indicate traffic priority. Currently, 21 DSCP values are commonly used to define 21 different priorities, which is shown in Table 8.1. The consequence of each rule is then an integer number between 1 and 21 to map the 21 commonly used priority levels of QoS.

8.2.2.3 Sparse TSK-style Rule Base

Based on the problem domain fuzzy partition discussed above, 176 fuzzy rules should be included in the rule base. However, the inclusion of the majority of these rules does not lead to a noticeable system improvement due to the generalisation ability of TSK-interpolation. Therefore, the rule base is reduced based on expert knowledge in this work. The Cisco QoS solution guide recommends a QoS priority mapping for different types of network services, which defines 11 important rules. The 11 fuzzy rules corresponding to the Cisco recommendation are kept in the rule base. For instance, Cisco recommends that the VoIP data requires a critical priority bandwidth; network delay should be no more than 150 ms, and should be marked to DSCP value 46. Also, the network management data should be marked to DSCP value 16, with the requirement of a minimum bandwidth and up to large delay. Based on those recommendations, 2 TSK rules are defined as follows:

$$\begin{aligned} R_1 : &\textbf{IF } x_{delay} \text{ is } S \text{ and } x_{DP} \text{ is } C10 \text{ and } x_{UPG} \text{ is } Critical \\ &\quad \textbf{THEN } z = 19, \\ R_2 : &\textbf{IF } x_{delay} \text{ is } L \text{ and } x_{DP} \text{ is } C4 \text{ and } x_{UPG} \text{ is } Normal \\ &\quad \textbf{THEN } z = 6, \end{aligned} \tag{8.1}$$

where the consequences $z = 19$ and $z = 6$ represent the DSCP values 46 and 16 (as shown in Table 8.1), respectively.

Note that only the domain of variable $x_{UserPriorityGroup}$ is fully covered by the 11 rules based on the Cisco recommendation. In order to have better coverage for the other two variable domains, more rules have been kept based on expert knowledge of network engineers, which results in a more compact and concise rule base containing 27 rules, as listed in Table 8.4.

Table 8.4 Generated rule base

No.	**Input**			**Output**	**No.**	**Input**			**Output**
	Delay	**ADP**	**UPG**	**Decision**		**Delay**	**ADP**	**UPG**	**Decision**
1	S	C11	C	20	**15**	L	C4	C	10
2	S	C10	C	19	**16**	L	C5	C	11
3	M	C9	P	15	**17**	L	C6	C	14
4	M	C8	P	14	**18**	M	C7	C	15
5	M	C7	N	11	**19**	M	C8	C	18
6	L	C6	N	10	**20**	M	C9	C	19
7	L	C5	N	7	**21**	VL	C9	C	20
8	L	C4	N	6	**22**	VL	C8	C	19
9	L	C3	L	3	**23**	VL	C6	C	15
10	VL	C2	L	2	**24**	VL	C5	C	13
11	VL	C1	L	1	**25**	VL	C4	C	14
12	VL	C1	C	2	**26**	VL	C3	C	9
13	VL	C2	C	3	**27**	VL	C2	C	5
14	VL	C3	VL	7					

8.2.2 Decision-making for QoS Priority

Once the sparse fuzzy rule base is generated, TSK-interpolation is then able to make decisions on the priority levels of data packets. In particular, for each network packet, the proposed system generates a valid DSCP value between 1 and 21 to replace the existing one included in the DSCP, thus to control its position in the priority queue. Note that only a sparse rule base which contains 27 rules is utilised in this work. The TSK fuzzy interpolation approach is naturally applied [111].

Assume that the network delay, the desired application priority, and the user group priority regarding a given traffic packet are given as $O = (x_1^*, x_2^*, x_3^*)$, where x_1^*, x_2^* and x_3^* are, of course, crisp values. The calculation process of the conclusion using the TSK-interpolation approach is summarised as follows:

Step 1: Calculate the similarity degrees between the given input (O) and rule antecedents of each rule using Eq. 3.7.

Step 2: Determine the firing degree of each rule by following Eq. 3.2.

Step 3: Aggregate the final crisp inference result through Eq. 3.5.

Step 4: Generate the final priority decision by applying the round function on the aggregated crisp inference result.

Once the crisp inference result has been generated, this value will be mapped to the corresponding DSCP value according to Table 8.1, and will be used to replace the existing one in the corresponding IP header. From this, the network device will put this packet into the priority queue waiting for a further operation.

8.3 Experimentation

The proposed TSK+ inference approach has been applied to deal with the dynamic QoS system in a real local area network (LAN), which is designed to simulate an enterprise network, for validation and evaluation purposes. The experimentation and its results are detailed in this chapter.

8.3.1 Experiment Environment

The network environment used in the experiment is shown in Fig. 8.4. In particular, the Cisco 2921 enterprise router with the DiffServ model enabled is used as the gateway router to handle different QoS requests. Two client PCs are employed as packets sender and receiver to simulate different types of network traffic. An IP phone has been set up with a SIP service to generate VoIP traffic. The proposed system is deployed upon a network bridge device between the router and the LAN devices. The network bridge decapsulates all network data packets between the router and terminal devices and then collects the required input values from network packets' IP headers for the proposed priority classification system.

The information in the protocol field of the IP header indicates the type of applications, and its desired priority can be acquired by looking it up in Table 8.2. The department of clients can be identified by the source IP address from the IP header; thus, the priority category of the corresponding end user group can be mapped using Table 8.3. Finally, the packet delay attribute can be detected by the network bridge itself. Based on these retrieved input values, the TSK-interpolation approach generates the corresponding priority values, which are in turn used to calculate the DSCP value. The calculated DSCP value is then re-encapsulated in the corresponding IP header for further processing. To facilitate network analysis, network monitoring software, such as Wireshark, is launched on the bridge side to capture and monitor the network traffic. The bandwidth on the router WAN port has been limited to 512 Kbps to simulate a network congestion in the real-world.

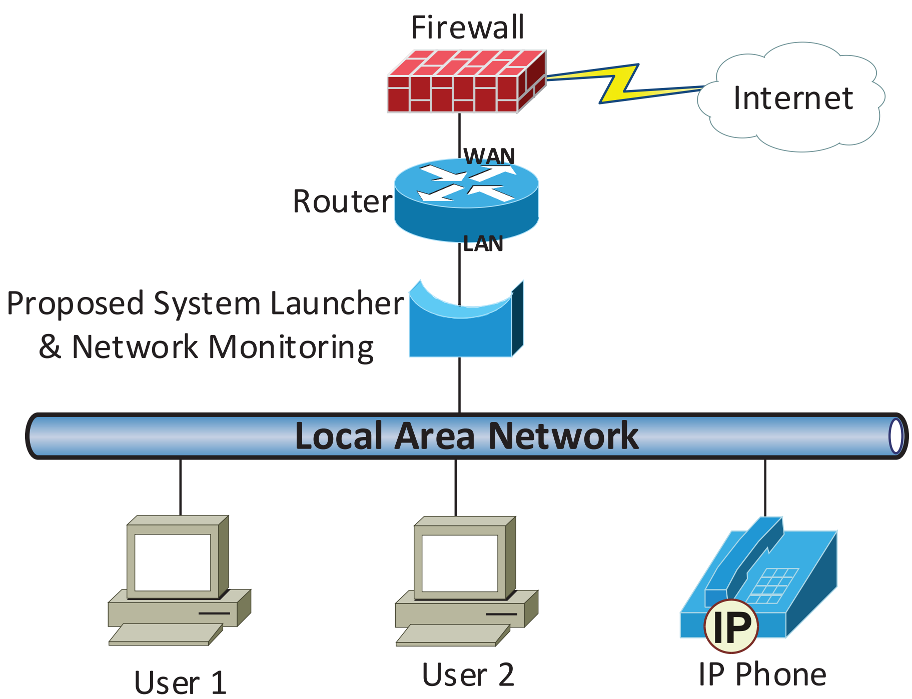


Fig. 8.4 The experiment environment

8.3.2 Experiment Results

The main purpose of this experimentation is to test whether the proposed TSK-interpolation system can successfully dynamically generate DSCP values according to network requirements for different situations. This experimentation was carried out by following three scenarios.

8.3.2.1 Scenario 1

This scenario considers a normal network condition where User 1 and User 2 are browsing web pages only, and the IP phone is making a telephone call via a SIP service. At this point, the end user priority groups for User 1 and User 2 both belong to low. The input feature values and the generated inference results by the proposed TSK-interpolation system are shown in Table 8.5. According to Table 8.1, the generated decisions 1 and 19 correspond to DSCP values 0 and 46, respectively. From the captured data by Wireshark, as shown in Fig. 8.5, the DSCP structure in binary is 101110 and its value in decimal is 46.

Table 8.5 Details for scenario 1

	Input			Output
	Delay	**Application Desired Priority**	**User Priority Group**	**Decision / Corresponding DSCP**
User 1	Very Small	C2	Low	1 / 0
User 2	Very Small	C2	Low	1 / 0
IP Phone	Very Small	C10	Critical	19 / 46

```
Differentiated Services Field: 0xb8 (DSCP: EF PHB, ECN: Not-ECT)
  1011 10.. = Differentiated Services Codepoint: Expedited Forwarding (46)
  .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
Total Length: 200
Identification: 0x0400 (1024)
Flags: 0x02 (Don't Fragment)
Fragment offset: 0
Time to live: 63
Protocol: UDP (17)
```

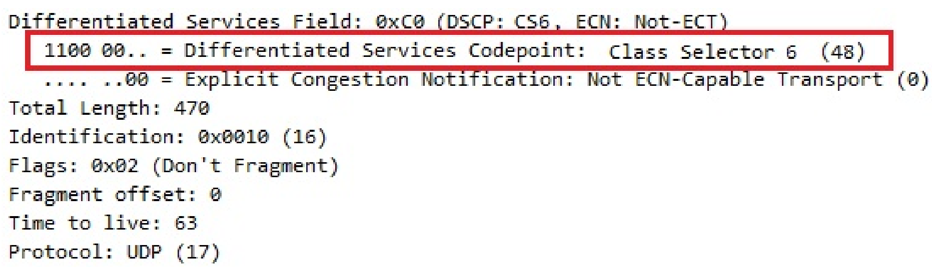
Fig. 8.5 Examplar DSCP value in VoIP packet for scenario 1

8.3.2.2 Scenario 2

From this point, User 1 enables a P2P (Peer to Peer) download, which usually takes all the available bandwidth, and causes a network congestion. As a result, as expected, the network delay is increased, which significantly limits the usable network resources and thus seriously affects the performance of the phone call through VoIP. Surely, this changing situation has been detected by the system, and the DSCP values for corresponding packets are accordingly adjusted. The DSCP results for this new situation are shown in Table 8.6 and the IP header structure captured by Wireshark is shown in Fig. 8.6. In particular, the priority value in binary is 110000, which in decimal is 48. This means the network resources have been successfully dynamically allocated for Scenario 2 by the proposed system.

Table 8.6 Details for scenario 2

	Input			**Output**
	Delay	**Application Desired Priority**	**User Priority Group**	**Decision / Corresponding DSCP**
User 1	Very Large	C1	Low	1 / 0
User 2	Very Large	C2	Low	1 / 0
IP Phone	Very Large	C10	Critical	20 / 48



```
Differentiated Services Field: 0xC0 (DSCP: CS6, ECN: Not-ECT)
  1100 00.. = Differentiated Services Codepoint: Class Selector 6 (48)
  .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
Total Length: 470
Identification: 0x0010 (16)
Flags: 0x02 (Don't Fragment)
Fragment offset: 0
Time to live: 63
Protocol: UDP (17)
```

Fig. 8.6 Examplar DSCP value in a VoIP packet for scenario 2

8.3.2.3 Scenario 3

In the above situation, the service quality of the web browser for User 2 is not guaranteed, because of the network congestion caused by P2P downloading. In order to get a better service quality for the web browsing, User 2 has submitted a request which has led to an increase of service level. The system input for different users based on the updated scenario and the generated DSCP results are listed in Table 8.7. It is clear from this table that the DSCP value of web browsing traffic has been increased to 10 from 0, which distinguishes the priorities between User 1 and User 2. In this situation, the DSCP value of VoIP traffic still remains as 48, which means the quality of the phone call over VoIP is not affected.

Table 8.7 Details for scenario 3

	Input			Output
	Delay	**Application Desired Priority**	**User Priority Group**	**Decision / Corresponding DSCP**
User 1	Very Large	C1	Low	1 / 0
User 2	Very Large	C2	Normal	3 / 10
IP Phone	Very Large	C10	Critical	20 / 48

8.3.3 Discussion

An illustrative example is given in this section to demonstrate the working procedure of the proposed dynamic QoS solution. In particular, there are three scenarios have been carried out. In the first scenario, a congestion-free network environment was considered. According to the information in the packet IP header, which has been captured and extracted by the Wireshark, the DSCP value of 46 in decimal was suggested by the proposed system for the VoIP traffic. In the second scenario, a congested network environment was simulated by enabling a number of P2P sessions. With the network congestion, the DSCP values of the current network traffic that extracted by the Wireshark confirmed the proposed system increased the corresponding DSCP values of the VoIP packet from 46 to 48. In the third scenario, one of the user's priority group has been manually increased. As a consequence, the proposed TSK-interpolation system detected the changes and slightly increased the DSCP values of the network packets, which generated by the corresponding user, to 10 while keeping the DSCP values of VoIP packets unchanged.

The above illustrative example confirmed that the proposed system is able to detect the network traffic changes and suggest the different DSCP value in the IP header based on dynamic needs, thus providing a dynamic QoS solution for the enterprise network.

8.4 Summary

This chapter presented a dynamic QoS solution for enterprise networks, which enhances the traditional QoS mechanisms by considering the data transmission requirements from applications and user groups. The proposed system is implemented by adapting the TSK fuzzy interpolation approach. The generated priority decisions are integrated in the DSCP fields of IP headers to manage the priority of data traffic. The experiment results show that the proposed system is able to dynamically advise data transmission priority for different situations to improve the quality of service and better the users/applications network requirements.

Chapter 9

Inverted Pendulum Control by FRI with Experience-based Rule Base Generation

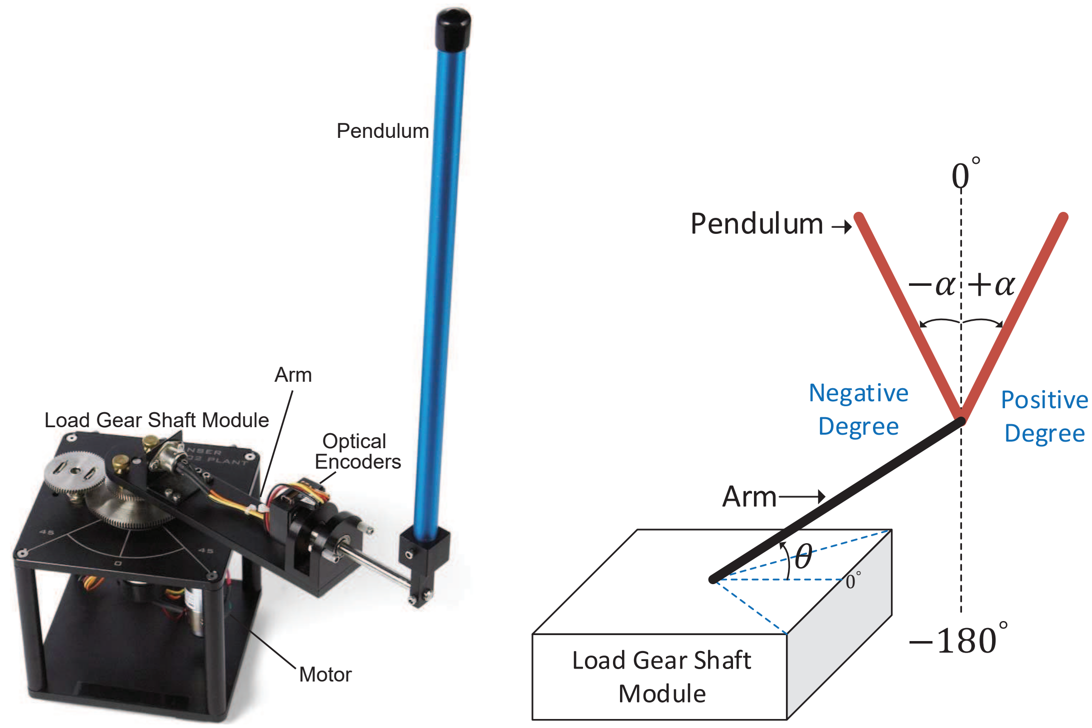
The inverted pendulum is a typical challenging problem in the area of control systems that is usually used for demonstration or experimentation when a novel theory or a new algorithm is proposed [1]. Generally speaking, there are two main types of inverted pendulum available: the cart type and rotary inverted pendulums. The first type is the classical model, by which a cart runs on a straight track to balance the pole. The second model uses a rotary arm instead of moving the chart to swing up and balance the pendulum. The controlling of inverted pendulum systems is underactuated and extremely non-linear due to the gravitational forces and the centripetal forces [25]. A number of researchers have focused on developing new control strategies for stabilisation of an inverted pendulum. As one of the most advanced technologies in the control field, the fuzzy inference system has been widely applied to solve such a problem, such as in [3, 51, 63, 82, 104, 146]. These systems usually require the expert knowledge to generate a dense fuzzy rule base for a fuzzy inference engine making a decision.

