CHAPTER 1

Introduction

1.1 EARLY HISTORY OF FUZZY CONTROL

Fuzzy control (also known as fuzzy logic control) is regarded as the most widely used application of fuzzy logic and is credited with being a well-accepted methodology for designing controllers that are able to deliver satisfactory performance in the face of uncertainty and imprecision (Lee, 1990; Sugeno, 1985; Feng, 2006). In addition, fuzzy logic theory provides a method for less skilled personnel to develop practical control algorithms in a user-friendly way that is close to human thinking and perception, and to do this in a short amount of time. Fuzzy logic controllers (FLCs) can sometimes outperform traditional control systems [like proportional–integral–derivative (PID) controllers] and have often performed either similarly or even better than human operators. This is partially because most FLCs are nonlinear controllers that are capable of controlling real-world systems (the vast majority of such systems are nonlinear) better than a linear controller can, and with minimal to no knowledge about the mathematical model of the plant or process being controlled.

Fuzzy logic controllers have been applied with great success to many real-world applications. The first FLC was developed by Mamdani and Assilian (1975), in the United Kingdom, for controlling a steam generator in a laboratory setting. In 1976, Blue Circle Cement and SIRA in Denmark developed a cement kiln controller (the first industrial application of fuzzy logic), which went into operation in 1982 (Holmblad and Ostergaard, 1982). In the 1980s, several important industrial applications of fuzzy logic control were launched successfully in Japan, including a water treatment system developed by Fuji Electric. In 1987, Hitachi put a fuzzy logic based automatic train operation control system into the Sendai city's subway system (Yasunobu and Miyamoto, 1985). These and other applications of FLCs motivated many Japanese engineers to investigate a wide range of novel applications for fuzzy logic. This led to a "fuzzy boom" in Japan, a result of close collaboration and technology transfer between universities and industry.

According to Yen and Langari (1999), in 1988, a large-scale national research initiative was established by the Japanese Ministry of International Trade and

Introduction to Type-2 Fuzzy Logic Control: Theory and Applications, First Edition.
Jerry M. Mendel, Hani Hagras, Woei-Wan Tan, William W. Melek, and Hao Ying.

Industry (MITI). The initiative established by MITI was a consortium called the Laboratory for International Fuzzy Engineering Research (LIFE). In late January 1990, Matsushita Electric Industrial (Panasonic) named their newly developed fuzzy-controlled automatic washing machine the fuzzy washing machine and launched a major commercial campaign of it as a *fuzzy* product. This campaign turned out to be a successful marketing effort not only for the product but also for fuzzy logic technology (Yen and Langari, 1999). Many other home electronics companies followed Panasonic's approach and introduced fuzzy vacuum cleaners, fuzzy rice cookers, fuzzy refrigerators, fuzzy camcorders (for stabilizing the image under hand jittering), fuzzy camera (for smart autofocus), and other applications. As a result, consumers in Japan recognized the now en-vogue Japanese word "fuzzy," which won the gold prize for a new word in 1990 (Hirota, 1995). Originating in Japan, the "fuzzy boom" triggered a broad and serious interest in this technology in Korea, Europe, the United States, and elsewhere. For example, Boeing, NASA, United Technologies, and other aerospace companies developed FLCs for space and aviation applications (Munakata and Jani, 1994).

Today FLCs are used in countless real-world applications that touch the lives of people all over the world, including white goods (e.g., washing machines, refrigerators, microwaves, rice cookers, televisions, etc.), digital video cameras, cars, elevators (lifts), heavy industries (e.g., cement, petroleum, steel), and the like.

While this book focuses on type-2 fuzzy logic control, it will also provide background material about type-1 fuzzy logic control. Indeed, before we can explain what type-2 fuzzy logic control is we must briefly explain what type-1 fuzzy sets, type-1 fuzzy logic control, and type-2 fuzzy sets are. In this chapter we do this from a high-level perspective without touching on the mathematical aspects in order to give a feel for the nature of fuzzy sets and their applications. Later chapters in this book provide rigorous treatments of mathematical underpinnings of the subjects just mentioned.

1.2 WHAT IS A TYPE-1 FUZZY SET?

Suppose that a group of people is asked about the temperature values they associate with the linguistic concepts Hot and Cold. If *crisp sets* are employed, as shown in Fig. 1.1a, then a threshold must be chosen above which temperature values are considered Hot and below which they are considered Cold. Reaching a consensus about such a threshold is difficult, and even if an agreement can be reached—for example, 18°C—, is it reasonable to conclude that 17.99999°C is Cold whereas 18.00001°C is Hot?

On the other hand, Hot and Cold can be represented as *type-1 fuzzy sets* (T1 FSs) whose membership functions (MFs) are shown in Fig. 1.1b. Note that, prior to the appearance of type-2 fuzzy sets, the phrase *fuzzy set* was used instead of the phrase *T1 fuzzy set*. Even today, in many publications that focus only on T1 FSs, such sets are called fuzzy sets. In this book we shall use the phrase *type-1 fuzzy set*. Returning to Fig. 1.1b, observe that no sharp boundaries exist between the two sets

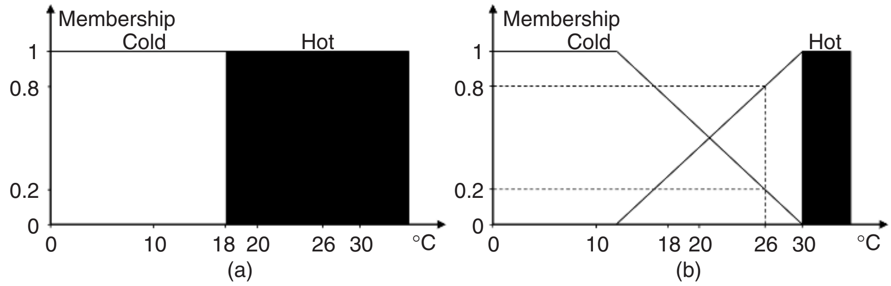


Figure 1.1 Representing Cold and Hot using (a) crisp sets, and (b) type-1 fuzzy sets.

and that each value on the horizontal axis may simultaneously belong to more than one T1 FS but with different degrees of membership. For example, 26°C, which is in the crisp Hot set with a membership value of 1.0 (Fig. 1.1a), is now in that set to degree 0.8, but is also in the Cold set to degree 0.2 (Fig. 1.1b).

Type-1 FSs provide a means for calculating intermediate values between the crisp values associated with being absolutely true (1) or absolutely false (0). Those values range between 0 and 1 (and can include them); thus, it can be said that a fuzzy set allows the calculation of shades of gray between white and black (or true and false). As will be seen in this book, the smooth transition that occurs between T1 FSs gives a good decision response for a type-1 fuzzy logic control system in the face of noise and other uncertainties.

1.3 WHAT IS A TYPE-1 FUZZY LOGIC CONTROLLER?

With the advent of type-2 fuzzy sets and type-2 fuzzy logic control, it has become necessary to distinguish between *type-2 fuzzy logic control* and all earlier fuzzy logic control that uses type-1 fuzzy sets (the distinctions between such fuzzy sets are explained in Section 1.4). We refer to fuzzy logic control that uses type-1 fuzzy sets as *type-1 fuzzy logic control*. When it does not matter whether the fuzzy sets are type-1 or type-2, we just use *fuzzy logic control* or *fuzzy control*.

Fuzzy logic control aims to mimic the process followed by the human mind when performing control actions. For example, when a person drives (controls) a car, he/she will not think:

> If the temperature is *10 degrees Celsius* and the rainfall is *70.5 mm* and the road is *40% slippery* and the distance between my car and the car in front of me is *3 meters*, then I will depress the acceleration pedal only *10%*.

Instead, it is much more likely that he/she thinks:

> If it is Cold and the rainfall is High and the road is Somewhat Slippery and the distance between my car and the car in front of me is Quite Close, then I will depress the acceleration pedal Slightly.

So, in systems controlled by humans, the control cycle starts by a person converting a physical quantity (e.g., a distance) from numbers into words or perceptions (e.g., Quite Close distance). The input words (or perceptions) then trigger a person's knowledge, accumulated through that person's experience, resulting in words representing actions (e.g., depress the acceleration pedal Slightly). The person then executes an action to actuate a given device that interfaces the person with the controlled system (e.g., depress the acceleration pedal only 10% might represent the person's implementation of "depress the accelerator pedal Slightly"). Because people think and reason by using imprecise linguistic information, FLCs try to mimic and convert linguistic control information into numerical control information that can be used in automatic control systems.

In its attempt to mimic human control actions, a type-1 FLC, whose structure is shown in Fig. 1.2, is composed of four main components: fuzzifier, rules, inference engine, and defuzzifier, where the operation of each component is summarized as follows:

- The fuzzifier maps each measured numerical input variable into a fuzzy set. One motivation for doing this is that measurements may be corrupted by noise and are somewhat uncertain (even after filtering). So, for example, a measured temperature of 26°C may be modeled as a triangular type-1 fuzzy set that is symmetrically centered around 26°C, where the base of the triangle is related to the uncertainty of this measurement. If, however, one believes that there is no measurement uncertainty, then the measurements can be modeled as crisp sets.
- Rules have an if–then structure, for example, *If Temperature is Low and Pressure is High*, *then Fan Speed is Low*. Each IF part of a rule is called its *antecedent*, and the THEN part of a rule is called its *consequent*. Rules relate input fuzzy sets to output fuzzy sets. All of the rules are collected into a rule base.