This chapter evaluates the proposed automatic rule base generation method that was presented in the Chapter 5. This evaluation is implemented by a well-known control problem: inverted pendulum. In this work, the control system is achieved by two fuzzy control systems: a fuzzy swing up controller and a fuzzy stabilisation controller. The rule base of the swing up controller is generated by adopting a knowledge-driven method, which manually creates fuzzy rules to map the way that human operators would swing up the pendulum. The fuzzy rule base of the stabilisation controller is developed by employing the proposed experience-based rule base generation and adaptation approach. In particular, the stabilisation control system initialises the rule base by generating two rules based on a random try mechanism.

From there, the KH approach is employed to predict the decision to control the arm. At the same time, the rule base revision process modifies the rule base that is guided by the performance index of each interpolated rule, as well as the usage frequency of existing rules. The experimental results confirmed that the control system is able to generate a reasonable rule base while performing fuzzy interpolative reasoning based on a feedback mechanism.

The rest of the chapter is structured as follows: Chapter 9.1 presents a description of the rotary inverted pendulum system. Chapter 9.2 gives the details of the system modelling. Chapter 9.3 details the experimental results. And finally, Chapter 9.4 concludes the chapter.

9.1 System Description



(a) Quanser rotary inverted pendulum model (b) Simplified schematic

Fig. 9.1 Quanser rotary inverted pendulum and its simplified schematic

The rotary inverted pendulum, also termed as the Furuta pendulum, is an underactuated mechanical system, which was first proposed by [67]. In this work, the Quanser rotary inverted pendulum model [7, 57] is used for the experimentation. The system model consists of three parts: an arm, a load gear shaft module with a motor and an actual pendulum, as shown in Fig. 9.1(a), and its simplified schematic is shown in Fig. 9.1(b). The arm is mounted on top of the load gear shaft module at one end and is fastened with the actual pendulum on the other end. The load gear shaft module drives the arm around in a circular motion, which leads the pendulum rotation. The optical encoders are instrumented on both ends of the arm, and are used to measure the angular displacement of the arm (θ) and the angle of the pendulum (α) in real time, respectively. Technically, both the arm and the pendulum are free to rotate 360°. However, in real-time operation, the arm is usually restricted to move within a fixed range, which is introduced by preventing the arm from becoming entangled with the connecting cables. By default, $\alpha = 180^\circ$ when the pendulum is at a natural stable downward position, and $\alpha = 0^\circ$ when the pendulum is at an upright position. For an easy explanation, Eq. 9.1 is employed to transfer the angle of the pendulum in a range of $(-180^\circ, 180^\circ)$ [2]:

$$\alpha = \frac{180}{\pi} \cdot \left(\mathrm{mod}\left(\frac{\text{Pendulum Encoder Reading} \cdot 2\pi}{4096}, 2\pi \right) - \pi \right) , \tag{9.1}$$

where 'Pendulum Encoder Reading' is the actual reading from the optical encoder attached to the pendulum, and $\mathrm{mod}(a, b)$ is the modulo operator, which returns the remainder after the division of a by b. The relationship between the pendulum status and the angle of the pendulum is illustrated in Fig. 9.1(b).

From this, for a given time t, the angular velocity of the pendulum $\dot{\alpha(t)}$, also represented as $\dot{\alpha}$ in this section, is able to be computed by a time derivative, such as [2]:

$$\dot{\alpha(t)} = \dot{\alpha} = \frac{\text{change in } \alpha}{\text{change in } t} = \frac{\triangle \alpha}{\triangle t} . \tag{9.2}$$

In the same manner, the angular displacement of the arm θ can be calculated as follows [2]:

$$\theta = \frac{180}{\pi} \cdot \frac{\text{Arm Encoder Reading}}{4096} \cdot 2\pi , \tag{9.3}$$

where 'Arm Encoder Reading' is the actual reading of the optical encoder for the arm. In particular, $\theta = 0^\circ$ when the arm is in the centre of the chosen sector, which is marked on the top panel of the load gear shaft module (which can be seen in Fig. 9.1(a)). According to the above description, a control system of this model usually takes two inputs, which are the angle of the pendulum (α) and the angle of the arm θ, to predict the signal output, which is the motor input voltage (v) for the load gear shaft model. A higher input voltage leads to a faster arm movement. The negative voltage causes the arm to turn counter-clockwise, and positive voltage leads to the arm rotating clockwise, and vice versa. Due to a health and safety reason, a range of -5 volts to + 5 volts is introduced by the Quanser rotary inverted pendulum user manual [2].

9.2 System Modelling

The controlling purpose of a rotary inverted pendulum is to swing up the pendulum from the downward position and then to balance it in the upright position. In this work, this is achieved by two fuzzy controllers, a swing up controller and a stabilisation controller, which are detailed in this section. In particular, in the beginning, the swing up controller continually moves the arm left and right to swing the pendulum up. Once the pendulum is in the upright position, the stabilisation controller then takes over the control to balance the pendulum. For simplicity, triangle membership functions are used in this work.

9.2.1 Swing Up Controller

In order to swing up the pendulum, a human operator usually moves the arm in one direction, and then moves it to the opposite direction when the angular velocity of the pendulum comes to 0, such as $\dot{\alpha} = 0$. The pendulum will be swung up to the upright position by repeating the above action. A number of fuzzy inference systems have been proposed in the literature to mimic human operation, such as [63, 104]. In this work, a Mamdani fuzzy inference system proposed in [104] is directly used for the fuzzy swing up control. In particular, the swing up FIS takes two inputs, which include the angular velocities of the arm and the pendulum, respectively, to produce a signal output that is the motor input voltage of the load gear shaft module. Note that the proposed experience-based rule base generation and adaptation method may also be considered to deal with the swing up control. As this is an initial investigation of application in this work, such consideration remains as future research.

9.2.2 Stabilisation Controller

A Mamdani-style fuzzy inference system is designed here to keep balancing the pendulum at the upright position continuously. The proposed FIS takes only one input, which is the angle of the pendulum (α), to predict the signal output, which is the motor input power voltage (v). In particular, the KH fuzzy rule interpolation approach is employed as the fuzzy inference engine, and the previously presented experience-based rule base generation and adaptation approach is employed in this work to create the rule base. The working processes of the rule base generation have followed exactly the framework outlined in Fig. 5.1, which is typically composed of two stages, the rule base initialisation, and the rule base generation and adaptation. Table 9.1 lists the parameters that are used in this implementation. In addition,

Table 9.1 System parameters

Parameters	**Definition**
α	The current angle of the pendulum
$\hat{\alpha}$	The previous angle of the pendulum
θ	The angle of the arm
v	The generated motor input voltage

the *Performance Index (PI)*, which is introduced to support the rule base generation and adaptation, is defined by $|\hat{\alpha}| - |\alpha|$. In particular, $|\hat{\alpha}| - |\alpha| > 0$ indicates a positive *PI*, which shows that the pendulum is more closed to the upright position after the arm is moved based on the generated decision. Contrarily, a negation *PI* will be obtained, if $|\hat{\alpha}| - |\alpha| \leq 0$, which shows that the pendulum is far away from the upright position after the arm action on the generated decision.

9.2.2.1 Rule Base Initialisation

The phase of the rule base initialisation initialises the rule base by creating two fuzzy rules based on a stochastic process. The processes of this stage can be summarised by following three steps, as illustrated in Fig. 9.2.

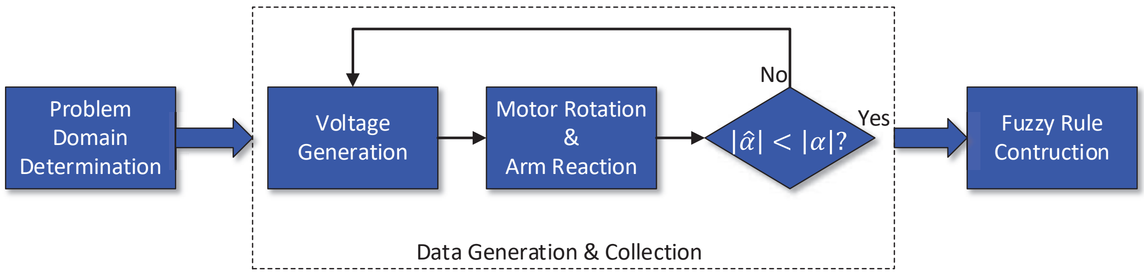


Fig. 9.2 The processes of rule base initialisation

1) *Determine problem domain:* The first step of the rule base initialisation is to determine the domain of the input variable for a given problem. According to the Quanser user manual [2], the problem domain for the stabilisation control problem is defined as follows:

$$\begin{aligned} &\textbf{Input: } x, \qquad x \in [-20,\ 20], \\ &\textbf{Output: } y, \qquad y \in [-5,\ 5]\ , \end{aligned} \tag{9.4}$$

where $x \in [-20, 20]$ indicates that the stabilisation controller will take over the balancing control only when $-20 \leq \alpha \leq 20$, and $y \in [-5, 5]$ restricts the maximum motor input voltage to five volts on each direction.

2) *Data generation and collection:* A stochastic process is used in this step to generate some useful data, based on the *performance index*. Basically, when the pendulum is swung up and is close to the upright position, where ($-20 \leq \alpha \leq 20$), the system randomly generates a number of motor input voltages (v), such as $-5 \leq v \leq 5$, to drive the arm. For each time the arm is actuated, the *performance index* is obtained to measure the quality of the generated v. The values of a previous pendulum's angle $\hat{\alpha}$ and the current input voltage v will be transformed to a fuzzy rule if a positive *PI* is returned. The system will keep randomly trying if a negative *PI* is obtained.

3) *Fuzzy rule construction:* Suppose a data instance, which contains an input value α and an output value v, has been collected from the previous step; a fuzzy rule can be transformed as:

$$R_i : \textbf{ IF } \alpha \text{ is } A_i, \textbf{ THEN } v \text{ is } B_i \; (w_i, \; EF_i, \; CD_i) \; , \tag{9.5}$$

where w, EF, and CD indicate the weight, *experience factor*, and *cooling down factor* of the rule, respectively, which have been detailed in Chapter 5.1.1. The relationship between three variables has been defined in Eq. 5.2. Based on the investigation through experimentation, the EF is initialised as 10,000 and the CD is always configured as 0, during the rule base initialisation.

In the rule base initialisation process, each variable value A_i is represented as an isosceles triangle fuzzy set and conveniently denoted as (a_{i1}, a_{i2}, a_{i3}), where (a_{i1}, a_{i3}) is the support of the fuzzy set and a_{i2} is the normal point. In this work, the support of the input fuzzy set is set to 5, and the support of the output fuzzy set is set to 1. Assume that a collected data instance contains a crisp input α and a crisp output v; the input fuzzy sets of A_i and the output fuzzy set B_i can be formed as:

$$\begin{aligned} A_i &= (\alpha - 2.5, \alpha, \alpha + 2.5) \\ B_i &= (v - 0.5, v, v + 0.5) \; . \end{aligned} \tag{9.6}$$

Note that the span of support of the fuzzy sets is defined based on a practice theory. A further study of the transformation from a crisp value to a fuzzy set remains for future work.

The rule base initialisation process creates two fuzzy rules that comprise the initialised rule base, as given below:

$$\begin{aligned} &R_1 : \textbf{IF } \alpha \text{ is } A_1,\ \textbf{THEN } v \text{ is } B_1\ (1,\ 10000,\ 0) \\ &R_2 : \textbf{IF } \alpha \text{ is } A_2,\ \textbf{THEN } v \text{ is } B_2\ (1,\ 10000,\ 0)\ . \end{aligned} \tag{9.7}$$

9.2.2.2 Rule Base Generation and Adaptation

Once the rule base has been initialised, the system comes into the second phase, the rule base generation and adaptation, which creates the fuzzy rule base while performing inference. The framework of this system is guided by the previously proposed method in Chapter 5, as outlined in Fig 5.1. Briefly, when the stabilisation controller takes over the balancing control, such as $-20 \leq \alpha \leq 20$, the current angle of the pendulum (α) is retrieved from the pendulum optical encoder based on Eq. 9.1, which is used as an input for the fuzzy control system. And then, based on a particular set of metrics, including the EF factor, the CD factor, and the distance between the input and the rule antecedents, the two 'best' suitable rules are selected from the current rule base for interpolative reasoning. From this, the KH approach is employed to interpolate a new rule. After that, a motor voltage signal is generated and sent to the load gear shaft module to actuate the arm, thus to sway the pendulum. Afterwards, the *performance index* is retrieved to support the rule base revision, including adding, removing, and modifying rules.

1) *Rule selection:* The KH approach requires two fuzzy rules to perform the interpolative reasoning. In this work, the *importance factor* (*IF*), which is described by Eq. 5.3 in Chapter 5.1.2, is also adopted in the process of rule selection. Note that it will be the case that all existing rules in the rule base are on the same side of the given input. In this special case, two rules with the first and second higher *IF* values in the current rule base are selected, and the KH extrapolation approach is used to perform the fuzzy extrapolative reasoning.

2) *Fuzzy interpolation:* Once rules have been selected, the fuzzy interpolation can be performed. As introduced in Chapter 2.1.1, the KH approach is able to handle both interpolation and extrapolation, as well as light computational complexity. Therefore, this approach is employed in this system. Given a crisp input, which is an angle of the pendulum α_i, Eq. 9.6 is used to fuzzify the crisp input to the fuzzy input. From this, a fuzzy output can be generated. Finally, a centre of gravity (COG) defuzzification technique [36] is applied to transfer a fuzzy output to a crisp number.

3) *Rule base revision:* The fuzzy rule base of the stabilisation controller is generated and adapted by the rule base revision process, which is outlined in Fig. 9.3. In particular, a motor

input voltage signal, which is generated by the KH approach, drives the motor rotating to actuate the arm and consequently to sway the pendulum. The system captures the current pendulum angle α and previous pendulum angle $\hat{\alpha}$ from the pendulum optical encoders for comparison, thus to obtain the *performance index*. A positive *PI* reflects that the decision generated by the KH approach is suitable for the current status of the pendulum, which leads to the pendulum moving closer to the upright position. Such an interpolated rule will be used for the rule base revision. Conversely, a negative *PI* indicates that the interpolated result causes the pendulum to come into an imbalanced status, which leads the pendulum away from the upright position. As a consequence, an interpolated rule with a negative *PI* will be ignored.

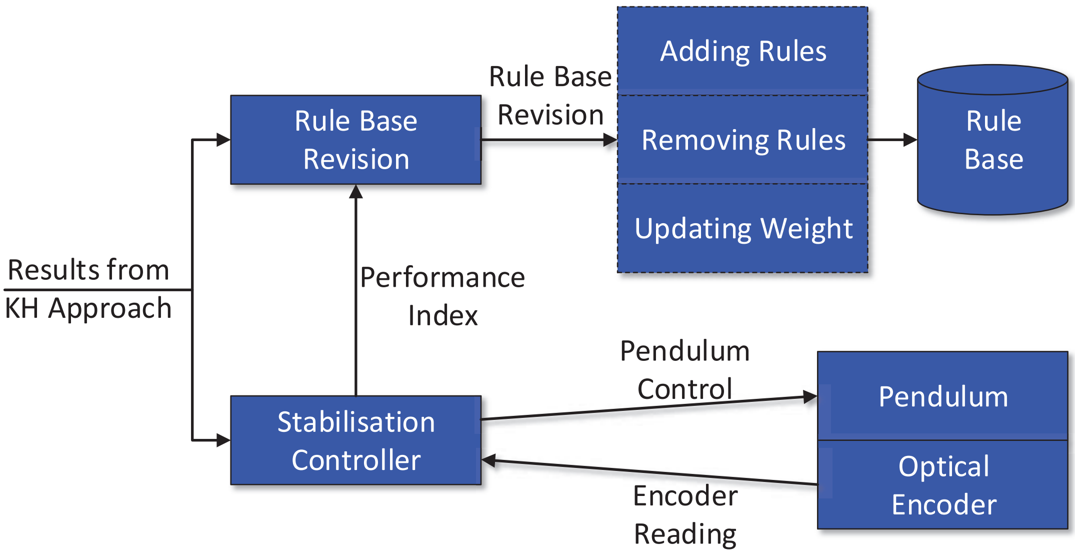


Fig. 9.3 The framework of the rule base revision

Given an interpolated rule (R_i) with its *PI* (PI_i), the rule base is able to be revised from three aspects: adding rules, removing rules, and updating weight. The working progress of the rule base revision has precisely followed the principle outlined in Fig. 5.2, which has been presented in Chapter 5.1.4. Therefore, the details are omitted here. At the beginning of the deployment of the system, the interpolated results may not be good enough to stabilise the pendulum, because the initiated rules may not be sufficiently great to support the KH approach for a specific output generation. Despite this, a decision can still be predicted for stabilisation control. Along with the performing of the fuzzy interpolation and the process of the rule base revision, the rule base will adaptively fit the requirement of the stabilisation control, and thus support the KH approach to generate more accurate results to balance the pendulum.

9.3 Results and Discussion

The evaluation of the proposed automatic rule base generation and adaptation approach by rotary inverted pendulum is carried out in this section.

9.3.1 Experimentation Environment

The QUARC, a real-time control software library for Matlab, has been released by the manufacturer, which provides an easy way to control the pendulum model [2]. From there, this system is developed based on the released library under the Matlab platform. The experimental set up of the system is illustrated in Fig. 9.4. The data acquisition board

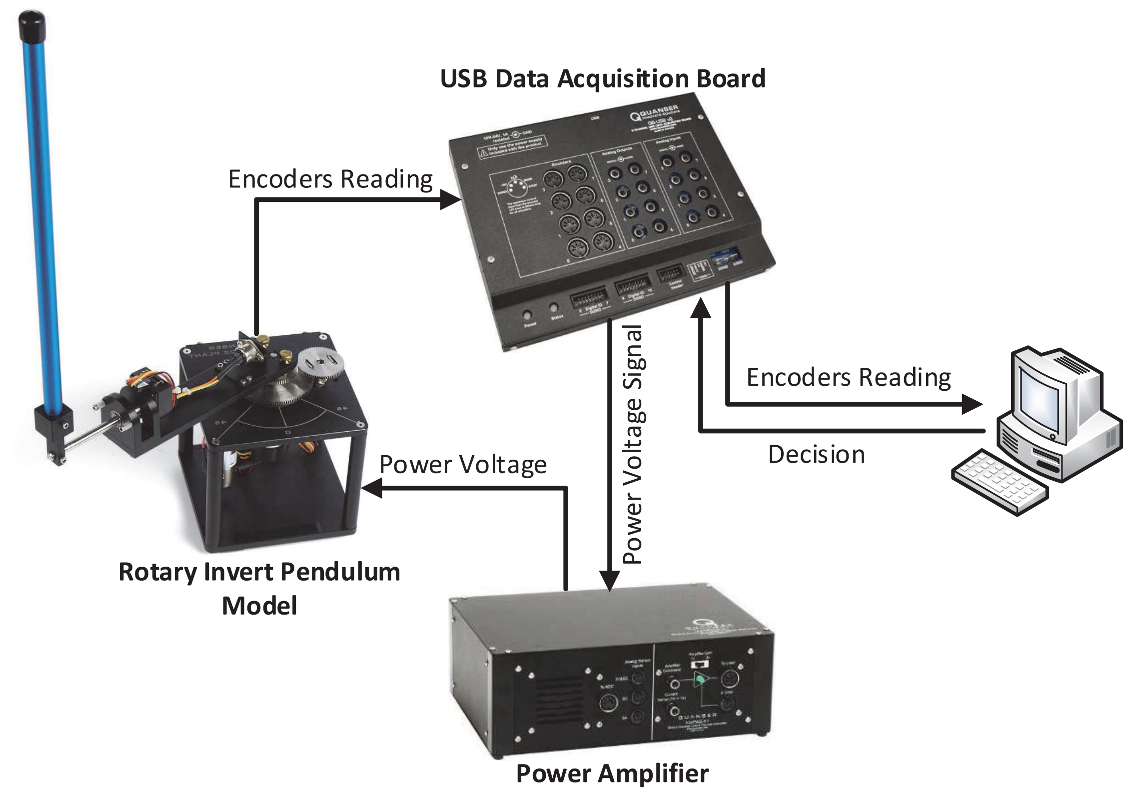


Fig. 9.4 The system connection between modules

transfers a decision signal, which is generated by the computer application system, to the power amplifier. Based on this signal, the power amplifier is able to generate a power voltage, which actuates the motor to rotate the arm. The measurements of the angles, including the pendulum and arm, are sent to the computer application system via the data acquisition board for decision prediction. The used techniques and the configurations of involved parameters in this evaluation are listed in Table 9.2. In addition, during the implementation, both the arm and the pendulum are free to rotate 360°. In order to avoid entanglement with the connecting cables, the system is configured to automatically rotate the arm back to the 0° position when it has made three rotations, such as $\theta > 1080°$ or $\theta < -1080°$.

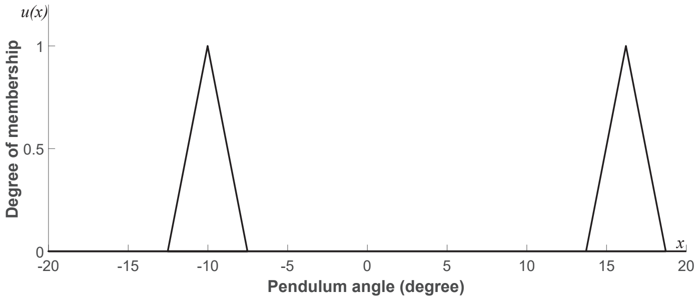
Table 9.2 The configuration of experimentation

Description	**Details**
Environment of development	Matlab 2016a
Development Library	QUARC 2.5
Type of the fuzzy model	Mamdani fuzzy model
Inference engine	KH fuzzy interpolation approach
Type of membership function	Triangle membership function
Defuzzification method	Centre of gravity (COG)
Experience factor	$n = 200$
Cooling down factor	$a = 100$, $b = 100$
Starting angle of pendulum	180°
Starting angle of arm	Any
The sampling time	0.001 second

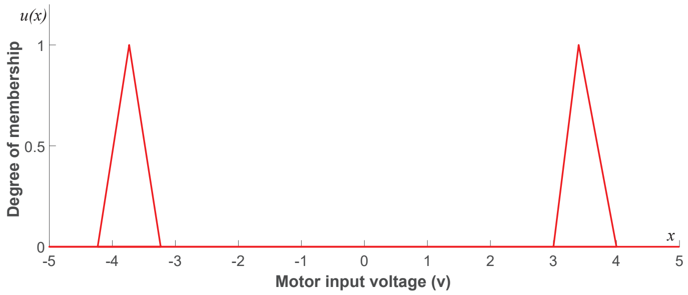
The rotary inverted pendulum is started from the swing up phase, which is controlled by the swing up controller. Note that the swing up strategy used in this work is a well-developed system, which has been proposed in [104]. Therefore, the swing up system will not be covered in this section.

9.3.2 Rule Base Initialisation

The rule base of the stabilisation controller is initialised based on a random try method. The system started from an empty rule base, and after a number of tries, two rules were initialised, as shown in Fig. 9.5. Note that Fig. 9.5 shows one example of the initialised rule base as the different rules will be initialised in different experimentations.



(a) Input - Angle of pendulum



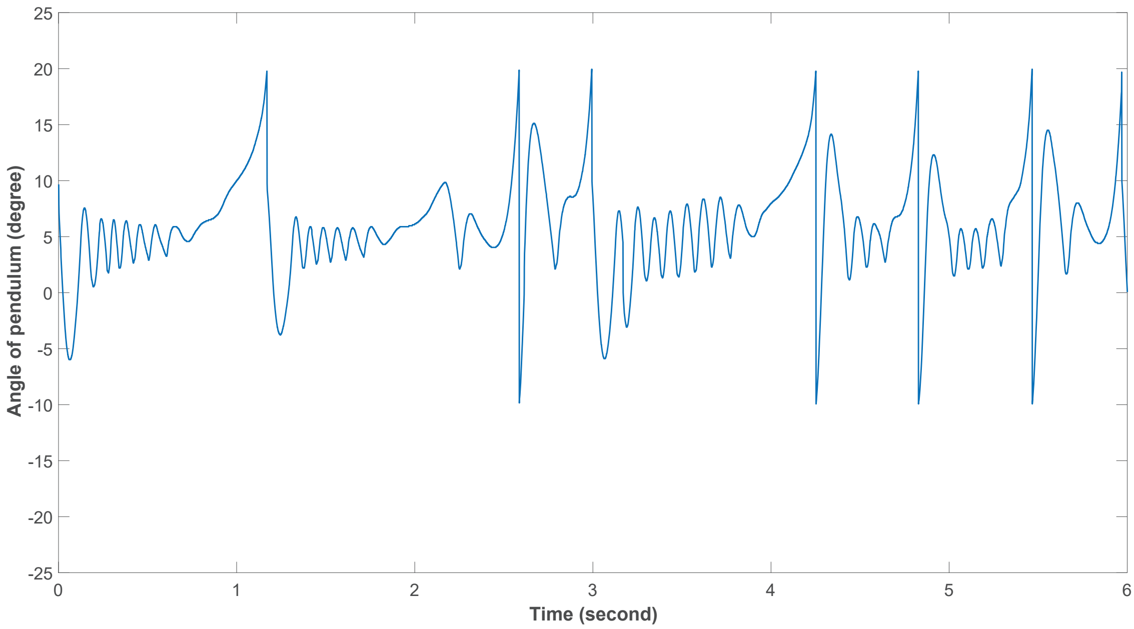
(b) Output - Motor input voltage

Fig. 9.5 The initialised rule base

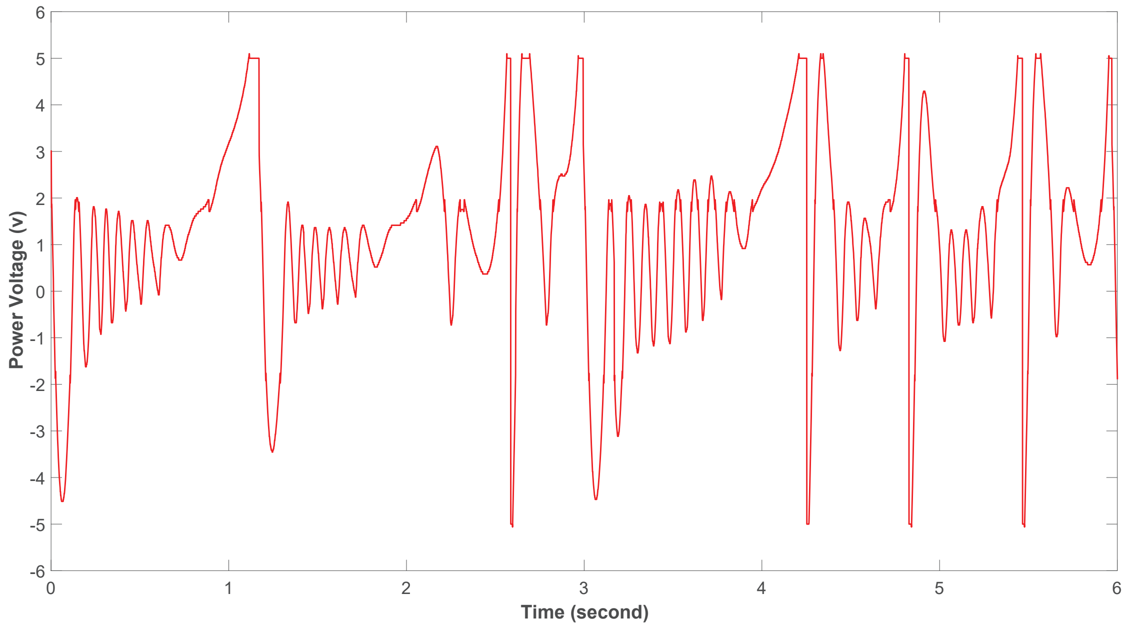
9.3.3 Rule Base Generation

From the initialised rule base, the KH approach is able to predict the output, which supports the rule base generation and revision. Based on the experimental results, the entire progresses of the rule base generation went through three stages: the initial learning stage, the adaptation stage, and the stabilisation stage.

1) *Initial learning stage:*This is a beginning stage of the rule base generation. In this stage, although the KH approach is able to generate a decision to actuate the rotary arm to sway the pendulum, the predicted decision is not accurate enough to support the pendulum to be balanced as the system extremely lacks the necessary knowledge. Figs. 9.6(a) and 9.6(b) take an example of a short period of time (six seconds) to illustrate the status of the pendulum and the corresponding motor input voltage. Note that the pendulum angle equates to 20° and represents the pendulum falling down, and the swing up controller takes over the progress to swing it up again. General speaking, the decision made by the KH approach based on the current rule base is not able to stabilise the pendulum, although the pendulum is able to be kept at the upright position for a very short period of time (0.5 seconds).



(a) Input - Angle of pendulum

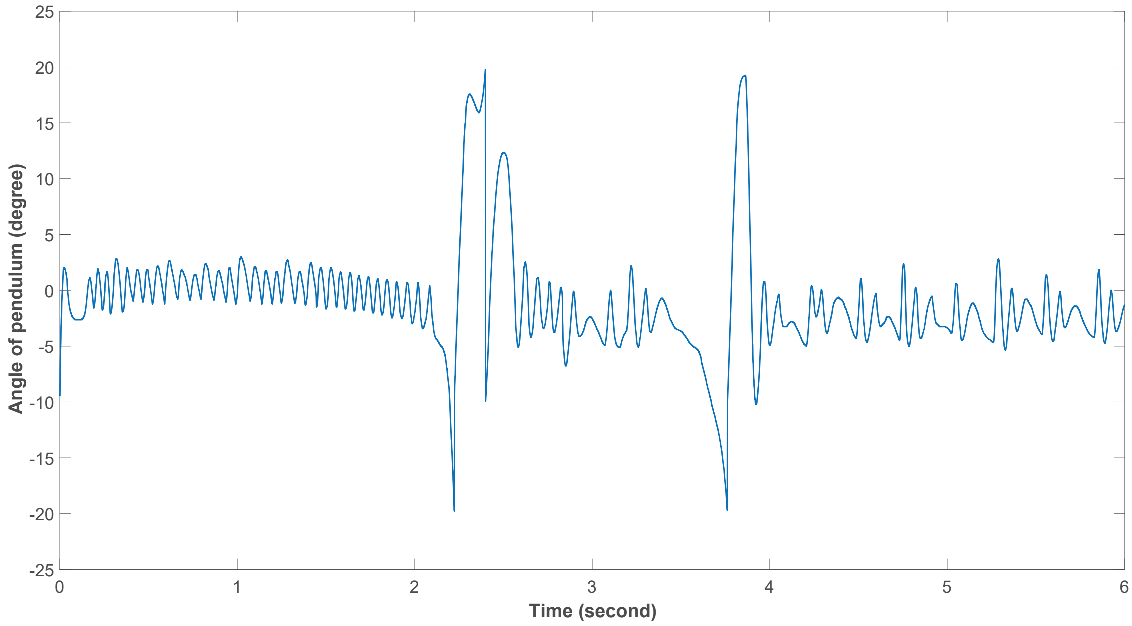


(b) Output - Motor input voltage

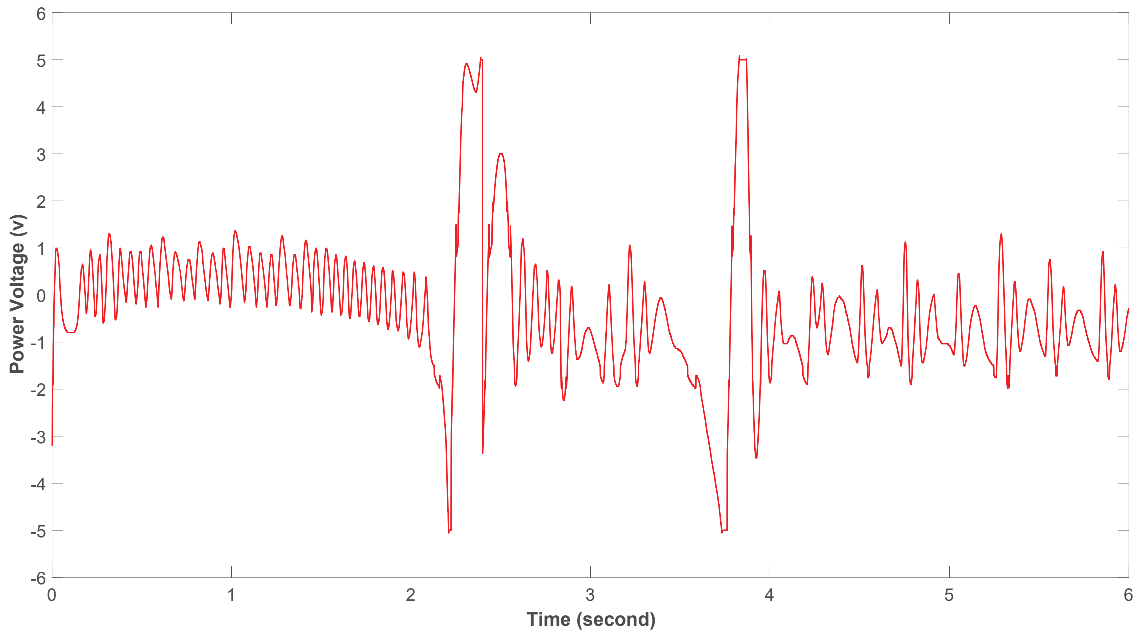
Fig. 9.6 The experimental results of initial learning stage

2) *Adaptation stage:* In this stage, a number of rules have been generated, and some of these have been adapted to fit the situation. Fig. 9.7 demonstrates the status of the stabilisation controller. The experimental results confirmed that the current rule base is able to support

the fuzzy control system to balance the pendulum for much longer (almost two seconds), compared with the previous stage (0.5 seconds).