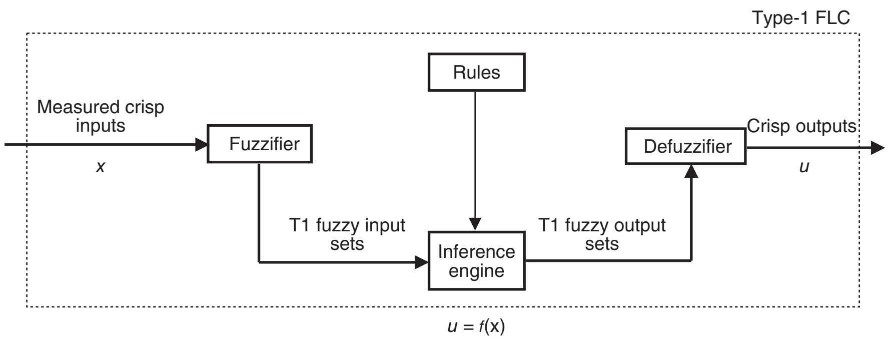


Figure 1.2 General structure of a type-1 FLC. The heavy lines with arrows indicate the path taken by signals during the actual operation of the FLC. Rules are used during the design of the FLC and are activated by the inference engine during the actual operation of the FLC (Mendel et al. (2006); © 2006, IEEE).

- The inference engine decides which rules from the rule base are fired and what their degrees of firing are, by using the fuzzy sets provided to it from the fuzzifier as well as some mathematics about fuzzy sets. The inference engine may also combine each rule's degree of firing with that rule's consequent fuzzy set to produce the rule's *output fuzzy set* (i.e., its fired-rule output set), and then combine all of those sets (across all of the fired rules) to produce an aggregated fuzzy output set using the mathematics of fuzzy sets; or it may send each rule's degree of firing directly to the defuzzifier where they are all aggregated in a different way.
- The defuzzifier receives either the aggregated fuzzy output sets from the inference engine or the degrees of firing for each rule plus some information about each consequent fuzzy set, and then processes this data to produce crisp outputs that are then passed to the physical actuators that control the actual plant.

In general, real-world control systems, such as fuzzy logic control systems, are affected by the following uncertainties:

- Uncertainties about the inputs to the FLC. For instance, sensor measurements can be affected by high noise levels and changing observation conditions such as changing environmental conditions, for example, wind, rain, humidity, and so forth. In addition to measurement noise, other possible inputs to the FLC, such as those estimated by an observer or computed using a process model, can also be imprecise and exhibit uncertainty.
- Uncertainties about control outputs that can occur because of changes in an actuator's characteristics due to wear and tear, environmental changes, and the like.
- Uncertainties about the change in operating conditions of the controller, such as changes in a plant's parameters.
- Uncertainties due to disturbances acting upon the system when those disturbances cannot be measured, for example, wind buffeting an airplane.

In a T1 FLC all of these uncertainties are handled by the T1 FSs in the antecedents and consequents of the rules, as well as through the chosen type of fuzzifier. Regarding the latter, one may choose to use: (1) a singleton fuzzifier in which a measured value is treated as perfect and is modeled as a crisp set; or (2) a type-1 fuzzifier in which a measured value is treated as signal plus stationary noise and is modeled as a normal, convex T1 FS (also called a *T1 fuzzy number*).

The type-1 FLC in Fig. 1.2 is a nonlinear controller that maps its inputs $\mathbf{x}$ into an output u, that is, $u = f(\mathbf{x})$, where f is a nonlinear function that is formed by fuzzy logic operations and the mathematics of fuzzy sets. Often, $f(\mathbf{x})$ is formed from linguistic rules that summarize human knowledge or experience (or may be constructed from data); thus, the type-1 FLC directly maps such knowledge or experience into a nonlinear control law whose explicit mathematical expression is unknown in most cases.

Many researchers (e.g., Wang, 1992; Wang and Mendel, 1992a; Castro, 1995; Kosko, 1994; Kreinovich et al. 1998) have shown that the type-1 FLC $f(\mathbf{x})$ can uniformly approximate any real continuous function on a compact domain to any degree of accuracy; hence, FLCs are known to be *universal approximators*. One way to interpret what this means is that the FLC $f(\mathbf{x})$ approximates a function by covering its graph with *fuzzy patches* (Kosko, 1994), where each rule in the FLC defines a fuzzy patch in system's input–output space, and it then averages overlapping patches. This approximation improves as the fuzzy patches grow in number and shrink in size; however, as more smaller patches are included, the complexity of the model increases (i.e., the number of fuzzy sets and rules increases).

Type-1 FLCs produce nonlinear control laws $f(\mathbf{x})$ that cannot be effectively generated by any other mathematical means because such $f(\mathbf{x})$ are derived from linguistic if–then rules. This has enabled fuzzy logic control to be used in complex ill-defined processes, especially those that can be controlled by a skilled human operator without the knowledge of their underlying dynamics (Mamdani and Assilian, 1975).

Recall that *variable structure control* (VSC) is a form of discontinuous nonlinear control that alters the dynamics of a nonlinear system through the application of high-frequency switching control. A T1 FLC can also be regarded as a variable structure controller by virtue of the mathematics of fuzzy sets and systems; that is, it partitions the state space *automatically* rather than by a planned design. This is because different rules are activated for different regions of the state space. Palm (1992) showed that an FLC can be regarded as an extension of a conventional variable structure controller with a boundary layer.

There are two widely used architectures for a type-1 FLC that mainly differ in their fuzzy rule consequents. Those architectures, both of which are examined in this book, are:

- Mamdani FLC, developed by Mamdani and Assilian (1975) in which the antecedents and consequents of the rules are linguistic terms, for example: If x_1 *is Low and* x_2 *is High, then u is Low*. The linguistic labels in a Mamdani FLC are represented by type-1 fuzzy sets.
- Takagi–Sugeno (TS) FLC or Takagi–Sugeno–Kang (TSK) FLC (Takagi and Sugeno, 1985) in which the antecedents of the rules are also linguistic terms (modeled as type-1 fuzzy sets), but each rule's consequent is modeled as a mathematical function of the input variables, for example: *If* x_1 *is Low and* x_2 *is High, then* $u = g(x_1, x_2)$, where $g(x_1, x_2)$ is a polynomial function of x_1 and x_2 (this can include a constant, a linear or affine function, a quadratic function, etc.). An example of a first-order TSK FLC rule, the most widely used order, is: *If* x_1 *is Low and* x_2 *is High, then* $u = c_0 + c_1x_1 + c_2x_2$, where c_0, c_1, and c_2 are the consequent parameters.

1.4 WHAT IS A TYPE-2 FUZZY SET?

Because T1 FSs (e.g., as in Fig. 1.1b) are themselves crisp and precise (i.e., their MFs are supposedly known perfectly), this does not allow for any uncertainties about membership values, which is a potential shortcoming when using such fuzzy sets. A *type-2 fuzzy set* (T2 FS) is characterized by a fuzzy MF, that is, the membership value for each element of this set is itself a fuzzy set in [0,1]. The MFs of T2 FSs are three dimensional (3D) and include a *footprint of uncertainty* (FOU) (which is shaded in gray in Fig. 1.3a). It is the new third dimension of T2 FSs (e.g., Fig. 1.4c) and its FOU that provide additional degrees of freedom that make it possible to directly model and handle MF uncertainties.

In Fig. 1.3a, observe that the 26°C membership value in Hot is no longer a crisp value of 0.8 (as was the case in Fig. 1.1b); instead, it is a function that takes values from 0.6 to 0.8 in the primary membership domain, and maps them into a triangular distribution in the third dimension (Fig. 1.3b), called a *secondary MF*. This triangular secondary MF weights the interval [0.6, 0.8] more strongly over its middle values and less strongly away from those middle values. Of course, other weightings are possible, including equal weightings, in which case the T2 FS is called an *interval type-2 FS* (IT2 FS). Being able to choose different kinds of secondary MFs demonstrates one of the flexibilities of T2 FSs.

Figure 1.4c depicts the 3D MF of a general T2 FS whose secondary MFs [$f_x(u)$] are triangles. By convention, such a T2 FS is called a *triangular T2 FS*. Its FOU is depicted in Fig. 1.4a and its secondary MF at x' [$f_{x'}(u)$] is depicted by the solid triangle in Fig. 1.4b. When the secondary membership values equal 1 for all the primary membership values (as in the dashed curve in Fig. 1.4b), this results in an interval-valued secondary membership function, and, as just mentioned, the resulting T2 FS is called an IT2 FS. In Fig. 1.4c, $\mu(x, u)$ denotes the MF value at (x, u).

Figure 1.5 depicts the FOU of an IT2 FS for Low. The three dashed functions that are embedded within that FOU are T1 FSs. Clearly, one can cover this FOU with a multitude of such T1 FSs. At this point it is not important whether there are a

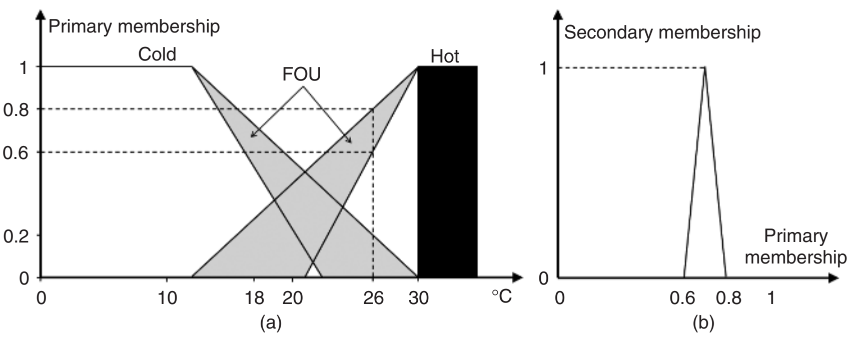


Figure 1.3 Type-2 fuzzy sets: (a) FOU and a primary membership and (b) a triangle secondary membership function.

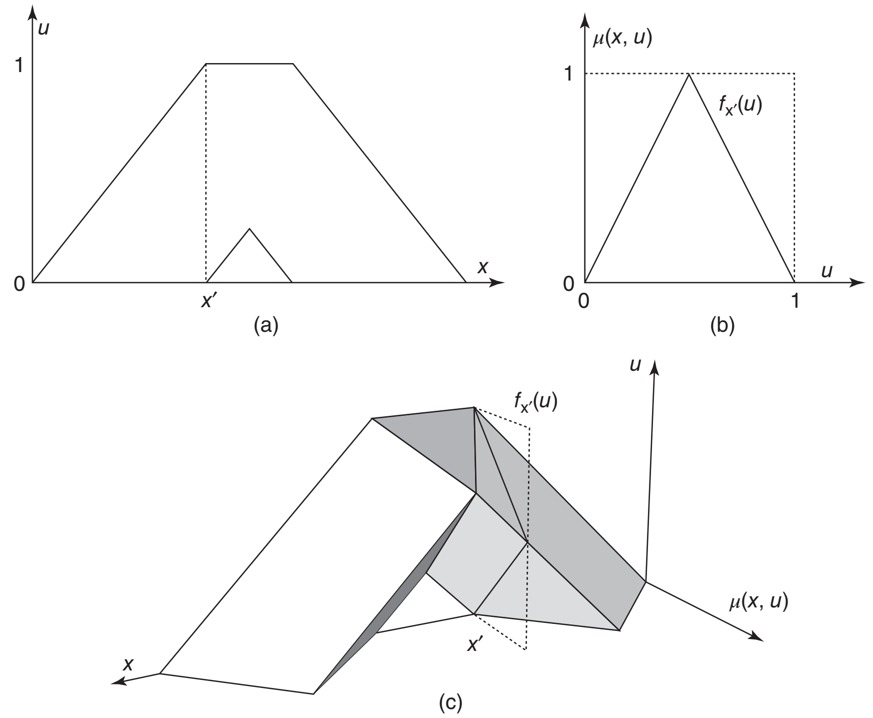


Figure 1.4 (a) FOU with primary membership (dashed) at x', (b) two possible secondary membership functions (triangle in solid line and interval in dashed line) associated with x', and, (c) the resulting 3D type-2 fuzzy set.

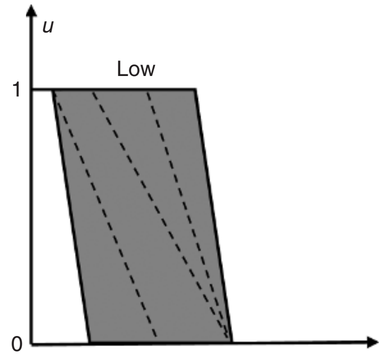


Figure 1.5 Three type-1 fuzzy sets that are embedded in the FOU of Low.

countable or uncountable number of such T1 FSs. What is important is interpreting an IT2 FS as the aggregation of a multitude of T1 FSs. This suggests that T1 FSs and everything that is already known about them can be used in derivations involving IT2 FSs, something that is exploited very heavily in this book. This interpretation also plays a very important role in understanding why an IT2 FLC may outperform a T1 FLC, something that we shall return to in the section below and in other chapters of this book.

1.5 WHAT IS A TYPE-2 FUZZY LOGIC CONTROLLER?

A type-2 FLC is depicted in Fig. 1.6. It contains five components: fuzzifier, rules, inference engine, type reducer, and defuzzifier. In a T2 FLC the inputs and/or outputs are represented by T2 FSs, and it operates as follows: crisp inputs, obtained from input sensors, are fuzzified into input T2 FSs, which then activate an inference engine that uses the same rules used in a T1 FLC to produce output T2 FSs. These are then processed by a type reducer that projects the T2 FSs into a T1 FS (this step is called *type reduction*) (Karnik et al., 1999; Liang and Mendel, 2000) after which that T1 FS is defuzzified to produce a crisp output that, for example, can be used as the command to an actuator in the control system. Type reduction followed by defuzzification is usually referred to as *output processing*.

In Section 1.3 we presented some sources of uncertainties that face real-world control systems in general. FLCs are also affected by:

- Linguistic uncertainties because the meaning of words that are used in the antecedents' and consequents' linguistic labels can be uncertain, that is, words mean different things to different FLC designers (Mendel, 2001).
- In addition, experts do not always agree and they often provide different consequents for the same antecedents. A survey of experts will usually lead to a histogram of possibilities for the consequent of a rule; this histogram represents the uncertainty about the consequent of a rule (Mendel, 2001).

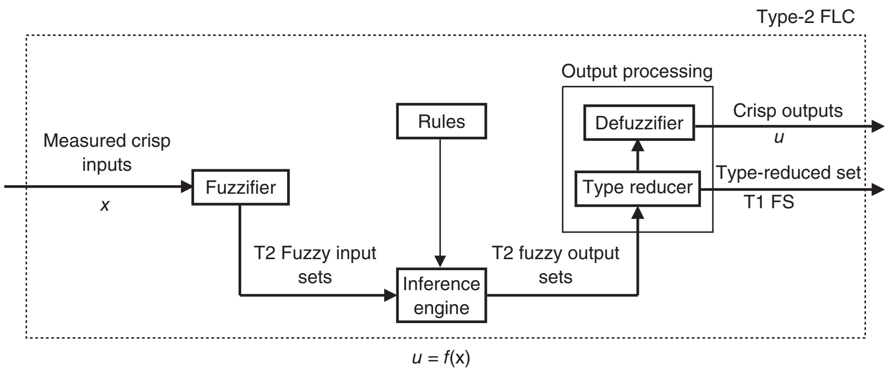


Figure 1.6 Overview of the architecture of a T2 FLC. The heavy lines with arrows indicate the path taken by signals during the actual operation of the FLC. Rules are used during the design of the FLC and are activated by the inference engine during the actual operation of the FLC (Mendel et al., 2006; © 2006, IEEE).

In a T2 FLC all of these uncertainties are modeled by the T2 FSs' MFs in the antecedents and/or consequents of the rules, as well as by the kind of fuzzifier. Regarding the latter, one may choose to use: (1) a singleton fuzzifier (as in a T1 FLC) in which a measured value is treated as perfect and is modeled as a crisp set; (2) a type-1 fuzzifier (as in a T1 FLC) in which a measured value is treated as signal plus stationary noise and is modeled as a normal, convex T1 FS (also called a *T1 fuzzy number*); or (3) a type-2 fuzzifier in which a measured value is treated as signal plus nonstationary noise and is modeled as a normal, convex T2 FS.

As we have explained in Section 1.4, a T2 FS can be thought of as a collection of many embedded T1 FSs (Mendel and John, 2002a). A T2 FLC may, therefore, be conceptually thought of as a collection of many (embedded) T1 FLCs whose crisp output is obtained by aggregating the outputs of all the embedded T1 FLCs (Karnik et al., 1999). Consequently, a T2 FLC has the potential to outperform a T1 FLC under certain conditions because it deals with uncertainties by aggregating a multitude of embedded T1 FLCs. The actual implementation of a T2 FLC does not actually require such an aggregation, but in this first chapter of this book, it is helpful to think of the output of a T2 FLC in this way.

Just as a T1 FLC is a variable structure controller so is a T2 FLC, and just as a T1 FLC has two architectures, Mamdani and TSK, a T2 FLC also has those two architectures. In a T2 Mamdani or TSK FLC, the fuzzy sets are type-2. Like their T1 FLC counterparts, T2 Mamdani and TSK FLCs are universal approximators (Ying, 2008, 2009). Both of these T2 FLC architectures will be covered in this book.

1.6 DISTINGUISHING AN FLC FROM OTHER NONLINEAR CONTROLLERS

Nonlinear control involves a nonlinear relationship between the controller's inputs and outputs and is more complicated than linear control; however, it is able to achieve better performance than linear control for many real-world control applications. Nonlinear control theory requires more challenging mathematical analysis and design than does linear control theory.

As mentioned in Section 1.3, an FLC is a nonlinear controller, that is, the function $f(\mathbf{x})$ is nonlinear. This will be demonstrated in later chapters of this book. What distinguishes an FLC, T1 or T2, from other nonlinear controllers is that it generates its nonlinear mapping function $f(\mathbf{x})$ through linguistic if–then rules and linguistic terms for the antecedents and consequents of the rules (e.g., Low Temperature, High Pressure). Such rules can be (easily) obtained from a human operator or can be postulated and learned from data. According to Kosko (1994), an FLC is unique in that it ties vague words like Low and High, and common sense rules, to state-space geometry.