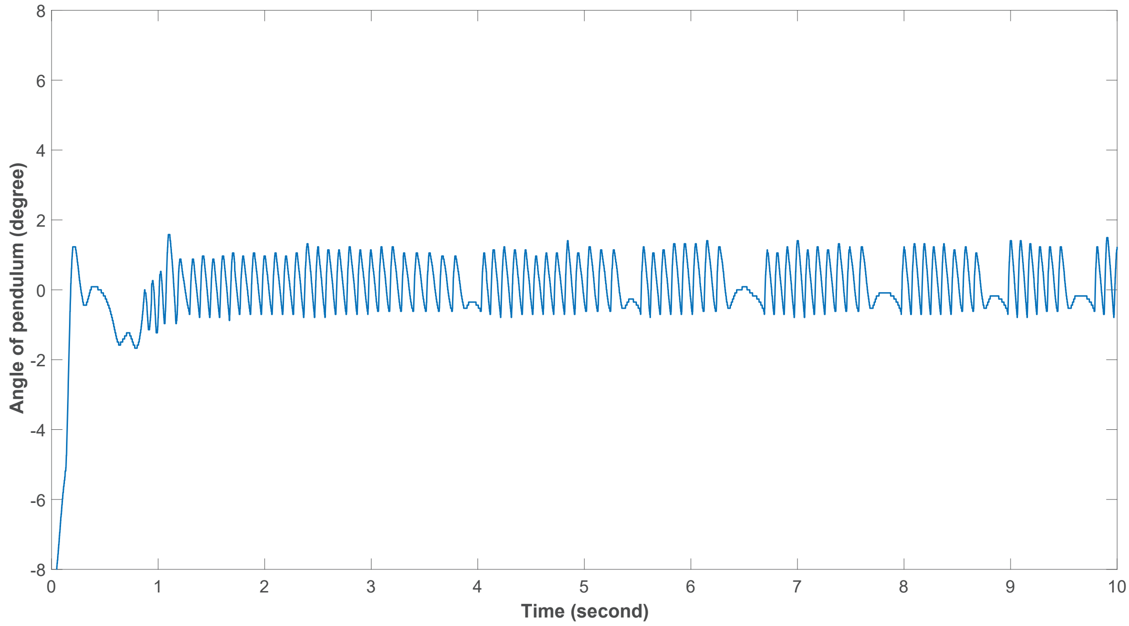
(a) Input - Angle of pendulum



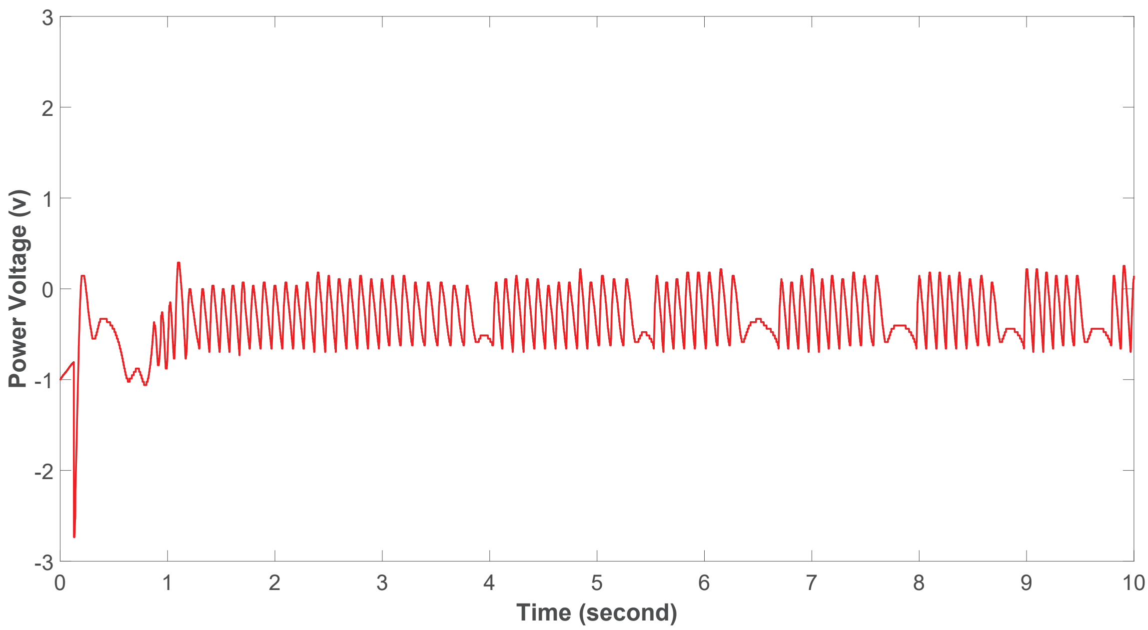
(b) Output - Motor input voltage

Fig. 9.7 The experimental results of adaptation stage

3) *Stabilisation stage:* In this stage, a reasonable rule base has been generated and the pendulum can be stabilised within a range of $[-1°, 1°]$, which has been illustrated in Fig. 9.8. The generated rule base is also shown in Fig. 9.9.

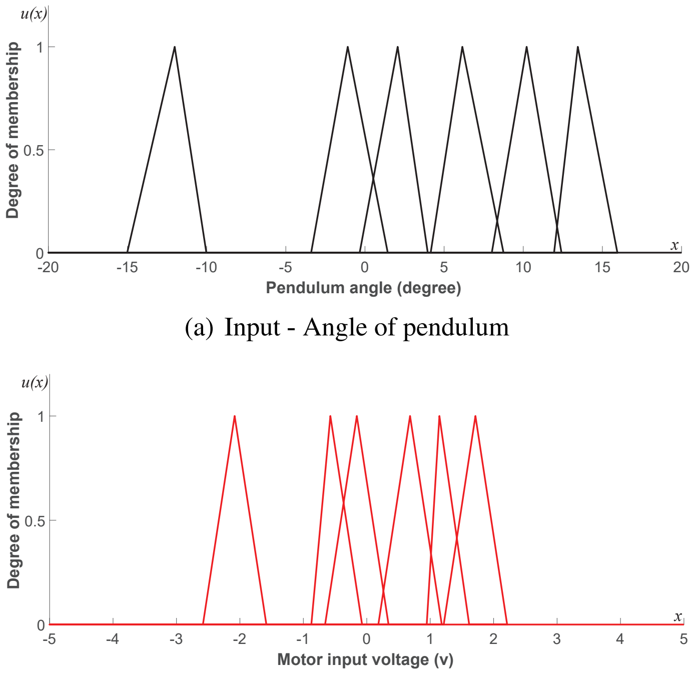


(a) Input - Angle of pendulum



(b) Output - Motor input voltage

Fig. 9.8 The experimental results of stabilisation stage



(a) Input - Angle of pendulum

(b) Output - Motor input voltage

Fig. 9.9 The final rule base

9.3.4 Discussion

The experimental results show that a reasonable fuzzy rule base can be generated from limited prior knowledge, as long as a performance index is able to be obtained. In this experimentation, the expert knowledge was only adopted to define the problem domain, and the entire progress of rule base generation was achieved by an experience-based learning method, which was presented in Chapter 5. In particular, the system started from an empty rule base, and the rule base has been gradually generated while the performing during the three stages of the learning, the initial learning stage, the adaptation stage, and the stabilisation stage. In the initial learning stage, the system lacks the knowledge for the decision making. Therefore, the pendulum can only be balanced at the upright position for around 0.5 seconds. In the adaptation stage, a number of rules have been generated. Such rules are able to support the fuzzy control system to balance the pendulum for much longer, which is almost 2 seconds. The rule base keeps revising itself, and finally, six rules have been generated, and the pendulum can be stabilised at the upright position.

According to the Fig. 9.8(b), it is interesting that there is a deviation in the output of the final generated rule base, in which a negative voltage has to be generated to keep the pendulum stabilised. This may be caused by an unstable surface or a non-horizontal position of the rotary inverted pendulum model. It is worthwhile to investigate how this deviation may be removed. In addition, the proposed rule base generation and adaptation method is able to adapt itself to fit the changing situations, which is provided by involving a cooling down factor. However, in this implementation, if the pendulum is stabilised at an upright position for a long period of time, two or three rules will always be used by the KH approach to predict the output, and the rest of the rules will not be selected at all. As a consequence, the cooling down factor will efficiently remove rules that were not selected continuously. Therefore, a mechanism is necessary to be considered to avoid such an issue.

9.4 Summary

This chapter presented an application of fuzzy interpolation with the previously presented rule base generation and adaptation method, which provides a solution to a well-known control problem, the inverted pendulum. In particular, the stabilisation control system initialises the rule base by generating two rules based on a random try mechanism. From this, the KH approach is employed to predict a decision, which generates a voltage to the motor, thus to actuate the rotary arm. At the same time, the system controller measures the angle of the pendulum to obtain a performance index. Such generated performance indexes will guide the rule base revision process to modify the rule base. The system will keep trying and learning while performing, until a reasonable rule base is generated. The experimental results confirmed that the proposed experience-based fuzzy rule base generation method can be used to deal with some real-world problems, which will increase the applicability of fuzzy interpolation techniques.

Chapter 10

Conclusion

This chapter concludes the thesis. A summary of the research, as detailed in the preceding chapters, is presented, with a focus on the main contribution: fuzzy interpolation systems with knowledge extraction and adaptation. The thesis has demonstrated that the developed TSK+ fuzzy interpolation approach has effectively extended the fuzzy interpolation techniques to deal with the TSK fuzzy model and its real-world applications, e.g. the network intrusion detection system and the network traffic management system. The thesis has also presented and evaluated the fuzzy rule base generation and adaptation approach that relaxed the requirement of the conventional rule base generation mechanism, which requires available knowledge. The developed applications have been experimentally evaluated and validated, which have confirmed the practicalities and applicabilities of the developed frameworks. A number of preliminary suggestions and future works have also been pointed out in conclusion.

10.1 Summary of Thesis

This thesis primarily concerned with fuzzy interpolative reasoning and its knowledge extraction. Fuzzy interpolative reasoning is a special type of fuzzy inference, which is not only able to handle an incomplete knowledge base, but to reduce the complexity of existing complex systems by using a more complexed rule base. However, all of the existing fuzzy interpolation techniques were developed based on the Mamdani fuzzy model, and none is able to deal with sparse TSK rule bases. It is, therefore, desirable to have a method to handle the TSK fuzzy model when the knowledge base is incomplete. The TSK+ fuzzy inference has been proposed here in filling such a gap, which allows fuzzy inferences to still be performed over a sparse TSK rule base.

A fuzzy knowledge base is a key component of a fuzzy inference system, which is usually extracted from the existing knowledge, i.e. human expert knowledge or existing data. If a priori knowledge is extremely lacking on a certain case, it will be difficult to apply the fuzzy inference approaches to deal with it. In addition, due to the fixed rule base generated by conventional methods, the difficulty in dealing with changing situations significantly restricts the applicability of the fuzzy inference systems. Therefore, it is worthwhile to suggest a mechanism to cover such deficiencies. The experience-based knowledge base generation and adaptation approach has been presented in this thesis to relieve the requirement of a priori knowledge during the system modelling.

Before presenting the proposed approaches, a detailed literature review of the fuzzy interpolative reasoning and conventional rule base generation has been given in Chapter 2. In particular, two groups of fuzzy interpolation approaches, i.e. resolution principle-based interpolation and analogy-based interpolation approaches, have been reviewed. Each group has been examined with a representative approach, as well as its extensions and improvements in detail. Besides fuzzy interpolation techniques, conventional rule base generation methods have also been systematically introduced, e.g. knowledge-driven and data-driven approaches.

In Chapter 3, the proposed TSK+ fuzzy inference approach has been detailed. A new similarity measurement between two fuzzy sets has been designed, which measures the degree of similarity between a given observation and every individual rule. Such obtained similarity degrees are used as the matching degrees instead of the overlapped matching degrees from the conventional TSK inference method. Then, the TSK fuzzy model is extended using the generated matching degrees to derive crisp inference results. In order to support the proposed TSK+ fuzzy inference approach, a data-driven rule base generation method has also been presented in this chapter. The proposed approach has been verified and evaluated by two mathematical models and shows the outperformance.

The proposed TSK+ approach is not only able to deal with the type-1 TSK fuzzy rule base, but is also further extended to allow the utilisation of interval type-2 TSK fuzzy rule bases. Chapter 4 detailed such extension. One illustrative case based on an example problem from the literature demonstrates the working of the proposed system, and the application on the cart centring problem reveals the power of the proposed system. The experimental investigation confirmed that the proposed TSK+ approach is able to perform fuzzy inferences using either dense or sparse interval type-2 TSK rule bases, with promising results generated.

A novel rule base generation and adaptation method that allows the creation of rule bases with minimal a priori knowledge has been theoretically developed in Chapter 5. The proposed system is supported by fuzzy interpolation techniques, which are able to add accurate interpolated rules into the rule base. In particular, a feedback mechanism has been introduced in this approach to obtain a performance index, which indicates whether the interpolated result is acceptable or not. In addition, the performance experience of rules, including rules' usage frequency information and historical performance information, has been employed to guide the system to fit the dynamic changing situations adaptively. Two simulation experimentations show that the proposed method can automatically generate a rule base from limited knowledge, as well as adapt to a changed situation.

Chapter 6 has presented a real-world application of the T-FRI approach to the problem of the smart home heating control system. In particular, a fuzzy inference system was designed to efficiently preheat the home by successfully predicting users' home time, which was supported by users' historical and current location data from portable devices. From this, the T-FRI approach was applied to reduce the complexity of the originally designed fuzzy inference system. The proposed method has been applied to a real-world case and the promising result has been generated.

The proposed TSK+ fuzzy inference has been validated in Chapter 7 by applying a practical application of the network intrusion detection system (NIDS). The proposed implementation was built upon the TSK+ inference approach and a sparse 0-order TSK rule base, which is generated by employing the introduced data-driven TSK rule base generation approach. A well-known benchmark dataset, KDD99, was used for system evaluation. Experimental results confirmed that by applying the TSK+ inference, the developed system not only guarantees the online performance of intrusion detection, but also allows the generation of security alerts from situations which are not directly covered by the existing knowledge base.

Apart from the network intrusion detection system, the TSK+ inference approach has also been employed in a real network environment to dynamically manage the overall performance of network services, which has been reported in Chapter 8. This system was developed based on the differentiated services (DiffServ) quality of service (QoS) architecture, which manages the priority levels of network services by modifying the value of the Differentiated Services Code Point (DSCP) in the Internet Protocol (IP) header of network packets. It is comprised of two main components: a 0-order sparse TSK rule base and the TSK+ inference engine. Compared with the proposed TSK+ NIDS, which adopted a data-driven approach for rule base generation, the rule base of the TSK+ QoS management system was transformed from human expert knowledge. The proposed system has been evaluated in a real network environment with promising results generated.