According to Mamdani (1994), when tuned, the parameters of a PID controller affect the shape of the entire control surface. Because fuzzy logic control is a rule-based controller, the shape of the control surface can be individually manipulated for the different regions of the state space, thus limiting possible effects only to neighboring regions.

Fuzzy logic controllers have two important advantages over other classes of nonlinear controllers, namely (1) they are able to incorporate linguistic terms in the designs of the input–output membership functions, and (2) they are capable of handling uncertainties in inputs and state measurements more effectively. Moreover, similar to other classes of nonlinear controllers, they can be mathematically expressed, analyzed, and designed.

If the FLC rules are obtained from a group of experts, they may not all agree on the rule's consequents. By using T2 FSs, one is able to model the group's histogram of rule consequents, something that cannot be done by using a T1 FLC.

An FLC can be studied like any other nonlinear controller, for example, for the Mamdani FLC, stability and robustness studies can be performed by extensive simulations and by analyzing its control surface; see Fig.1.7, which depicts the mathematical function that maps robot controller inputs [e.g., right sensor front (RSF) and right sensor back (RSB)] into a control output (e.g., Steering). For a TSK FLC, it is possible to perform the same kinds of mathematical analyses that are applied to other nonlinear controllers, such as Lyapunov stability and robustness, and the like. Performance analyses of T2 Mamdani and TSK FLCs are given in later chapters of this book.

1.7 T2 FLCs VERSUS T1 FLCs

Type-1 FLCs use T1 FSs that have precise MFs, that is, there is nothing uncertain about such MFs. The following uncertainties that an FLC may encounter have been enumerated in Section 1.3: uncertainties about the inputs to the FLC, the control outputs, changing operating conditions of the controller, and disturbances acting upon the plant. Such uncertainties must somehow be mapped into MF uncertainties, and this is feasible to a greater extent in a T2 FLC than it is in a TI FLC because of the "noncrisp" nature of a T2 FS, the FOU for an IT2 FLC, or the combination of an FOU and secondary MFs for a general T2 FLC.

In addition to the above traditional kinds of uncertainties, which affect any kind of a controller, fuzzy or nonfuzzy, an FLC is also affected by the following additional uncertainties:

- Uncertainties about a rule's consequent, when rules are obtained from a group of experts, because, as we have mentioned above, experts do not generally all agree on the same consequent.
- Linguistic uncertainties about the meanings of the words used in a rule's antecedent and consequent linguistic terms, because *words mean different things to different people* (Mendel, 2001).
- Uncertainties associated with noisy training data that may be used to optimize (learn, tune) the MF parameters of an FLC.

It is difficult to directly model or minimize the effects of such uncertainties using T1 FSs. Consequently, using T1 FSs in an FLC may cause degradation in the performance of such a system.

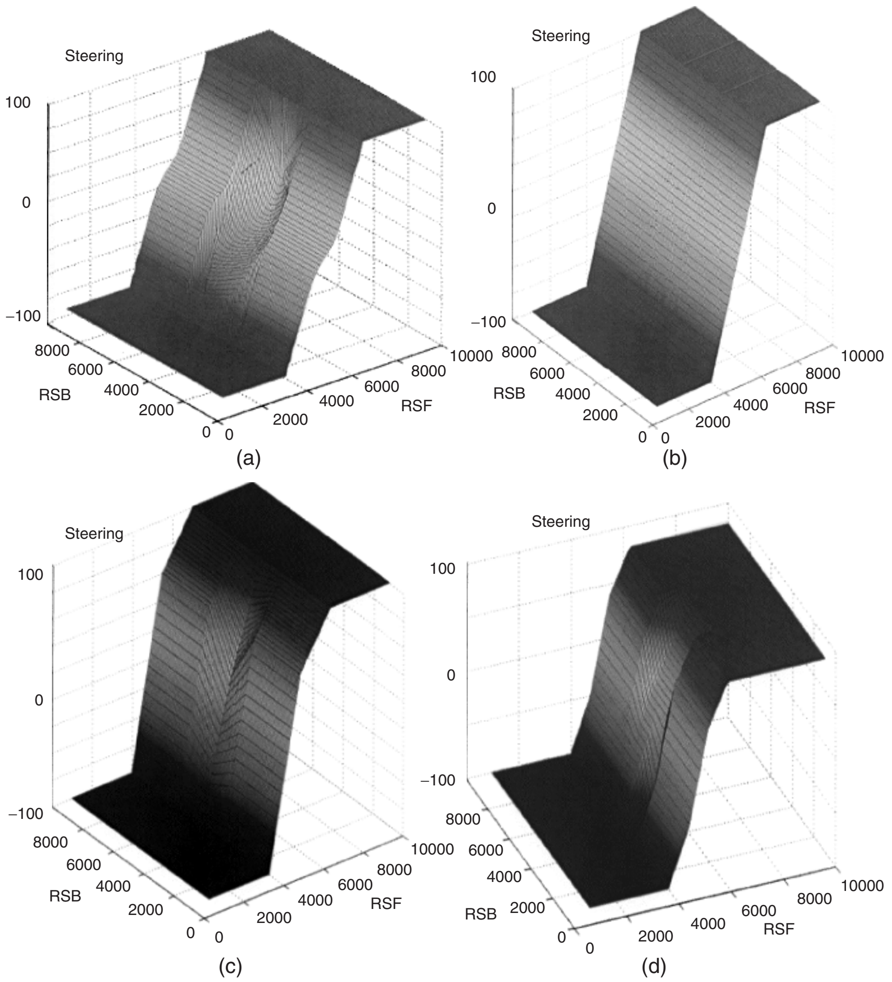


Figure 1.7 (a) Control surface of a robot T2 FLC with 4 rules, (b) control surface of a robot T1 FLC with 4 rules, (c) control surface of a robot T1 FLC with 9 rules, and (d) control surface of a robot T1 FLC with 25 rules (Hagras, 2004; © 2004, IEEE).

Because the MFs of a T2 FS are fuzzy, that is, have an FOU (and secondary MFs for a general T2 FS), they have more design degrees of freedom; hence, they have a greater *potential* to better model and handle all of the uncertainties just described in comparison to T1 FSs. Consequently, an FLC that is based on T2 FSs has the *potential* to produce better performance than a T1 FLC when dealing with such uncertainties. Observe that we have twice put emphasis on the word "potential." We have done this so as not to fool the reader into believing that a T2 fuzzy logic control system will always outperform a T1 fuzzy logic control system. The later chapters in this book will examine and compare the relative performances of both

T1 and T2 fuzzy logic control systems so that we may all better understand when or if a T2 fuzzy logic control system will outperform a T1 fuzzy logic control system.

As a preview to what will be demonstrated in those chapters, we note the following from Hagras (2004), Hagras (2007), and Wu (2012):

1. Using T2 FSs to represent the FLC inputs and outputs can lead to a smaller FLC rule base because MF uncertainties, represented by the FOUs of T2 FSs, let the T2 MFs cover the same range as T1 FSs, but with a smaller number of terms. This *rule reduction* (at the expense of more complicated MFs) increases as the number of FLC inputs increases.
2. A T2 FLC may give a *smoother control surface* than its T1 counterpart, especially in the region around the steady state [for a proportional–integral (PI) controller this means both the error and the change of error approach zero]. For example, Wu and Tan (2010) have shown that when a baseline T1 FLC implements a linear PI control law and the IT2 FSs of an IT2 FLC are obtained from symmetrical perturbations of the respective T1 FSs, the resulting IT2 FLC implements a variable gain PI controller around the steady state. These gains are smaller than the PI gains of the baseline T1 FLC, which results in a smoother control surface around the steady state. The PI gains of the IT2 FLC also change with the inputs, something that cannot be achieved by the baseline T1 FLC.
3. Type-2 FLCs *may realize more complex input–output relationships* than T1 FLCs. Karnik et al. (1999) pointed out that an IT2 fuzzy logic system can be thought of as a collection of many different embedded T1 fuzzy logic systems (as mentioned above). Additionally, Wu and Tan (2005) proposed a systematic method to identify the equivalent generalized T1 FSs that can be used to replace the FOU. They showed that the equivalent generalized T1 FSs are significantly different from traditional T1 FSs, and there are different equivalent generalized T1 FSs for different inputs. Du and Ying (2010) and Nie and Tan (2010) also showed that a symmetrical IT2 fuzzy PI [or the corresponding proportional–derivative (PD)] controller, obtained from a baseline T1 PI FLC, partitions the input domain into many small regions, and in each region the IT2 fuzzy PI controller is equivalent to a nonlinear PI controller with variable gains. The control law of the IT2 FLC in each small region is much more complex than that of the baseline T1 FLC, and hence it can realize more complex input–output relationships that cannot be achieved by a T1 FLC using the same rule base.
4. Type-2 FLCs have a *novelty* that does not exist in traditional T1 FLCs. Wu (2011) showed that in an IT2 FLC different membership grades from the same IT2 FS can be used in different rules (due to an IT2 FS being described by lower and upper MFs), whereas for a traditional T1 FLC the same membership grade from the same T1 FS is always used in the different rules. This further supports item 3, that an IT2 FLC can realize more complex input–output relationships than a T1 FLC, and that an IT2 FLC cannot be implemented by a T1 FLC using the same set of rules.