Finally, an application of fuzzy interpolation with the presented rule base generation and adaptation method to a real-world control problem, the inverted pendulum, has been detailed in Chapter 9. The proposed system starts from an empty rule base and creates the rule base by adding accurate interpolated rules and revising existing rules, which are guided by performance indexes generated by the system control module. The experimental results showed that a reasonable rule base can be produced to support fuzzy interpolative reasoning to make proper decisions, thus to balance the pendulum, which not only validated and evaluated the theoretical development of the experience-based rule base generation and adaptation method but also pointed out a future research direction.

In summary, two main contributions have been made during the development of this PhD research project: 1) the development of the TSK+ fuzzy inference approach that allows TSK fuzzy inference to still be performed over sparse TSK rule bases, and 2) the proposal of a fuzzy rule base generation and adaptation method that is able to create a reasonable rule base while performing inference, as well as adapt to a changed situation. Both developed theories have been validated and evaluated by real-world applications with promising results generated. However, further research is also required to enhance the proposed systems.

10.2 Future Works

Although promising, much can be done to further improve the works presented in the thesis. The following subsections address a number of interesting issues whose successful solutions will help to strengthen the current research and approaches.

10.2.1 Short-term Developments

This subsection discusses extensions and tasks that could be readily implemented if additional time were available.

10.2.1.1 TSK Sparse Rule Base Generation with Other Clustering Algorithms

The data-driven TSK rule base generation approach has been proposed to support the TSK+ inference engine, which is implemented by using a two-level clustering architecture, the sparse K-Means and K-Means clustering algorithms, to divide the given training dataset into multiple rule clusters. Although the combination of two clustering techniques is working well in the experimentation, additional processes are always required for the number of clusters determination. Fortunately, a sparsity-aware possibilistic clustering algorithm (SPCM) was proposed [198, 199], which has the ability to estimate the actual number of clusters. Therefore, it is desired to investigate how this clustering algorithm can support the proposed TSK+ inference system. In addition, it would also be interesting to define the desirable sparsity or density of a given dataset during the clustering process, e.g. the density-based clustering algorithm [61, 103], thus to achieve more accurate clustering results during the rule cluster generation.

10.2.1.2 TSK+ Inference with Gaussian Membership Function

The proposed TSK+ was developed based on a newly designed similarity measurement, which is only workable with polygonal membership functions (MF), e.g. triangular, trapezoidal and hexagonal MFs. In fact, Gaussian MF is another popular method for system modelling, because it is usually preferred for a smooth transition and simple adaptability. As such, it would be interesting to implement the TSK+ with the use of Gaussian MF. Actually, a number of Gaussian MFs extraction methods, such as [12, 40, 139], and the similarity degree measurement between two Gaussian MFs, e.g. [117], have been proposed in the literature, which will provide the guidance for such implementation.

10.2.1.3 Termination Condition of Experience-based Rule Base Generation

The proposed experience-based rule base generation and adaptation approach is able to automatically create a rule base as well as adapt to a changed situation. However, according to experimental results discussed in Chapter 9.3.4, the adaptation process will not be terminated itself if a reasonable rule base has been generated. As a result, some important rules may be removed if they have not been involved in fuzzy interpolative reasoning for a long period of time. Therefore, the entire domain coverage rate, the average similarity degree between input membership functions, and the system performance may be jointly considered as a termination condition, which could be used to indicate whether a reasonable rule base has been generated or not. For instance, given a threshold, if the value of the termination condition is higher then the threshold, the rule base adaptation process will be terminated.

10.2.2 Longer-term Developments

This section proposes several future directions of research.

10.2.2.1 Higher-order TSK+ Inference

Currently, the proposed TSK+ inference approach is able to deal with 0-order and 1-order type-1 and interval type-2 TSK fuzzy models. Compared with 0-order and 1-order TSK fuzzy models, the higher-order TSK fuzzy models can reduce drastically the number of rules needed to perform the approximation and improve transparency and interpretation in many high-dimensional systems [75, 157]. Therefore, it would be ideal if the TSK+ inference approach could be extended in the future to support the higher-order TSK fuzzy model.

10.2.2.2 TSK+ and Experience-based Rule Base Generation

The proposed experience-based rule base generation approach has been implemented by the T-FRI and the KH fuzzy interpolation approaches, which are developed based on the Mamdani fuzzy model. Although a data-driven TSK rule base generation approach has been presented in this thesis, it requires training datasets for the system modelling. It would be interesting to integrate the TSK+ approach and the experience-based rule base generation approach to enable the TSK rule base automatic generation. It would significantly increase the applicability of the TSK fuzzy model.

10.2.2.3 More Real-world Applications

The fuzzy interpolation approach has been successfully employed to a smart home heating management system, presented in Chapter 6, where the rule base was predefined based on the historical data of a particular property and its residents and thus it is only able to deal with fixed predefined situations. The proposed experience-based rule base generation and adaptation approach may provide solutions for such real-world problems. Two benefits will take place if the proposed system is applied. Firstly, only the most general/common a priori knowledge is required to initialise the system; thus, the heating management system can be mass-produced (commercialised). Secondly, the system will be able to handle changing situations such as changes of radiators, residents or their living styles; thus, the model is highly adaptable. As such, it would be worthwhile to investigate how the proposed rule base generation approach can be integrated to support such real-world applications, thus to improve our quality of life.

Appendix A

Initialised Rule Base of the proposed NIDS in Chapter 7

Assume that a 0-order TSK fuzzy rule R_i $(1 \leq i \leq n)$, where n is the total number of generated fuzzy rules, can be expressed as:

$$\begin{aligned} R_i : &\ \textbf{IF}\ x_1 \text{ is } A_{i1} \text{ and } x_2 \text{ is } A_{i2} \text{ and } x_3 \text{ is } A_{i3} \text{ and } x_4 \text{ is } A_{i4} \\ &\ \textbf{THEN}\ y = f_i, \end{aligned} \tag{A.1}$$

where A_{i1}, A_{i2}, A_{i3} and A_{i4} are triangular fuzzy sets, which can be represented as $A_{i1} = (a_{(i1)1}, a_{(i1)2}, a_{(i1)3})$, $A_{i2} = (a_{(i2)1}, a_{(i2)2}, a_{(i2)3})$, $A_{i3} = (a_{(i3)1}, a_{(i3)2}, a_{(i3)3})$, and $A_{i4} = (a_{(i4)1}, a_{(i4)2}, a_{(i4)3})$, and f_i is an integer number that maps either a type of attack or the normal traffic. Note that the mapping details have been listed in Table 7.3. The initialised 46 TSK fuzzy rules are detailed below:

i	Input												Output
	A_{i1}			A_{i2}			A_{i3}			A_{i4}			f_i
	$a_{(i1)1}$	$a_{(i1)2}$	$a_{(i1)3}$	$a_{(i2)1}$	$a_{(i2)2}$	$a_{(i2)3}$	$a_{(i3)1}$	$a_{(i3)2}$	$a_{(i3)3}$	$a_{(i4)1}$	$a_{(i4)2}$	$a_{(i4)3}$	
1	0.000	0.210	0.714	0.000	0.228	0.438	0.000	0.100	1.000	0.000	0.000	1.000	1
2	0.000	0.212	0.711	0.000	0.520	0.787	0.025	0.100	0.599	0.000	0.000	1.000	1
3	0.000	0.233	0.450	0.805	0.833	0.834	0.025	0.025	0.125	0.000	0.020	0.430	1
4	0.723	0.760	0.843	0.000	0.000	0.723	0.025	0.050	0.050	0.000	0.040	1.000	1
5	0.833	0.833	0.833	0.000	0.000	0.788	0.025	0.050	0.150	0.000	0.040	1.000	1
6	0.834	0.834	0.834	0.689	0.830	0.833	0.025	0.025	0.050	0.040	0.185	0.360	1
7	0.834	0.834	0.834	0.000	0.670	0.761	0.025	0.025	0.075	0.000	0.090	0.610	1
8	0.836	0.836	0.836	0.561	0.833	0.833	0.025	0.025	0.025	0.030	0.055	0.080	1
9	0.848	0.848	0.848	0.834	0.834	0.834	0.025	0.025	0.025	0.030	0.030	0.030	1
10	0.837	0.837	0.837	0.682	0.687	0.689	0.025	0.025	0.025	0.030	0.030	0.030	1
11	0.835	0.835	0.835	0.000	0.747	0.783	0.025	0.050	0.050	0.090	0.160	0.920	1
12	0.588	0.599	0.601	0.000	0.387	0.414	0.050	0.075	0.100	0.000	0.000	0.000	2
13	0.000	0.000	0.263	0.000	0.000	0.047	0.554	0.584	0.623	0.000	0.070	1.000	2
14	0.000	0.000	0.272	0.000	0.000	0.000	0.025	0.524	0.554	0.000	0.050	0.760	2
15	0.593	0.601	0.601	0.468	0.487	0.487	0.025	0.075	0.725	0.000	0.000	0.000	2
16	0.000	0.000	0.263	0.000	0.000	0.000	0.624	0.663	0.710	0.000	0.070	0.750	2
17	0.580	0.585	0.587	0.414	0.468	0.487	0.050	0.100	0.150	0.000	0.000	0.000	2
18	0.554	0.568	0.579	0.360	0.401	0.487	0.025	0.088	0.150	0.000	0.000	0.000	2
19	0.352	0.352	0.352	0.000	0.000	0.000	0.025	0.543	0.605	0.000	0.020	0.260	2
20	0.352	0.352	0.352	0.000	0.000	0.000	0.752	1.000	1.000	0.000	0.020	0.260	2
21	0.360	0.360	0.520	0.000	0.000	0.330	0.025	0.050	0.659	0.000	0.020	1.000	2
22	0.352	0.352	0.352	0.000	0.000	0.000	0.606	0.667	0.751	0.000	0.000	0.260	2
23	0.263	0.263	0.263	0.000	0.000	0.000	0.711	0.970	0.993	0.000	0.000	0.000	2
24	0.524	0.535	0.549	0.360	0.487	0.487	0.050	0.100	0.150	0.000	0.000	0.000	2
25	0.000	0.000	0.362	0.406	0.439	0.498	0.025	0.025	0.075	0.000	0.000	0.020	3
26	0.000	0.149	0.219	0.000	0.296	0.371	0.025	0.025	0.100	0.000	0.000	0.750	3
27	0.215	0.362	0.365	0.368	0.384	0.390	0.025	0.025	0.599	0.000	0.000	0.060	3
28	0.376	0.378	0.382	0.404	0.405	0.417	0.025	0.025	0.050	0.000	0.000	0.000	3
29	0.244	0.244	0.244	0.631	0.631	0.631	0.025	0.025	0.025	0.000	0.000	0.000	3
30	0.365	0.407	0.449	0.531	0.538	0.545	0.025	0.025	0.025	0.000	0.010	0.020	3
31	0.000	0.229	0.358	0.000	0.000	0.362	0.025	0.025	0.100	0.000	0.000	0.670	4
32	0.000	0.356	0.359	0.371	0.378	0.573	0.025	0.025	0.050	0.000	0.020	1.000	4
33	0.000	0.305	0.361	0.754	0.771	0.787	0.025	0.025	0.025	0.000	0.000	0.000	4
34	0.000	0.000	0.000	0.850	0.833	0.833	0.025	0.025	0.025	0.000	0.000	0.000	4
35	0.464	0.464	0.464	0.000	0.000	0.000	0.025	0.050	0.075	0.000	0.000	0.220	4
36	0.834	0.834	0.834	0.000	0.000	0.000	0.025	0.025	0.025	0.000	0.000	0.290	4
37	0.834	0.834	0.834	0.000	0.000	0.000	0.025	0.025	0.025	0.000	0.000	0.290	4
38	0.834	0.834	0.834	0.000	0.000	0.000	0.025	0.025	0.050	0.000	0.000	0.130	4
39	0.000	0.000	0.000	0.834	0.834	0.834	0.025	0.025	0.025	0.000	0.000	0.000	4
40	0.000	0.000	0.000	0.834	0.834	0.834	0.025	0.050	0.075	0.000	0.000	0.000	4
41	0.000	0.000	0.000	0.834	0.834	0.834	0.025	0.025	0.025	0.000	0.000	0.000	4
42	0.000	0.002	0.836	0.000	0.000	0.596	0.000	0.025	1.000	0.000	0.270	1.000	5
43	0.897	0.897	0.897	0.000	0.000	0.000	0.025	0.025	0.025	0.120	0.120	0.120	5
44	0.869	0.869	0.869	0.000	0.000	0.000	0.025	0.025	0.025	0.110	0.110	0.110	5
45	0.000	0.000	0.000	0.900	0.900	1.000	0.025	0.025	0.050	0.500	0.500	0.650	5
46	0.937	0.988	1.000	0.000	0.000	0.000	0.025	0.025	0.538	0.080	0.105	0.120	5

Appendix B

Optimised Rule Base of the proposed NIDS in Chapter 7

Assume that a 0-order TSK fuzzy rule R_i $(1 \leq i \leq n)$, where n is the total number of generated fuzzy rules, can be expressed as:

$$\begin{aligned} R_i : &\textbf{IF } x_1 \text{ is } A_{i1} \text{ and } x_2 \text{ is } A_{i2} \text{ and } x_3 \text{ is } A_{i3} \text{ and } x_4 \text{ is } A_{i4} \\ &\textbf{THEN } y = f_i, \end{aligned} \tag{B.1}$$

where A_{i1}, A_{i2}, A_{i3} and A_{i4} are triangular fuzzy sets, which can be represented as $A_{i1} = (a_{(i1)1}, a_{(i1)2}, a_{(i1)3})$, $A_{i2} = (a_{(i2)1}, a_{(i2)2}, a_{(i2)3})$, $A_{i3} = (a_{(i3)1}, a_{(i3)2}, a_{(i3)3})$, and $A_{i4} = (a_{(i4)1}, a_{(i4)2}, a_{(i4)3})$, and f_i is an integer number that indicates either a type of attack or the normal traffic. Note that the mapping details have been listed in Table 7.3. The optimised 46 TSK fuzzy rules are detailed below:

i	Input												Output
	A_{i1}			A_{i2}			A_{i3}			A_{i4}			f_i
	$a_{(i1)1}$	$a_{(i1)2}$	$a_{(i1)3}$	$a_{(i2)1}$	$a_{(i2)2}$	$a_{(i2)3}$	$a_{(i3)1}$	$a_{(i3)2}$	$a_{(i3)3}$	$a_{(i4)1}$	$a_{(i4)2}$	$a_{(i4)3}$	
1	0.000	0.199	0.332	0.000	0.210	0.350	0.000	0.157	0.261	0.000	0.157	0.262	1
2	0.000	0.138	0.184	0.000	0.124	0.166	0.000	0.178	0.226	0.237	0.675	0.868	1
3	0.000	0.167	0.331	0.000	0.176	0.333	0.746	0.782	0.774	0.175	0.480	0.750	1
4	0.333	0.453	0.667	0.317	0.455	0.700	0.000	0.081	0.225	0.000	0.109	0.289	1
5	0.348	0.567	0.731	0.000	0.000	0.000	0.000	0.116	0.204	0.000	0.086	0.150	1
6	0.475	0.787	0.829	0.301	0.599	0.602	0.000	0.235	0.237	0.000	0.260	0.276	1
7	0.602	0.873	0.995	0.300	0.529	0.632	0.000	0.164	0.226	0.000	0.173	0.250	1
8	0.750	0.780	0.950	0.000	0.026	0.175	0.000	0.071	0.474	0.000	0.036	0.238	1
9	0.831	0.912	1.000	0.873	0.932	0.995	0.000	0.108	0.214	0.000	0.120	0.249	1
10	0.833	0.889	1.000	0.000	0.013	0.158	0.000	0.024	0.304	0.000	0.021	0.262	1
11	0.916	0.936	0.998	0.317	0.503	0.632	0.000	0.154	0.249	0.000	0.148	0.250	1
12	0.000	0.152	0.175	0.000	0.118	0.135	0.748	0.920	0.945	0.000	0.252	0.289	2
13	0.572	0.701	0.774	0.000	0.152	0.317	0.788	0.866	0.998	0.000	0.132	0.276	2
14	0.632	0.896	0.950	0.000	0.305	0.368	0.000	0.197	0.249	0.000	0.218	0.263	2
15	0.500	0.717	0.752	0.000	0.259	0.317	0.746	1.000	1.000	0.000	0.203	0.236	2
16	0.745	0.753	0.775	0.000	0.307	0.349	0.080	0.081	0.072	0.000	0.000	0.000	2
17	0.461	0.515	0.691	0.275	0.334	0.404	0.040	0.119	0.145	0.000	0.000	0.000	2
18	0.323	0.498	0.695	0.432	0.498	0.579	0.018	0.086	0.155	0.000	0.000	0.000	2
19	0.281	0.324	0.380	0.000	0.000	0.000	0.026	0.586	0.680	0.000	0.018	0.459	2
20	0.000	0.049	0.158	0.000	0.057	0.184	0.474	0.637	1.000	0.000	0.078	0.250	2
21	0.298	0.377	0.427	0.000	0.000	0.320	0.027	0.038	0.671	0.000	0.009	0.760	2
22	0.177	0.228	0.442	0.000	0.000	0.000	0.577	0.631	0.648	0.000	0.000	0.168	2
23	0.244	0.270	0.326	0.000	0.000	0.000	0.665	0.765	0.836	0.000	0.000	0.000	2
24	0.792	0.848	0.948	0.000	0.000	0.735	0.000	0.090	0.249	0.000	0.095	0.263	2
25	0.000	0.000	0.303	0.365	0.394	0.647	0.021	0.022	0.073	0.000	0.000	0.025	3
26	0.551	0.831	0.919	0.602	0.904	1.000	0.000	0.200	0.263	0.000	0.181	0.238	3
27	0.543	0.815	0.842	0.831	1.000	1.000	0.000	0.214	0.226	0.000	0.249	0.263	3
28	0.000	0.157	0.333	0.498	0.667	0.831	0.000	0.102	0.238	0.500	0.745	1.000	3
29	0.150	0.263	0.276	0.525	0.641	0.705	0.023	0.026	0.033	0.000	0.000	0.000	3
30	0.320	0.437	0.461	0.477	0.498	0.758	0.022	0.027	0.033	0.000	0.014	0.022	3
31	0.000	0.144	0.150	0.833	1.000	1.000	0.000	0.072	0.249	0.000	0.076	0.263	4
32	0.000	0.357	0.466	0.296	0.428	0.838	0.026	0.028	0.036	0.000	0.022	0.755	4
33	0.000	0.154	0.367	0.633	0.746	0.903	0.000	0.100	0.226	0.000	0.099	0.236	4
34	0.000	0.019	0.175	0.000	0.031	0.286	0.000	0.028	0.250	0.000	0.029	0.262	4
35	0.300	0.428	0.528	0.000	0.000	0.000	0.035	0.042	0.085	0.000	0.000	0.249	4
36	0.831	1.000	1.000	0.000	0.096	0.150	0.000	0.152	0.237	0.000	0.160	0.238	4
37	0.386	0.477	0.663	0.000	0.047	0.150	0.000	0.082	0.249	0.000	0.078	0.238	4
38	0.665	0.846	1.000	0.664	0.930	1.000	0.000	0.141	0.262	0.000	0.134	0.248	4
39	0.000	0.144	0.150	0.833	1.000	1.000	0.000	0.072	0.249	0.000	0.076	0.263	4
40	0.000	0.000	0.000	0.663	0.661	1.000	0.032	0.036	0.040	0.000	0.000	0.000	4
41	0.000	0.000	0.000	0.671	0.701	0.807	0.030	0.023	0.030	0.000	0.000	0.000	4
42	0.000	0.000	0.175	0.000	0.000	0.166	0.000	0.000	0.250	0.750	0.750	0.815	5
43	0.000	0.160	0.166	0.000	0.152	0.158	0.000	0.228	0.238	0.525	0.930	0.948	5
44	0.000	0.023	0.166	0.000	0.026	0.175	0.750	0.765	0.857	0.522	0.606	0.950	5
45	0.831	0.857	0.857	0.000	0.146	0.150	0.000	0.219	0.238	0.000	0.458	0.473	5
46	0.000	0.003	0.158	0.875	0.881	0.987	0.000	0.006	0.275	0.275	0.285	0.788	5

Appendix C

Publications Arising from the Thesis

All publications are presented in chronological order.

C.1 Published or Accepted for Publication

- Z. Zuo, J. Li, B. Wei, L. Yang, F. Chao, and N. Naik. Adaptive Activation Function Generation for Artificial Neural Networks through Fuzzy Inference with Application in Grooming Text Categorisation. *In 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, June 2019.

- J. Li, Y. Qu, F. Chao, H. P. H. Shum, E. SL Ho, and L. Yang. Machine Learning Algorithms for Network Intrusion Detection. *In AI in Cybersecurity*, pp. 151-179. Springer, Cham, 2019.

- N. Elisa, J. Li, Z. Zuo, and L. Yang. Dendritic Cell Algorithm with Fuzzy Inference System for Input Signal Generation. *In UK Workshop on Computational Intelligence,* pp. 203-214. Springer, Cham, 2018. (Best Paper Award)

- L. Yang, J. Li, F. Chao, P. Hackney, M. Flanagan. Job Shop Planning and Scheduling for Manufacturers with Manual Operations. *Expert System*, 2018

- Z. Zuo, J. Li, P. Anderson, L. Yang, N. Nitin. Grooming Detection using Fuzzy-Rough Feature Selection and Text Classification. *In IEEE World Congress on Computation Intelligence International Conference*, 2018.

- J. Li, L. Yang, X. Fu, F. Chao, Y. Qu. Interval Type-2 TSK+ Fuzzy Inference System. *In IEEE World Congress on Computation Intelligence International Conference,* 2018.

- J. Li, L. Yang, Y. Qu, G. Sexton. An Extended Takagi-Sugeno-Kang Inference System (TSK+) with Fuzzy Interpolation and Its Rule Base Generation, *Soft Computing*, 22(10):3155-3170, 2018.
- J. Li, L. Yang, X. Fu, F. Chao, Y. Qu. Dynamic QoS Solution for Enterprise Networks Using TSK Fuzzy Interpolation. *In 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, July 2017.
- L. Yang, J. Li, G. Fehringer, P. Barraclough, G. Sexton, Y. Cao. Intrusion Detection System by Fuzzy Interpolation. *In 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, July 2017.
- L. Yang, J. Li, P. Hackney, F. Chao, M. Flanagan. Manual Task Completion Time Estimation for Job Shop Scheduling Using A Fuzzy Inference System. *The International Workshop on Engineering Data-Model-Driven Applications*, 2017.
- J. Li, Y. Qu, H. P. H. Shum, and L. Yang. TSK Inference with Sparse Rule Bases, pages 107–123. *Springer International Publishing*, Cham, 2017. (Best Paper Award)
- J. Li, H. P. H. Shum, X. Fu, G. Sexton, and L. Yang. Experience-based rule base generation and adaptation for fuzzy interpolation. *In 2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 102–109, July 2016.
- Y. Tan, J. Li, M. Wonders, F. Chao, H. P. H. Shum, and L. Yang. Towards sparse rule base generation for fuzzy rule interpolation. *In IEEE World Congress on Computation Intelligence International Conference*, 2016.
- J. Li, L. Yang, H. P. H. Shum, G. Sexton, and Y. Tan. Intelligent home heating controller using fuzzy rule interpolation. *In UK Workshop on Computational Intelligence*, 2015.

C.2 Published or Accepted for Publication

- L. Yang, J. Li, N. Elisa, and F. Chao. Towards Big Data Governance in Cybersecurity. *Data-Enabled Discovery and Applications*, Springer, 2018

References

[1] Akesson, J. and Astrom, K. J. (2001). Safe manual control of the furuta pendulum. In *Control Applications, 2001. (CCA '01). Proceedings of the 2001 IEEE International Conference on*, pages 890–895. IEEE.

[2] Apkarian, J. and Karam, P. (2011). *Quanser Rotary Inverted Pendulum Instructor Workbook*. Quanser Inc.

[3] Aranda-Escolástico, E., Guinaldo, M., Santos, M., and Dormido, S. (2016). Control of a chain pendulum: A fuzzy logic approach. *International Journal of Computational Intelligence Systems*, 9(2):281–295.

[4] Atefi, K., Yahya, S., Rezaei, A., and Hashim, S. H. B. M. (2016). Anomaly detection based on profile signature in network using machine learning technique. In *2016 IEEE Region 10 Symposium (TENSYMP)*, pages 71–76. IEEE.

[5] Back, T. (1996). *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press.

[6] Baker, J. E. (1985). Adaptive selection methods for genetic algorithms. In *Proceedings of an International Conference on Genetic Algorithms and their applications*, pages 101–111. Hillsdale.

[7] Balamurugan, S., Venkatesh, P., and Varatharajan, M. (2017). Fuzzy sliding-mode control with low pass filter to reduce chattering effect: an experimental validation on quanser srip. *Sādhanā*, 42(10):1693–1703.

[8] Baranyi, P., Kóczy, L. T., and Gedeon, T. D. (2004). A generalized concept for fuzzy rule interpolation. *IEEE Transactions on Fuzzy Systems*, 12(6):820–837.

[9] Baranyi, P., Tikk, D., Yam, Y., Koczy, L. T., and Nadai, L. (1999). A new method for avoiding abnormal conclusion for a-cut based rule interpolation. In *Fuzzy Systems Conference Proceedings, 1999. FUZZ-IEEE'99. 1999 IEEE International*, pages 383–388. IEEE.

[10] Beggas, M., Médini, L., Laforest, F., and Laskri, M. T. (2014). Towards an ideal service qos in fuzzy logic-based adaptation planning middleware. *Journal of Systems and Software*, 92:71–81.

[11] Bellaaj, H., Ketata, R., and Chtourou, M. (2013). A new method for fuzzy rule base reduction. *Journal of Intelligent & Fuzzy Systems*, 25(3):605–613.

[12] Bhatt, R. B., Narayanan, S. J., Paramasivam, I., and Khalid, M. (2012). Approximating fuzzy membership functions from clustered raw data. In *2012 Annual IEEE India Conference (INDICON)*, pages 487–492. IEEE.

[13] Bholowalia, P. and Kumar, A. (2014). Ebk-means: A clustering technique based on elbow method and k-means in wsn. *International Journal of Computer Applications*, 105(9):17–24.

[14] Bhuyan, M. H., Bhattacharyya, D. K., and Kalita, J. K. (2014). Network anomaly detection: methods, systems and tools. *IEEE Communications Surveys & Tutorials*, 16(1):303–336.

[15] Bostani, H. and Sheikhan, M. (2017). Modification of supervised opf-based intrusion detection systems using unsupervised learning and social network concept. *Pattern Recognition*, 62:56–72.

[16] Bouchon-Meunier, B., Esteva, F., Godo, L., Rifqi, M., and Sandri, S. (2005). A principled approach to fuzzy rule base interpolation using similarity relations. In *EUSFLAT–LFA 2005*, pages 757–763. EUSDLAR-LFA.

[17] Bouchon-Meunier, B., Marsala, C., and Rifqi, M. (2000). Interpolative reasoning based on graduality. In *Fuzzy Systems, 2000. FUZZ IEEE 2000. The Ninth IEEE International Conference on*, pages 483–487. IEEE.

[18] Bouchon-Meunier, B. and Valverde, L. (1999). A fuzzy approach to analogical reasoning. *Soft Computing*, 3(3):141–147.

[19] Boumella, N., Djouani, K., and Boulemden, M. (2012). A robust interval type-2 tsk fuzzy logic system design based on chebyshev fitting. *International Journal of Control, Automation and Systems*, 10(4):727–736.

[20] Bozdogan, H. (1994). *Mixture-model cluster analysis using model selection criteria and a new informational measure of complexity*, pages 69–113. Springer Netherlands, Dordrecht.

[21] Buckley, J., Siler, W., and Tucker, D. (1986). A fuzzy expert system. *Fuzzy Sets and Systems*, 20(1):1–16.

[22] Buckley, J. J. (1993). Sugeno type controllers are universal controllers. *Fuzzy Sets and Systems*, 53(3):299–303.

[23] Castro, J. L. (1995). Fuzzy logic controllers are universal approximators. *IEEE Transactions on Systems, Man, and Cybernetics*, 25(4):629–635.

[24] Cayrol, M., Farreny, H., and Prade, H. (1982). Fuzzy pattern matching. *Kybernetes*, 11(2):103–116.

[25] Cazzolato, B. S. and Prime, Z. (2011). On the dynamics of the furuta pendulum. *Journal of Control Science and Engineering*, 2011:3.

[26] Chang, P.-C. and Fan, C.-Y. (2008). A hybrid system integrating a wavelet and tsk fuzzy rules for stock price forecasting. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(6):802–815.

[27] Chang, P.-C. and Liu, C.-H. (2008). A tsk type fuzzy rule based system for stock price prediction. *Expert Systems with Applications*, 34(1):135 – 144.

[28] Chang, Y. C. and Chen, S. M. (2008). A new method for multiple fuzzy rules interpolation with weighted antecedent variables. In *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*, pages 76–81. IEEE.

[29] Chang, Y.-C., Chen, S.-M., and Liau, C.-J. (2008). Fuzzy interpolative reasoning for sparse fuzzy-rule-based systems based on the areas of fuzzy sets. *IEEE Transactions on Fuzzy Systems*, 16(5):1285–1301.

[30] Chen, C., Mac Parthaláin, N., Li, Y., Price, C., Quek, C., and Shen, Q. (2016a). Rough-fuzzy rule interpolation. *Information Sciences*, 351:1–17.

[31] Chen, C., Parthaláin, N. M., Li, Y., Price, C., Quek, C., and Shen, Q. (2016b). Rough-fuzzy rule interpolation. *Information Sciences*, 351:1 – 17.

[32] Chen, C. C. and Wong, C. C. (2002). Self-generating rule-mapping fuzzy controller design using a genetic algorithm. *Control Theory and Applications*, 149(2):143–148.

[33] Chen, M.-Y. and Linkens, D. (2004). Rule-base self-generation and simplification for data-driven fuzzy models. *Fuzzy Sets and Systems*, 142:243–265.

[34] Chen, M.-Y. and Linkens, D. A. (2001). Rule-base self-generation and simplification for data-driven fuzzy models. In *Fuzzy Systems, 2001. The 10th IEEE International Conference on*, pages 424–427. IEEE.

[35] Chen, S. and Hsin, W. (2015). Weighted fuzzy interpolative reasoning based on the slopes of fuzzy sets and particle swarm optimization techniques. *Cybernetics, IEEE Transactions on*, 45(7):1250–1261.

[36] Chen, S. J. and Chen, S. M. (2003). Fuzzy risk analysis based on similarity measures of generalized fuzzy numbers. *IEEE Transactions on Fuzzy Systems*, 11(1):45–56.

[37] Chen, S. M. and Chang, Y. C. (2011). Weighted fuzzy rule interpolation based on ga-based weight-learning techniques. *IEEE Transactions on Fuzzy Systems*, 19(4):729–744.

[38] Chen, S.-M., Cheng, S.-H., and Chen, Z.-J. (2015). Fuzzy interpolative reasoning based on the ratio of fuzziness of rough-fuzzy sets. *Information Sciences*, 299:394 –411.

[39] Chen, S.-M. and Ko, Y.-K. (2008). Fuzzy interpolative reasoning for sparse fuzzy rule-based systems based on -cuts and transformations techniques. *IEEE Transactions on Fuzzy Systems*, 16(6):1626–1648.

[40] Chiu, S. (1996). Method and software for extracting fuzzy classification rules by subtractive clustering. In *1996 Biennial Conference of the North American*, pages 461–465. IEEE.

[41] Cho, S., Ersoy, O. K., and Lehto, M. (1992). An algorithm to compute the degree of match in fuzzy systems. *Fuzzy Sets and Systems*, 49(3):285–299.

[42] Chua, T. and Tan, W. (2009). A new fuzzy rule-based initialization method for k-nearest neighbor classifier. In *Fuzzy Systems, 2009. FUZZ-IEEE 2009. IEEE International Conference on*, pages 415–420. IEEE.

[43] Collotta, M. and Pau, G. (2015). Bluetooth for internet of things: A fuzzy approach to improve power management in smart homes. *Computers & Electrical Engineering*, 44:137–152.

[44] de Amorim, R. C. and Hennig, C. (2015). Recovering the number of clusters in data sets with noise features using feature rescaling factors. *Information Sciences*, 324:126–145.

[45] De Kleer, J. and Williams, B. C. (1987). Diagnosing multiple faults. *Artificial Intelligence*, 32(1):97–130.

[46] De Matteis, A. D., Marcelloni, F., and Segatori, A. (2015). A new approach to fuzzy random forest generation. In *Fuzzy Systems (FUZZ-IEEE), 2015 IEEE International Conference on*, pages 1–8. IEEE.

[47] Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons.