Figure 1.7, which shows control surfaces for an outdoor mobile robot, demonstrates how a T2 FLC with a rule base of only four rules (Fig. 1.7a) can produce a smoother control surface than its T1 counterparts that use a rule base of 4 (Fig. 1.7b), 9 (Fig. 1.7c), and 25 rules (Fig. 1.7d), respectively (Hagras, 2004). Observe, also, that as the T1 FLC rule base increases, its response approaches that of the T2 FLC because the latter includes a multitude of embedded type-1 FLCs.

1.8 REAL-WORLD APPLICATIONS OF IT2 MAMDANI FLCs

The last 10 years have witnessed a continuous increase in the deployment of IT2 Mamdani FLCs to real-world control problems. This trend promises to replicate the widespread use of type-1 FLCs to applications that touch the lives of people all over the world. The following subsections provide a brief *overview* of some of recent IT2 Mamdani FLCs for real-world control applications that are grouped into high-level application areas. We want to emphasize that all of the reported results are for specific systems and that we do not claim they apply universally. They are meant to whet the curiosity of the reader about potential performance improvements of IT2 FLC over T1 FLC, so as to encourage him or her to read the rest of this book.

1.8.1 Applications to Industrial Control

1.8.1.1 Speed Control of Marine Diesel Engines The first heavy-industry application of IT2 Mamdani FLCs was for the speed control of marine diesel engines (Lynch et al., 2005, 2006a, 2006b). These are huge engines classified according to their speeds, as slow-speed engines, medium-speed engines, or high-speed engines.

Due to their vast size and large power output, marine diesel engines require *accurate* and *robust* speed control/governing. Accurate speed control of such engines is of critical importance because significant deviations from the speed set point can be detrimental and damaging to the engine and its respective loads. Moreover, for applications such as power generation sets, the engine speed in revolutions per minute (rpm) must be stable in relation to multiples of the generated base frequency, that is, 50 Hz frequency requires the engine to operate at 1000 rpm, 1500 rpm, and so forth; hence, significant speed deviation can cause the generation of incorrect frequencies, resulting in loss of synchronization between the generator and its associated power grid, which is very problematic for any power generation system and its coupled loads.

Robustness in speed control is required for the marine diesel engine to overcome and recover quickly from the inherent instabilities and disturbances associated with the fast and dynamic changes of the environment, as well as load and operating conditions that marine diesel engines are exposed to on an everyday basis.

The ability to provide improved speed control response for marine diesel engines is not just desirable but is a requirement of the British Standard BS5514 “Reciprocating Internal Combustion Engines: Speed Governing,” which details regulations concerning the speed controller's ability to recover from load changes and disturbances in terms of settling time, overshoot, and undershoot (British Standards).

Marine diesel engines operate in highly dynamic and uncertain environments and experience vast changes in ambient temperature, fuel, humidity, and load. There are many sources of uncertainty facing speed controllers of marine diesel engines, including:

- Uncertainties associated with the change in engine operation and load conditions due to varying loads, weather and sea conditions, wind strength, hull fouling (growth of algae, sea grass, and barnacles), and vessel displacement (which is dependent on cargo). For example, the resistance (the force working against the ship propulsion) as a result of weather and sea variations can, in general, increase by as much as 100% of the total ship resistance in calm weather. Also, experience shows that hull fouling may cause an increase of up to 40% in ship resistance. An increase in ship resistance can consequently cause a drastic reduction of the ship's speed and significant vibration that can affect the engine's sensors and actuators. These uncertainties are considered to be the most dynamic and severe uncertainties that can affect both the inputs and output of the FLC and can cause serious degradation in the performance of the marine diesel engine.
- Uncertainties affecting the inputs to the controller, because sensor measurements are affected by high noise levels from various sources, such as electromagnetic and radio frequency interference, and vibration-induced triboelectric cable charges.
- Uncertainties affecting the outputs of the controller, which can be due to the change of the actuator's characteristics because of wear and tear or environmental changes, for example, worn linkages between the actuator output and the fuel pump can result in excessive friction and/or backlash causing instability in the control loop.
- Linguistic uncertainties because the meanings of the words that are used in the antecedent's and consequent's linguistic labels are inherently uncertain, since words mean different things to different engineers, which causes uncertainties when designing the FLC for marine diesel engine control.

Due to the size and cost of marine diesel engines it is important to test and verify the engine speed controllers under different operating and load conditions before their deployment on a specific engine.

Speed controllers can be tested and verified by using the testing platform depicted in Fig. 1.8. This testing platform is designed to realistically reflect the characteristics and operating conditions of the marine diesel engines and has the ability to alter speed,[1] load, inertia, and torque. It uses the real-world noisy sensors that are used by a specific marine diesel engine and has the ability to introduce the same uncertainty levels faced by that engine.

[1]The speed of a marine diesel engine is associated with the rate of fuel delivery to its cylinders, which is a function of a hydraulic servoactuator that is controlled by an electronic embedded speed controller.

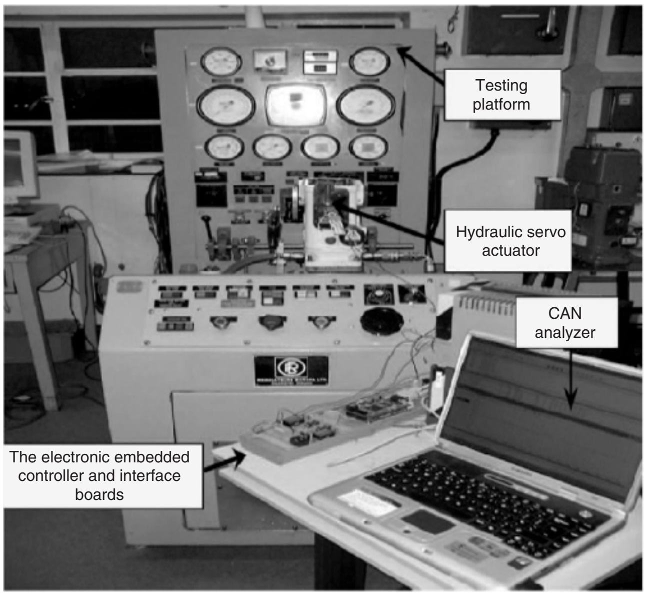


Figure 1.8 Marine diesel engine testing platform (Hagras, 2007; © 2007, IEEE).

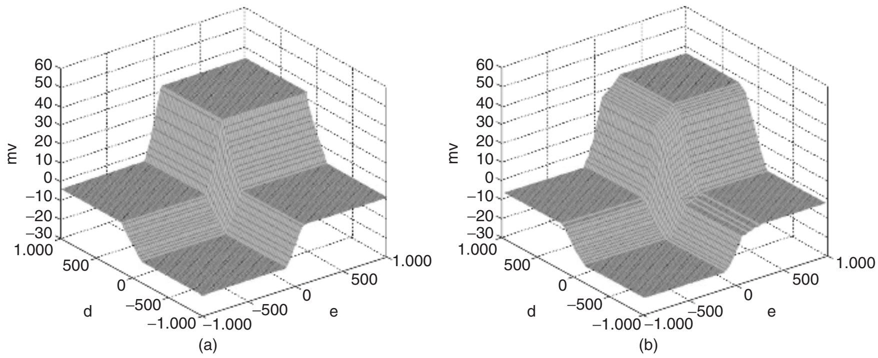


Figure 1.9 Control surfaces for (a) T1 Mamdani FLC and (b) IT2 Mamdani FLC (Hagras, 2007; © 2007, IEEE).

Figure 1.9a depicts the control surface for a T1 Mamdani FLC that was used in one of the marine diesel engine's speed controllers, and Fig. 1.9b depicts the control surface of an IT2 Mamdani FLC that was used for the same engine. Observe that the control surface for the T1 FLC is steep and nonsmooth, especially near the set point where the error (*e*) between the speed set point and the actual value, as well as the change of error (*d*), should both be equal to zero. Consequently, any small variations of *e* and *d* can cause considerable changes to the manipulated variable (*mv*) (i.e., the actuator controlling the fuel supply to the engine), which means that the T1 FLC is vulnerable to noise and uncertainties. Moreover, the larger the variations in e and d, due to the uncertainties, the larger are the disturbances to mv, which can cause instability and can potentially lead to the destruction of the engine.

The control surface that is depicted in Fig. 1.9b for the IT2 Mamdani FLC shows a very smooth and gradual response with no steep changes because it is (in theory) aggregating the outputs of a large number of embedded T1 FLCs. This smooth response gives very good control performance and can handle the uncertainties and disturbances that are near the set point where $e = 0$ and $d = 0$, that is, small variations in e and d do not cause significant changes to mv.

Many control experiments were performed in order to evaluate the performance of the IT2 and T1 Mamdani FLCs for handling uncertainties. The real operation of the diesel engines was mimicked where in each experiment the controllers were allowed to reach the set point and stabilize with no load, after which different loads were added suddenly to mimic the uncertainties associated with change of operation and load conditions. It is necessary for the diesel engine's speed controller to be able to deal quickly with the uncertainties associated with a change of load (for up to a 100% load addition) producing minimum overshoot/undershoot and settling times that must be in accordance with the British Standard BS5514 (British Standards).

In Lynch et al. (2006a), an IT2 Mamdani Real-Time Neuro-Fuzzy Controller[2] (RT2NFC) was developed. The performance of the RT2NFC was compared to the performances from a T1 FLC and a Viking 25 controller. The latter has been used in the past to control marine diesel engines and uses a PID algorithm with various nonlinear and gain-scheduling functions. Both the T1 and IT2 FLCs were coded in ANSI C and embedded in the industrial controller. For the engine testing platform, a set point of 905 rpm was chosen that corresponds with the requirements of medium-speed diesel engines.

All three controllers were tuned so that they could handle disturbances that were equivalent to 20% of the full load (which is a common disturbance for engines at a normal sea condition). It was noticed (not shown here) during the design process that the performances from all three controllers were very similar for the 20% load disturbance that they were designed to handle. However, as the uncertainty associated with the change of load increased to 100% load, the performance of both the Viking 25 and T1 FLC degraded significantly (see Fig. 1.10), producing large overshoots/undershoots as well as long settling times; hence, the performance of the Viking 25 and the T1 FLCs became unacceptable under these levels of uncertainties, which did not satisfy the desired standards.

A common practice in such situations is to retune the controller, which is a time-consuming process. The IT2 Mamdani FLC effectively handled the uncertainties associated with the change of the load and operating conditions to give a very good performance with small overshoots/undershoots as well as short settling times (see Fig. 1.10). The performance of the IT2 Mamdani FLC satisfied the required standards and required no further tuning. Therefore, the IT2 Mamdani

[2] A *neuro-fuzzy controller* is an FLC whose MF parameters are optimized using a tuning algorithm such as the back-propagation algorithm that is commonly used to tune the weights of a neural network.

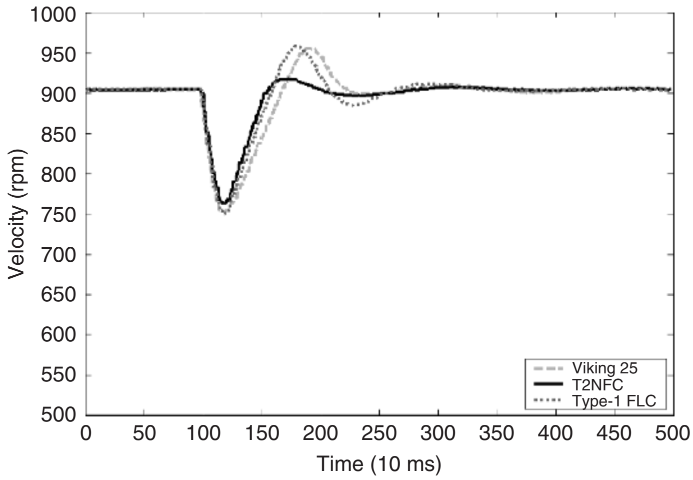


Figure 1.10 Comparison of the responses of the T1 FLC and Viking 25 against a T2NFC with 100% load addition (Lynch et al., 2006b; © 2006, IEEE).

FLC could be used effectively to produce accurate and robust speed controllers for marine diesel engines.

1.8.1.2 Liquid-Level Process Control In Wu and Tan (2004), a genetic algorithm[3] was used to design an IT2 Mamdani FLC to control a liquid-level process. The controlled process is the coupled tank apparatus depicted in Fig. 1.11a, which consists of two small tower-type tanks mounted above a reservoir that stores water that is pumped into the top of each tank by two independent pumps. The level of water in each tank is measured using a capacitive-type probe sensor, and each tank is outfitted with an outlet at the side near its base. Raising the baffle between the two tanks allows for water to flow between them. The amount of water that returns to the reservoir is approximately proportional to the square root of the height of the water column in the tank, and this is the main source of nonlinearity in this coupled-tank system. The volumetric flow rate of the pumps in the coupled-tank apparatus is nonlinear, and the system has nonzero transport delay.

It was observed (not shown here) that both the T1 and IT2 FLCs were able to attenuate oscillations when the modeling uncertainties were small. The liquid level in a tank eventually reached the desired set point, although the settling time was shorter when the IT2 FLC was used.

When, however, modeling uncertainties became larger, the T1 FLC gave rise to persistent oscillations (see Fig. 1.11b), whereas the IT2 FLC was able to eliminate these oscillations and the liquid level reached its desired height at steady state. Wu

[3]A *genetic algorithm* is a biologically inspired optimization algorithm that is used for tuning the MF parameters of the FLC as well as many other kinds of systems such as a neural network. See Section 3.6.2 for more details.

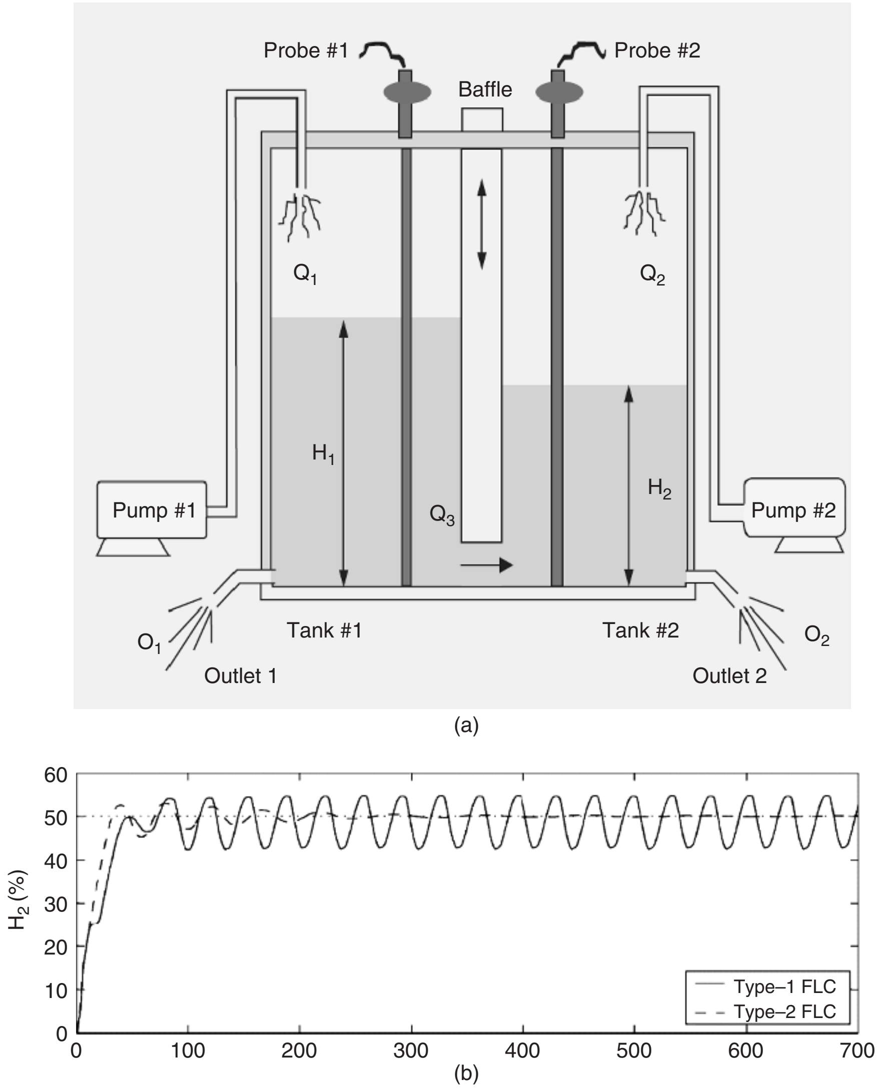
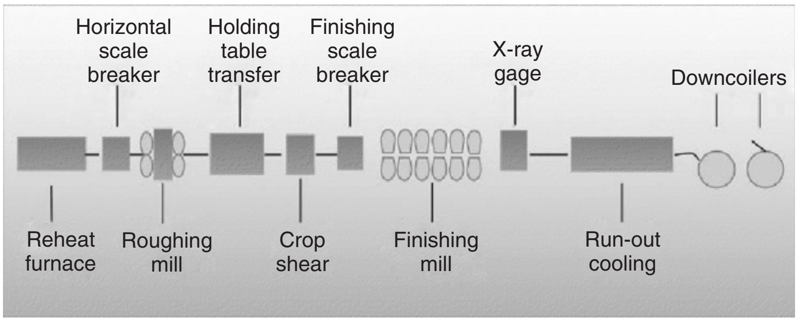


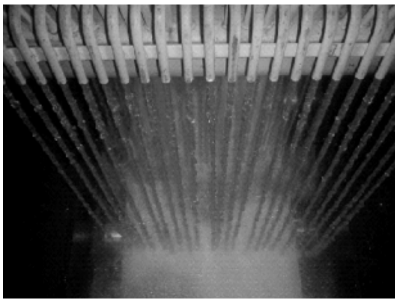
Figure 1.11 (a) Coupled-tank liquid-level control system and (b) T1 FLC (solid line) and IT2 FLC (dashed line) responses (Wu and Tan, 2004; © 2004, IEEE).

and Tan (2004) concluded that the IT2 FLC is more robust than the T1 FLC because the IT2 FLC outperformed its T1 FLC counterpart, especially when the uncertainty was large.