[48] Diao, R., Chao, F., Peng, T., Snooke, N., and Shen, Q. (2014a). Feature selection inspired classifier ensemble reduction. *IEEE Transactions on Cybernetics*, 44(8):1259–1268.

[49] Diao, R., Jin, S., and Shen, Q. (2014b). Antecedent selection in fuzzy rule interpolation using feature selection techniques. In *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 2206–2213. IEEE.

[50] Diao, R. and Shen, Q. (2012). Feature selection with harmony search. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(6):1509–1523.

[51] Dominik, I. (2011). Fuzzy logic control of rotational inverted pendulum. In *Solid State Phenomena*, volume 177, pages 84–92. Trans Tech Publ.

[52] Dubois, D. and Prade, H. (1992). Gradual inference rules in approximate reasoning. *Information Sciences*, 61(1-2):103–122.

[53] Dubois, D. and Prade, H. (1996). What are fuzzy rules and how to use them. *Fuzzy Sets and Systems*, 84(2):169–185.

[54] Dubois, D. and Prade, H. (1999). On fuzzy interpolation. *International Journal Of General System*, 28(2-3):103–114.

[55] Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57.

[56] Duţu, L., Mauris, G., and Bolon, P. (2018). A fast and accurate rule-base generation method for mamdani fuzzy systems. *IEEE Transactions on Fuzzy Systems*, 26(2):715–733.

[57] Dwivedi, P., Pandey, S., and Junghare, A. (2018). Robust and novel two degree of freedom fractional controller based on two-loop topology for inverted pendulum. *ISA Transactions*, 75:189 – 206.

[58] EIA (2005). Household Energy Consumption and Expenditures by End Use and Energy Source, Selected Years 1978-2005. http://www.eia.doe.gov/emeu/aer/consump.html/. [Online; accessed April-2011].

[59] Elisa, N., Li, J., Zuo, Z., and Yang, L. (2018). Dendritic cell algorithm with fuzzy inference system for input signal generation. In *UK workshop on computational intelligence*, pages 203–214. Springer.

[60] Enyinna, N., Karimoddini, A., Opoku, D., Homaifar, A., and Arnold, S. (2015). Developing an interval type-2 tsk fuzzy logic controller. In *2015 Annual Conference of the North American Fuzzy Information Processing Society (NAFIPS) held jointly with 2015 5th World Conference on Soft Computing (WConSC)*, pages 1–6. IEEE.

[61] Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, volume 96, pages 226–231. AAAI.

[62] Evsukoff, A., Branco, A. C., and Galichet, S. (2002). Structure identification and parameter optimization for non-linear fuzzy modeling. *Fuzzy Sets and Systems*, 132(2):173–188.

[63] Fang, Z., Song, N., and Wang, L. (2009). Design and implementation of a novel fuzzy controller with dsp for rotary inverted pendulum. In *Control and Decision Conference, 2009. CCDC'09. Chinese*, pages 6122–6127. IEEE.

[64] Farag, W., Quintana, V., and Lambert-Torres, G. (1998). A genetic-based neuro-fuzzy approach for modeling and control of dynamical systems. *Neural Networks, IEEE Transactions on*, 9(5):756–767.

[65] Fernández, A. and Herrera, F. (2012). Linguistic fuzzy rules in data mining: follow-up mamdani fuzzy modeling principle. In *Combining Experimentation and Theory*, pages 103–122. Springer.

[66] Friedman, N., Geiger, D., and Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, 29(2-3):131–163.

[67] Furuta, K., Yamakita, M., and Kobayashi, S. (1992). Swing-up control of inverted pendulum using pseudo-state feedback. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 206(4):263–269.

[68] Garba, S., Bajoga, B. G., Mu'azu, M. B., Dajab, D. D., and Abdu-Aguye, U. F. (2011). Development of an anfis-based qos model for a gsm service provider (mtn nigeria kano region network). In *3rd IEEE International Conference on Adaptive Science and Technology (ICAST 2011)*, pages 165–173. IEEE.

[69] García-Escudero, L. A., Gordaliza, A., Matrán, C., Mayo-Iscar, A., et al. (2008). A general trimming approach to robust cluster analysis. *The Annals of Statistics*, 36(3):1324–1345.

[70] GOV.UK (2014). Energy from waste: a guide to the debate. https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/284612/pb14130-energy-waste-201402.pdf/. [Online; accessed April-2019].

[71] Guha, S., Yau, S. S., and Buduru, A. B. (2016). Attack detection in cloud infrastructures using artificial neural network with genetic feature selection. In *2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing*, pages 414–419. IEEE.

[72] Gupta, P. and Kulkarni, N. (2013). An introduction of soft computing approach over hard computing. *International Journal of Latest Trends in Engineering and Technology (IJLTET)*, 3(1):254–258.

[73] Haissig, C. (2000). Adaptive fuzzy temperature control for hydronic heating systems. *Control Systems, IEEE*, 20(2):39–48.

[74] Hamam, A. and Georganas, N. D. (2008). A comparison of mamdani and sugeno fuzzy inference systems for evaluating the quality of experience of hapto-audio-visual applications. In *Haptic Audio visual Environments and Games, 2008. HAVE 2008. IEEE International Workshop on*, pages 87–92. IEEE.

[75] Heydari, G., Gharaveisi, A., and Vali, M. (2016). New formulation for representing higher order tsk fuzzy systems. *IEEE Transactions on Fuzzy Systems*, 24(4):854–864.

[76] Hsiao, W. H., Chen, S. M., and Lee, C. H. (1998). A new interpolative reasoning method in sparse rule-based systems. *Fuzzy Sets and Systems*, 93(1):17–22.

[77] Huang, Z. and Shen, Q. (2004). Scale and move transformation-based fuzzy interpolative reasoning: a revisit. In *Fuzzy Systems, 2004. Proceedings. 2004 IEEE International Conference on*, volume 2, pages 623–628. IEEE.

[78] Huang, Z. and Shen, Q. (2005). Transformation based interpolation with generalized representative values. In *Fuzzy Systems, 2005. FUZZ'05. The 14th IEEE International Conference on*, pages 821–826. IEEE.

[79] Huang, Z. and Shen, Q. (2006). Fuzzy interpolative reasoning via scale and move transformations. *Fuzzy Systems, IEEE Transactions on*, 14(2):340–359.

[80] Huang, Z. and Shen, Q. (2008). Fuzzy interpolation and extrapolation: A practical approach. *Fuzzy Systems, IEEE Transactions on*, 16(1):13–28.

[81] Hwang, K., Cai, M., Chen, Y., and Qin, M. (2007). Hybrid intrusion detection with weighted signature generation over anomalous internet episodes. *IEEE Transactions on Dependable and Secure Computing*, 4(1):41–55.

[82] Jain, A., Tayal, D., and Sehgal, N. (2013). Control of non-linear inverted pendulum using fuzzy logic controller. *International Journal of Computer Applications*, 69(27).

[83] Jin, S., Diao, R., Quek, C., and Shen, Q. (2014). Backward fuzzy rule interpolation. *IEEE Transactions on Fuzzy Systems*, 22(6):1682–1698.

[84] Johanyák, Z. C. and Kovács, S. (2005). Distance based similarity measures of fuzzy sets. *Proceedings of SAMI*, 2005.

[85] Johanyák, Z. C. and Kovács, S. (2006). Fuzzy rule interpolation based on polar cuts. In *Computational Intelligence, Theory and Applications*, pages 499–511. Springer.

[86] Johanyák, Z. C. and Kovács, S. (2007). Sparse fuzzy system generation by rule base extension. In *Proceedings of 11th IEEE International Conference of Intelligent Engineering Systems (INES 2007), Budapest, Hungary*, pages 99–104. IEEE.

[87] Juang, C. F. and Hsu, C. H. (2009). Reinforcement ant optimized fuzzy controller for mobile-robot wall-following control. *IEEE Transactions on Industrial Electronics*, 56(10):3931–3940.

[88] Kang, G., Lee, W., and Sugeno, M. (1998). Design of tsk fuzzy controller based on tsk fuzzy model using pole placement. In *Fuzzy Systems Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, pages 246–251. IEEE.

[89] Kempton, W. (1986). Two theories of home heat control*. *Cognitive Science*, 10(1):75–90.

[90] Kerk, Y. W., Pang, L. M., Tay, K. M., and Lim, C. P. (2016). Multi-expert decision-making with incomplete and noisy fuzzy rules and the monotone test. In *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 94–101. IEEE.

[91] Kleer, J. D. (1986a). An assumption-based tms. *Artificial Intelligence*, 28(2):127–162.

[92] Kleer, J. D. (1986b). Extending the atms. *Artificial Intelligence*, 28(2):163–196.

[93] Klir, G. J. and Yuan, B. (1996). Fuzzy sets and fuzzy logic: theory and applications. *Possibility Theory versus Probab. Theory*, 32(2):207–208.

[94] Kóczy, L. and Hirota, K. (1993a). Approximate reasoning by linear rule interpolation and general approximation. *International Journal of Approximate Reasoning*, 9(3):197–225.

[95] Kóczy, L. and Hirota, K. (1993b). Interpolative reasoning with insufficient evidence in sparse fuzzy rule bases. *Information Sciences*, 71(1-2):169–201.

[96] Kóczy, L. and Hirota, K. (1993c). Ordering, distance and closeness of fuzzy sets. *Fuzzy Sets and Systems*, 59(3):281–293.

[97] Kóczy, L. T. and Hirota, K. (1997). Size reduction by interpolation in fuzzy rule bases. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 27(1):14–25.

[98] Kodinariya, T. M. and Makwana, P. R. (2013). Review on determining number of cluster in k-means clustering. *International Journal*, 1(6):90–95.

[99] Kouikoglou, V. S. and Phillis, Y. A. (2009). On the monotonicity of hierarchical sum–product fuzzy systems. *Fuzzy Sets and Systems*, 160(24):3530–3538.

[100] Kovács, S. (2006). Extending the fuzzy rule interpolation" five" by fuzzy observation. In *Computational Intelligence, Theory and Applications*, pages 485–497. Springer.

[101] Kovacs, S. and Koczy, L. T. (1997). Approximate fuzzy reasoning based on interpolation in the vague environment of the fuzzy rulebase. In *Intelligent Engineering Systems, 1997. INES'97. Proceedings., 1997 IEEE International Conference on*, pages 63–68. IEEE.

[102] Kovács, S. and Kóczy, L. T. (1997). The use of the concept of vague environment in approximate fuzzy reasoning. *Fuzzy Set Theory and Applications*, 12:169–181.

[103] Kriegel, H.-P., Kröger, P., Sander, J., and Zimek, A. (2011). Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3):231–240.

[104] Krishen, J. and Becerra, V. M. (2006). Efficient fuzzy control of a rotary inverted pendulum based on lqr mapping. In *Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE*, pages 2701–2706. IEEE.

[105] Lange, T., Roth, V., Braun, M. L., and Buhmann, J. M. (2004). Stability-based validation of clustering solutions. *Neural Computation*, 16(6):1299–1323.

[106] Larsen, P. M. (1980). Industrial applications of fuzzy logic control. *International Journal of Man-Machine Studies*, 12(1):3–10.

[107] Lawry, J. (2008). Appropriateness measures: an uncertainty model for vague concepts. *Synthese*, 161(2):255–269.

[108] Lawry, J. and Tang, Y. (2009). Uncertainty modelling for vague concepts: A prototype theory approach. *Artificial Intelligence*, 173(18):1539–1558.

[109] Lee, L.-W. and Chen, S.-M. (2007). A new fuzzy interpolative reasoning method for sparse fuzzy rule-based systems. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 745–755. Springer.

[110] Li, J., Qu, Y., Chao, F., Shum, H. P., Ho, E. S., and Yang, L. (2019). Machine learning algorithms for network intrusion detection. In *AI in Cybersecurity*, pages 151–179. Springer.

[111] Li, J., Qu, Y., Shum, H. P., and Yang, L. (2017a). Tsk inference with sparse rule bases. In *Advances in Computational Intelligence Systems*, pages 107–123. Springer.

[112] Li, J., Shum, H. P. H., Fu, X., Sexton, G., and Yang, L. (2016). Experience-based rule base generation and adaptation for fuzzy interpolation. In *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 102–109. IEEE.

[113] Li, J., Yang, L., Fu, X., Chao, F., and Qu, Y. (2017b). Dynamic qos solution for enterprise networks using tsk fuzzy interpolation. In *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–6. IEEE.

[114] Li, J., Yang, L., Fu, X., Chao, F., and Qu, Y. (2018a). Interval type-2 tsk+ fuzzy inference system. In *IEEE World Congress on Computation Intelligence Internation Conference*. IEEE.

[115] Li, J., Yang, L., Qu, Y., and Sexton, G. (2018b). An extended takagi–sugeno–kang inference system (tsk+) with fuzzy interpolation and its rule base generation. *Soft Computing*, 22(10):3155–3170.

[116] Li, J., Yang, L., Shum, H. P. H., Sexton, G., and Tan, Y. (2015a). Intelligent home heating controller using fuzzy rule interpolation. In *UK Workshop on Computational Intelligence*, pages 1–8. IEEE.

[117] Li, W., Qiao, J., and Zeng, X.-J. (2015b). Accurate similarity analysis and computing of gaussian membership functions for fnn simplification. In *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pages 402–409. IEEE.

[118] Li, Z., Liu, J., Chen, S., and Tang, X. (2007). Noise robust spectral clustering. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE.

[119] Lleti, R., Ortiz, M., Sarabia, L., and Sanchez, M. (2004). Selecting variables for k-means cluster analysis by using a genetic algorithm that optimises the silhouettes. *Analytica Chimica Acta*, 515(1):87 – 100.

[120] Lu, J., Sookoor, T., Srinivasan, V., Gao, G., Holben, B., Stankovic, J., Field, E., and Whitehouse, K. (2010). The smart thermostat: Using occupancy sensors to save energy in homes. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 211–224. ACM.

[121] L.Yang and Shen, Q. (2010). Adaptive fuzzy interpolation and extrapolation with multiple-antecedent rules. In *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, pages 1–8. IEEE.

[122] L.Yang and Shen, Q. (2011). Adaptive fuzzy interpolation. *Fuzzy Systems, IEEE Transactions on*, 19(6):1107–1126.

[123] L.Yang and Shen, Q. (2013). Closed form fuzzy interpolation. *Fuzzy Sets and Systems*, 225:1–22.

[124] Ma, X.-J., Sun, Z.-Q., and He, Y.-Y. (1998). Analysis and design of fuzzy controller and fuzzy observer. *IEEE Transactions on Fuzzy Systems*, 6(1):41–51.

[125] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and PSrobability*, pages 281–297. Oakland.

[126] Mamat, M. and Ghani, N. M. (2009). Fuzzy logic controller on automated car braking system. In *2009 IEEE International Conference on Control and Automation*, pages 2371–2375. IEEE.

[127] Mamdani, E. H. (1977). Application of fuzzy logic to approximate reasoning using linguistic synthesis. *Computers, IEEE Transactions on*, C-26(12):1182–1191.

[128] Mamdani, E. H. and Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-machine Studies*, 7(1):1–13.

[129] Mannle, M. et al. (1999). Identifying rule-based tsk fuzzy models. *Institute for Computer Design and Fault Tolerance, University of Karlsruhe, Germany*.

[130] Mendel, J., John, R., and Liu, F. (2006). Interval type-2 fuzzy logic systems made simple. *IEEE Transactions on Fuzzy Systems*, 14(6):808–821.

[131] Mendel, J. M. and Mouzouris, G. C. (1997). Designing fuzzy logic systems. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 44(11):885–895.

[132] Mendi, F., Boran, K., and Kulekci, M. K. (2002). Fuzzy controlled central heating system. *International Journal of Energy Research*, 26(15):1313–1322.

[133] Mucientes, M., Vidal, J. C., Bugarín, A., and Lama, M. (2009). Processing time estimations by variable structure tsk rules learned through genetic programming. *Soft Computing*, 13(5):497–509.

[134] Naik, N., Diao, R., and Shen, Q. (2014). Genetic algorithm-aided dynamic fuzzy rule interpolation. In *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*, pages 2198–2205. IEEE.

[135] Naik, N., Diao, R., and Shen, Q. (2016). Application of dynamic fuzzy rule interpolation for intrusion detection: D-fri-snort. In *2016 IEEE International Conference on Fuzzy Systems*, pages 78–85. IEEE.

[136] Naik, N., Diao, R., and Shen, Q. (2018). Dynamic fuzzy rule interpolation and its application to intrusion detection. *IEEE Transactions on Fuzzy Systems*, 26(4):1878–1892.