1.8.1.3 Control of Entry Temperature of a Steel Hot Strip Mill Mendez et al. (2010) applied a Mamdani IT2 FLC to control the coiling entry temperature of a steel hot strip mill (HSM). Figure 1.12a depicts an overview of an HSM from its



(a)



(b)

Figure 1.12 (a) Overview of a hot strip mill and (b) photo of a laminar cooling header at run-out table (Mendez et al., 2010).

initial stage at the reheat furnace entry to the final stage at the coiler side. In HSM there is a major need to satisfy quality requirements, for example, steel strip thickness, finishing temperature, and coiler temperature (the latter determines the final strip's mechanical properties). The most critical section of the coil is the head-end section due to the uncertainties involved at the head end of the incoming steel bar and the varying conditions from bar to bar.

As of 2010, in order to achieve head-end quality requirements, automation systems based on physical modeling were used, particularly for the reheat furnace, roughing mill (RM), finishing mill (FM), and the run-out cooling zone. As the market became more competitive, there was a need for flexible manufacturing capable of rolling a wider range of products in shorter periods of time. Such flexibility requirements yield higher time-varying conditions for the rolling process, thereby demanding automation systems that are better able to handle the encountered uncertainties. Most commercial systems employ proportional or proportional–integral controllers, which only compensate for the errors under current conditions; hence, the first batch in a given production cycle is usually below the given specifications.

A slab generally leaves the furnace at ~1200°C and is transported to the roughing mill by the transfer table. After several passes, the roughing stands adjust the slab thickness from ~200 to ~28 mm. The product from the roughing mill is called the *transfer bar*. The transfer bar is taken to the finishing mill where the finishing temperature and final width specifications have to be fulfilled. During the time the transfer bar travels from the roughing mill to the finishing mill scale forms on its surface. The scale breaker washes out the scale in order to allow proper rolling of the bar. Figure 1.12b shows a photograph of a top strip laminar cooling header. There are 34 top cooling headers divided into 6 sections of top header control. In addition, there are 27 bottom cooling headers divided into 3 sections of bottom spray control, giving 9 control sprays.

Strip resistance, and therefore force and gap setup, depend greatly on the strip temperature of the incoming bar, which is also essential for the speed setup, since strip temperature of the incoming bar depends on the entry bar thread speed, and the former is required to achieve both the specified finishing mill exit target head gauge and temperature. However, the bar surface temperature measurement at the scale breaker entry is not reliable due to scale formation and is therefore measured using a pyrometer located at the roughing mill exit side. Later, the head-end bar scale breaker entry temperature is estimated and used for the finishing mill and run-out cooling setup. The measurement at the roughing mill exit is affected by noise produced by transfer bar scale growth, environmental water steam, pyrometer location, calibration, resolution, and repeatability.

Experiments and results presented in Mendez et al. (2010) show that IT2 FLCs are able to model and control the cooling water flow to achieve the target coiler entry temperature in an HSM. They show that there is a substantial improvement in performance and stability of an IT2 Mamdani FLC over a T1 Mamdani FLC (e.g., Fig. 1.13). As can be seen from this figure, the IT2 FLC converged under real production conditions and had better performance in terms of the root-mean-square error (RMSE) than the T1 FLC. These results show the feasibility of the IT2 FLC for this particular industrial application.

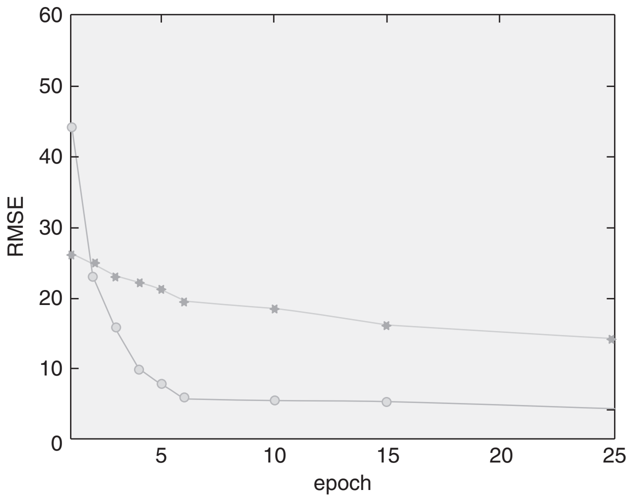
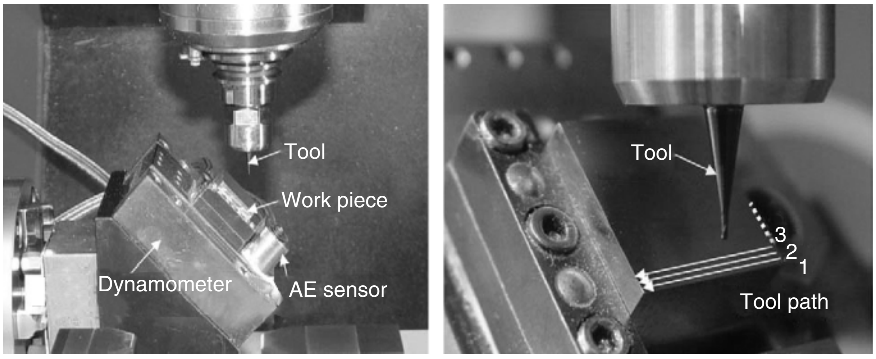


Figure 1.13 Root-mean-squared errors (RMSEs) for type-A cooling coil: (*) T1 FLC and (●) IT2 FLC models (Mendez et al., 2010).



(a) (b)

Figure 1.14 High-precision milling setup at Mondragón University (Spain). (a) Side view and (b) front view (Ren et al., 2010; © 2010, IEEE).

1.8.1.4 Modeling of Micromilling Cutting Forces Ren et al. (2010) designed an IT2 Mamdani FLC for the estimation of dynamic micromilling cutting forces. The resulting system was tested at the Micro-machining Laboratory at the Mondragón University in Spain. Figure 1.14 shows the actual setup. Researchers there noted that type-2 fuzzy estimation not only filters the noise and estimates the instantaneous cutting force in micromilling using observations acquired by sensors during cutting experiments but also assesses the uncertainties associated with the prediction caused by the manufacturing errors and signal processing. Moreover, the interval output of the type-2 fuzzy system gives very useful information to machine tool controllers in order to maximize material removal while controlling tool wear or tool failure to maintain part quality specifications.

1.8.1.5 Thyristor-Controlled Series Capacitor to Improve Power System Stability Tripathy and Mishra (2011) applied a Mamdani IT2 FLC to a thyristor-controlled series capacitor (TCSC) for improving power system stability. They report that the IT2 FLC along with the power system stabilizer (PSS) in the system satisfactorily damp out the speed and power oscillations following different critical faults. They show that the damping performance of the IT2 FLC is considerably better compared to its fixed gain bacteria-swarm-based tuned PSS and TCSC counterpart. Moreover, the performance of the IT2 FLC did not deteriorate even under uncertainty in the input signal to the controller, which shows the power of the IT2 Mamdani FLC in providing adequate performance even under conditions of increased uncertainty (in the inputs).

1.8.1.6 Control of Buck Direct-Current–Direct-Current (DC–DC) Convertors Lin et al. (2005) applied an IT2 Mamdani FLC to the control of buck DC–DC converters, which are nonlinear power electronic systems that convert one level of electrical voltage into another level by a switching action. They are used extensively in personal computers, computer peripherals, and adapters of consumer electronic devices.

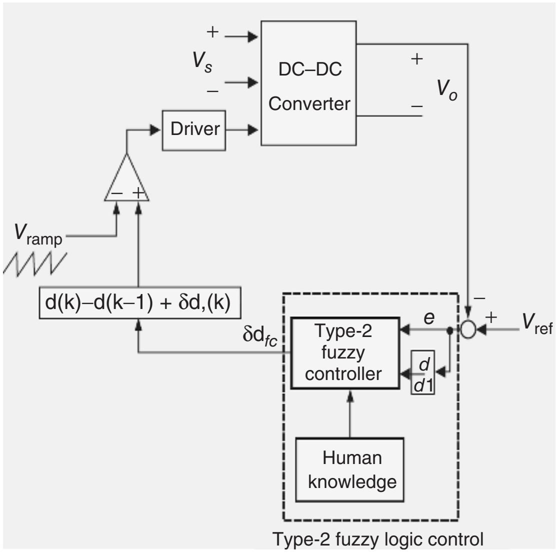


Figure 1.15 Block diagram of an IT2 FLC DC−DC converter system (Lin et al., 2005; © 2005, IEEE).

A control technique for DC−DC converters must cope with their wide input voltage and load variations to ensure stability in any operating condition while providing fast transient response. The control problem is to control the duty cycle so that the output voltage can supply a fixed voltage in the presence of input voltage uncertainty and load variations.

A block diagram of the IT2 Mamdani FLC DC−DC converter system is depicted in Fig. 1.15. Lin et al. (2005) have shown that the performance of an IT2 Mamdani FLC is better than its T1 counterpart, namely the rise time response of the IT2 Mamdani FLC is faster than that of T1 FLC and the former has no overshoot.