[137] Najeeb, R. F. and Dhannoon, B. N. (2017). Classification for intrusion detection with different feature selection methods: A survey (2014-2016). *International Journal*, 7(5):305–311.

[138] Nandi, A. K. and Klawonn, F. (2007). Detecting ambiguities in regression problems using tsk models. *Soft Computing*, 11(5):467–478.

[139] Narayanan, S. J., Paramasivam, I., Bhatt, R. B., and Khalid, M. (2015). A study on the approximation of clustered data to parameterized family of fuzzy membership functions for the induction of fuzzy decision trees. *Cybernetics and Information Technologies*, 15(2):75–96.

[140] Nauck, D. and Kruse, R. (1999). *Neuro-fuzzy methods in fuzzy rule generation*, pages 305–334. Springer US.

[141] Neapolitan, R. E. (2012). *Probabilistic reasoning in expert systems: theory and algorithms*. CreateSpace Independent Publishing Platform.

[142] Negnevitsky, M. (2005). *Artificial intelligence: a guide to intelligent systems*. Pearson Education.

[143] Nozaki, K., Ishibuchi, H., and Tanaka, H. (1997a). A simple but powerful heuristic method for generating fuzzy rules from numerical data. *Fuzzy Sets and Systems*, 86(3):251–270.

[144] Nozaki, K., Ishibuchi, H., and Tanaka, H. (1997b). A simple but powerful heuristic method for generating fuzzy rules from numerical data. *Fuzzy Sets and Systems*, 86(3):251–270.

[145] Núñez, H., Angulo, C., and Català, A. (2006). Rule-based learning systems for support vector machines. *Neural Processing Letters*, 24(1):1–18.

[146] Özbek, N. S. and Efe, M. Ö. (2010). Swing up and stabilization control experiments for a rotary inverted pendulum—an educational comparison. In *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, pages 2226–2231. IEEE.

[147] Pal, T. and Pal, N. R. (2003). Sogarg: A self-organized genetic algorithm-based rule generation scheme for fuzzy controllers. *IEEE Transactions on Evolutionary Computation*, 7(4):397–415.

[148] Patel, A.and Taghavi, M., Bakhtiyari, K., and JúNior, J. C. (2013). An intrusion detection and prevention system in cloud computing: A systematic review. *Journal of Network and Computer Applications*, 36(1):25–41.

[149] Paul, A. K., Shill, P. C., Rabin, M. R. I., and Murase, K. (2017). Adaptive weighted fuzzy rule-based system for the risk level assessment of heart disease. *Applied Intelligence*, pages 1–18.

[150] Pearl, J. (2014). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier.

[151] Peckham, S. D. (2011). Profile, plan and streamline curvature: A simple derivation and applications. *Proceedings of Geomorphometry*, pages 27–30.

[152] Pelleg, D., Moore, A. W., et al. (2000). X-means: Extending k-means with efficient estimation of the number of clusters. In *International Conference on Machine Learning (ICML)*, volume 1, pages 727–734. ICML.

[153] Pigg, S. (2000). Programmable thermostats that go berserk? taking a social perspective on space heating in wisconsin. *Proc. ACEEE Summer Study on Energy Efficiency in Buildings*, 8:233–244.

[154] Plourde, A. (2003). Programmable thermostats as means of generating energy savings: some pros and cons. *Technical Report CBEEDAC 2003-RP*, 11(2):230–232.

[155] Priyono, A., Ridwan, M., Alias, A. J., Atiq, R., Rahmat, O., Hassan, A., and Ali, M. (2005). Generation of fuzzy rules with subtractive clustering. *Jurnal Teknologi*, 43(D):143–153.

[156] Rehman, S. U., Asghar, S., Fong, S., and Sarasvady, S. (2014). Dbscan: Past, present and future. In *The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014)*, pages 232–238. SOCIO.

[157] Ren, Q., Balazinski, M., and Baron, L. (2012). High-order interval type-2 takagi-sugeno-kang fuzzy logic system and its application in acoustic emission signal modeling in turning process. *The International Journal of Advanced Manufacturing Technology*, 63(9):1057–1063.

[158] Rezaee, B. and Zarandi, M. F. (2010). Data-driven fuzzy modeling for takagi–sugeno–kang fuzzy system. *Information Sciences*, 180(2):241–255.

[159] Robinson, J. A. (1965). A machine-oriented logic based on the resolution principle. *Journal of the ACM (JACM)*, 12(1):23–41.

[160] Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

[161] Salgado, P. (2008). Rule generation for hierarchical collaborative fuzzy system. *Applied Mathematical Modelling*, 32(7):1159 – 1178.

[162] Samak, T. and Al-Shaer, E. (2012). Fuzzy conflict analysis for qos policy parameters in diffserv networks. *IEEE Transactions on Network and Service Management*, 9(4):459–472.

[163] Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):210–229.

[164] Samuel, A. L. (1988). *Some Studies in Machine Learning Using the Game of Checkers. I*, pages 335–365. Springer New York.

[165] Sander, J., Ester, M., Kriegel, H.-P., and Xu, X. (1998). Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data Mining and Knowledge Discovery*, 2(2):169–194.

[166] Saunders, J., McClure, C., and Mandel, L. (2012). Broadband applications: Categories, requirements, and future frameworks. *First Monday*, 17(11).

[167] Scott, J., Bernheim Brush, A., Krumm, J., Meyers, B., Hazas, M., Hodges, S., and Villar, N. (2011). Preheat: Controlling home heating using occupancy prediction. In *Proceedings of the 13th International Conference on Ubiquitous Computing*, UbiComp '11, pages 281–290. ACM.

[168] Shanmugavadivu, R. and Nagarajan, N. (2011). Network intrusion detection system using fuzzy logic. *Indian Journal of Computer Science and Engineering (IJCSE)*, 2(1):101–111.

[169] Shen, Q. and Yang, L. (2011). Generalisation of scale and move transformation-based fuzzy interpolation. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 15(3):288–298.

[170] Singhala, P., Shah, D., and Patel, B. (2014). Temperature control using fuzzy logic. *International Journal of Instrumentation and Control Systems (IJICS)*, 4(1):1–10.

[171] Sitompul, E. and Bukhori, I. (2014). A new approach in self-generation of fuzzy logic controller by means of genetic algorithm. In *Information Technology and Electrical Engineering (ICITEE), 2014 6th International Conference on*, pages 1–6. IEEE.

[172] Smyth, P. (1996). Clustering using monte carlo cross-validation. In *KDD*, volume 1, pages 26–133. ACM New York.

[173] Sonbol, A. and Fadali, M. S. (2002). A new approach for designing tsk fuzzy systems from input-output data. In *American Control Conference, 2002. Proceedings of the 2002*, volume 2, pages 989–994. IEEE.

[174] Song, J. (2016). Feature selection for intrusion detection system. *PhD Thesis, Aberystwyth University*, pages 43–54.

[175] Streit, R. L. and Luginbuhl, T. E. (1994). Maximum likelihood training of probabilistic neural networks. *IEEE Transactions on Neural Networks*, 5(5):764–783.

[176] Szigeti, T., Hattingh, C., Barton, R., and Briley Jr, K. (2013). *End-to-End QoS Network Design: Quality of Service for Rich-Media & Cloud Networks*. Cisco Press.

[177] Takagi, T. and Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control. *Systems, Man and Cybernetics, IEEE Transactions on*, SMC-15(1):116–132.

[178] Tan, Y., Li, J., Wonders, M., Chao, F., Shum, H. P. H., and Yang, L. (2016). Towards sparse rule base generation for fuzzy rule interpolation. In *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 110–117. IEEE.

[179] Tang, Y. and Chen, S. (2007). An automated signature-based approach against polymorphic internet worms. *IEEE Transactions on Parallel and Distributed Systems*, 18(7):879–892.

[180] Tavallaee, M., Bagheri, E., Lu, W., and Ghorbani, A. A. (2009). A detailed analysis of the kdd cup 99 data set. In *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pages 1–6. IEEE.

[181] Tavallaee, M., Stakhanova, N., and Ghorbani, A. A. (2010). Toward credible evaluation of anomaly-based intrusion-detection methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(5):516–524.

[182] Thorndike, R. L. (1953). Who belongs in the family? *Psychometrika*, 18(4):267–276.

[183] Tikk, D. and Baranyi, P. (2000). Comprehensive analysis of a new fuzzy rule interpolation method. *IEEE Transactions on Fuzzy Systems*, 8(3):281–296.

[184] Tikk, D., Joó, I., Kóczy, L., Várlaki, P., Moser, B., and Gedeon, T. D. (2002). Stability of interpolative fuzzy kh controllers. *Fuzzy Sets and Systems*, 125(1):105–119.

[185] Uslan, V. and Seker, H. (2016). Quantitative prediction of peptide binding affinity by using hybrid fuzzy support vector regression. *Applied Soft Computing*, 43:210 – 221.

[186] Vainio, A.-M., Valtonen, M., and Vanhala, J. (2006). Learning and adaptive fuzzy control system for smart home. In *Developing Ambient Intelligence*, pages 28–47. Springer.

[187] Von Altrock, C. (1995). *Fuzzy logic and NeuroFuzzy applications explained*. Englewood Cliffs, N.J. : Prentice Hall PTR.

[188] Von Altrock, C., Arend, H.-O., Krausse, B., and Steffens, C. (1994a). Customer-adaptive fuzzy logic control of home heating system. In *Fuzzy Systems, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the Third IEEE Conference on*, pages 1713–1718. IEEE.

[189] Von Altrock, C., Krausse, B., and Steffens, C. (1994b). Customer-adaptive fuzzy logic control of home heating system. In *Fuzzy Systems, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the Third IEEE Conference on*, pages 1713–1718 vol.3. IEEE.

[190] Wang, G., Hao, J., Ma, J., and Huang, L. (2010). A new approach to intrusion detection using artificial neural networks and fuzzy clustering. *Expert Systems with Applications*, 37(9):6225–6232.

[191] Wang, L.-X. and Mendel, J. M. (1992a). Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. *IEEE transactions on Neural Networks*, 3(5):807–814.

[192] Wang, L.-X. and Mendel, J. M. (1992b). Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(6):1414–1427.

[193] Wang, Z. (2001). *Internet QoS: architectures and mechanisms for quality of service*. Morgan Kaufmann.

[194] Witten, D. M. and Tibshirani, R. (2010). A framework for feature selection in clustering. *Journal of the American Statistical Association*, 105(490):713–726.

[195] Wu, D. (2010). A brief tutorial on interval type-2 fuzzy sets and systems. *Fuzzy Sets and Systems*.

[196] Wu, D. and Nie, M. (2011). Comparison and practical implementation of type-reduction algorithms for type-2 fuzzy sets and systems. In *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, pages 2131–2138. IEEE.

[197] Wu, Z. Q., Masaharu, M., and Shi, Y. (1996). An improvement to kóczy and hirota's interpolative reasoning in sparse fuzzy rule bases. *International Journal of Approximate Reasoning*, 15(3):185–201.

[198] Xenaki, S. D., Koutroumbas, K. D., and Rontogiannis, A. A. (2016a). A novel adaptive possibilistic clustering algorithm. *IEEE Transactions on Fuzzy Systems*, 24(4):791–810.

[199] Xenaki, S. D., Koutroumbas, K. D., and Rontogiannis, A. A. (2016b). Sparsity-aware possibilistic clustering algorithms. *IEEE Transactions on Fuzzy Systems*, 24(6):1611–1626.

[200] Xu, L. (2002). Byy harmony learning, structural rpcl, and topological self-organizing on mixture models. *Neural Networks*, 15(8-9):1125–1151.

[201] Yager, R. R. and Filev, D. P. (1994). Generation of fuzzy rules by mountain clustering. *Journal of Intelligent and Fuzzy Systems*, 2(3):209–219.

[202] Yam, Y. and Kóczy, L. T. (2000). Representing membership functions as points in high-dimensional spaces for fuzzy interpolation and extrapolation. *IEEE Transactions on Fuzzy Systems*, 8(6):761–772.

[203] Yan, S., Mizumoto, M., and Qiao, W. Z. (1995). Reasoning conditions on kóczy's interpolative reasoning method in sparse fuzzy rule bases. *Fuzzy Sets and Systems*, 75(1):63 – 71.

[204] Yang, L., Chao, F., and Shen, Q. (2017a). Generalised adaptive fuzzy rule interpolation. *IEEE Transactions on Fuzzy Systems*, 25(4):839–853.

[205] Yang, L., Chen, C., Jin, N., Fu, X., and Shen, Q. (2014). Closed form fuzzy interpolation with interval type-2 fuzzy sets. In *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*, pages 2184–2191. IEEE.

[206] Yang, L., Li, J., Chao, F., Hackney, P., and Flanagan, M. (2018). Job shop planning and scheduling for manufacturers with manual operations. *Expert Systems*, page e12315.

[207] Yang, L., Li, J., Fehringer, G., Barraclough, P., Sexton, G., and Cao, Y. (2017b). Intrusion detection system by fuzzy interpolation. In *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–6. IEEE.

[208] Yang, L., Li, J., Hackney, P., Chao, F., and Flanagan, M. (2017c). Manual task completion time estimation for job shop scheduling using a fuzzy inference system. In *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 139–146. IEEE.

[209] Yang, L. and Shen, Q. (2009a). Towards adaptive interpolative reasoning. In *Fuzzy Systems, 2009. FUZZ-IEEE 2009. IEEE International Conference on*, pages 542–549. IEEE.

[210] Yang, L. and Shen, Q. (2009b). Towards adaptive interpolative reasoning. In *Fuzzy Systems, 2009. FUZZ-IEEE 2009. IEEE International Conference on*, pages 542–549. IEEE.

[211] Yang, L. and Shen, Q. (2011a). Adaptive fuzzy interpolation with prioritized component candidates. In *Fuzzy Systems (FUZZ), 2011 IEEE International Conference on*, pages 428–435. IEEE.

[212] Yang, L. and Shen, Q. (2011b). Adaptive fuzzy interpolation with uncertain observations and rule base. In *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, pages 471–478. IEEE.

[213] Yang, L., Shen, Q., et al. (2009). Extending adaptive interpolation: From triangular to trapezoidal. In *2009 9th UK Workshop on Computational Intelligence (UKCI)*, pages 25–30. Springer.

[214] Yang, L., Zuo, Z., Chao, F., and Qu, Y. (2017d). Fuzzy interpolation systems and applications. In *Modern Fuzzy Control Systems and Its Applications*, pages 49–70. InTech.

[215] Yasunobu, S. and Miyamoto, S. (1985). Automatic train operation system by predictive fuzzy control. *Industrial applications of fuzzy control*, 1(18):1–18.

[216] Yoo, C. K., Vanrolleghem, P. A., and Lee, I.-B. (2003). Nonlinear modeling and adaptive monitoring with fuzzy and multivariate statistical methods in biological wastewater treatment plants. *Journal of Biotechnology*, 105(1-2):135–163.

[217] Yuan, Y. and Zhuang, H. (1996). A genetic algorithm for generating fuzzy classification rules. *Fuzzy Sets and Systems*, 84(1):1–19.

[218] Zadeh, L. (1965). Fuzzy sets. *Information and Control*, 8(3):338 – 353.

[219] Zadeh, L. A. (1971). Quantitative fuzzy semantics. *Information Sciences*, 3(2):159–176.

[220] Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(1):28–44.

[221] Zadeh, L. A. (1994). Fuzzy logic, neural networks, and soft computing. *Communications of the ACM*, 37(3):77–85.

[222] Zhang, R. and Lu, Z. (2016). Large scale sparse clustering. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 2336–2342. AAAI Press.

[223] Zhang, R. and Ma, J. (2000). Fuzzy qos management in diff-serv networks. In *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*, volume 5, pages 3752–3757. IEEE.

[224] Zhenya, H., Chengjian, W., Luxi, Y., Xiqi, G., Susu, Y., Eberhart, R. C., and Shi, Y. (1998). Extracting rules from fuzzy neural network by particle swarm optimisation. In *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, pages 74–77. IEEE.

[225] Zhou, D., Shi, M., Chao, F., Lin, C.-M., Yang, L., Shang, C., and Zhou, C. (2018). Use of human gestures for controlling a mobile robot via adaptive cmac network and fuzzy logic controller. *Neurocomputing*, 282:218 – 231.

[226] Zuo, Z., Li, J., Anderson, P., Yang, L., and Naik, N. (2018). Grooming detection using fuzzy-rough feature selection and text classification. In *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8. IEEE.

[227] Zuo, Z., Li, J., Wei, B., Yang, L., Chao, F., and Naik, N. (2019). Adaptive activation function generation for artificial neural networks through fuzzy inference with application in grooming text categorisation. In *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–6. IEEE.