1.8.2 Airplane Altitude Control

Zaheer and Kim (2011) applied an IT2 Mamdani FLC to airplane altitude control for a propulsion-based airplane as shown in Fig. 1.16a. The throttle is used to regulate the speed of the airplane by varying the rotational speed of the propeller, the elevator is used to control the airplane's ascent and descent, the ailerons are used for airplane's lateral stabilization and midair turning, and the rudder is used for the on-ground taxiing of the airplane. They compared T1 and IT2 Mamdani FLCs for airplane control, and found that under high uncertainty levels, the IT2 Mamdani FLC outperformed the T1 FLC, namely that the T1 FLC showed oscillatory behavior around the reference altitude set points as shown in Figs. 1.16b and 1.16c.

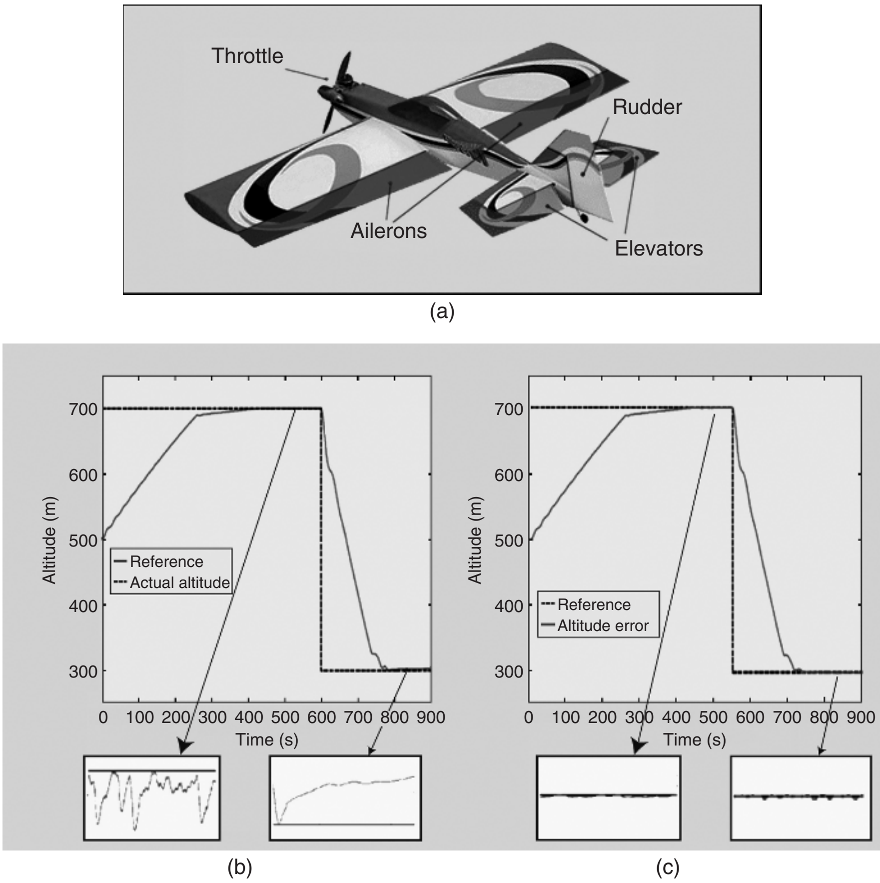


Figure 1.16 (a) Basic airplane control; (b) results of the T1 FLC in the simulation setup with uncertainties [bottom blocks are the magnified steady-state responses (RMSE = 3.58 m)]; and (c) results of IT2 Mamdani FLC in the simulation setup with uncertainties [bottom blocks are the magnified steady-state responses (RMSE = 0.43 m)] (Zaheer and Kim, 2011; © 2011, IEEE).

1.8.3 Control of Mobile Robots

Autonomous mobile robots navigating in real-world unstructured environments must be able to operate under conditions of imprecision and uncertainties present in such environments, where the uncertainties can be in the form of numerical uncertainties[4] (that affect the inputs and/or outputs of the controller). The numerical uncertainties associated with changing unstructured environments cause

[4]Numerical uncertainties refer to noise and change of the sensor signal due to change of operating conditions, for example, an ultrasound sensor assumes that the speed of sound is constant, however, the speed of sound varies with wind, rain, humidity, and the like, so a sonar sensor at a distance of 1 m will read different readings in wind, rain, and so forth.

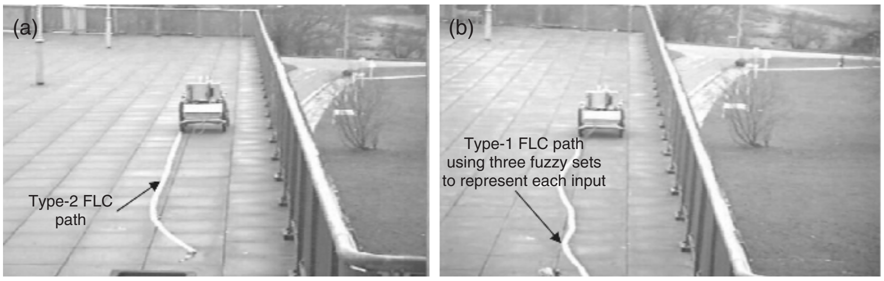


Figure 1.17 (a) Outdoor robot path using IT2 Mamdani FLC to implement the right edge following behavior to follow an irregular edge and (b) robot path using a T1 Mamdani FLC, which gave a poor response when the environment changed (windy weather) (Hagras, 2004; © 2004, IEEE).

problems in determining the exact and precise antecedents' and consequents' membership functions during the FLC design. The designed T1 fuzzy sets can be suboptimal for specific environment and robot conditions; however, as the robot operating conditions change from the design conditions, the T1 fuzzy sets will not be optimal any more, which can cause degradation in the mobile robot FLC performance. Hagras (2004) employed an IT2 Mamdani FLC for mobile robot control involving indoor and outdoor robots and found that the IT2 FLC always outperformed its T1 counterpart, and it also used a smaller number of rules. The former was demonstrated by examining robot paths and control surfaces (see Fig. 1.7). For the robot shown in Fig. 1.17a, the control surface of the IT2 Mamdani FLC has a smooth shape, which translated into a smooth control response that was able to deal effectively with uncertainty and imprecision. By means of control surface analyses, the more T1 fuzzy sets were used in the T1 FLC the more its response approached the smooth response of the IT2 Mamdani FLC (see Figs. 1.7b–1.7d). This is because the T2 fuzzy sets contain a large number of embedded T1 fuzzy sets, which allow for the detailed description of the control surface.

Hagras (2004) also performed experiments with robots in outdoor unstructured environments in order to evaluate the real-time performance of the robot IT2 FLC so as to see how they could handle large amounts of uncertainty and imprecision, as is present in such changing and dynamic environments. The robots were tested under different environmental conditions (e.g., rain, wind, sunshine), different ground conditions (e.g., slippery and dry ground), and at different times of the day. These experiments also involved using different challenging environmental features such as metallic and vegetation edges, which result in poor responses (i.e., echo) from the ultrasound sensor. They observed that the T1 FLC gave a good response under specific weather, ground, and robot conditions, but if any of these conditions changed, for example, when operating in windy weather conditions, then a nine-rule T1 FLC controlling the robot (see Fig. 1.17b) gave a poor oscillatory response because it could not handle the uncertainties associated with the outdoor environment conditions. On the other hand, they observed that the IT2 Mamdani FLC controlling the

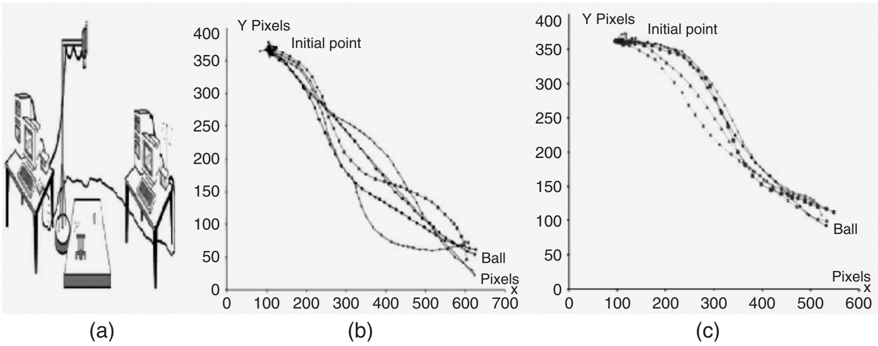


Figure 1.18 (a) Typical robot soccer platform, (b) player paths when a T1 FLC was used, and (c) player paths when an IT2 Mamdani FLC was used (Figueroa et al., 2005; © 2005, IEEE).

robot (see Fig. 1.17a) could handle such uncertainties and gave a better response while also using a smaller rule base.

Figueroa et al. (2005) described an IT2 Mamdani FLC for a robotic agent that tracks a mobile object in the context of robot soccer games, where the robotic agent has to track a ball accurately. In this application, the final goal of a player is to reach the position of the ball.

In robotic soccer games, positions of players and balls are captured through image processing because it is simple to do this. The basic configuration of a typical platform for robotic soccer games is shown in Fig. 1.18a; it comprises a football pitch (ground plane), a camera for image capture, one or two computers (server and client), and an radio frequency (RF) data transmitter.

Type-1 FLCs have been used in the past to control players; however, such FLCs face many sources of uncertainty, which include image processing algorithms (that cause uncertainties in the FLC inputs) as well as uncertainties in the actuators and networking resources. Hence, Figueroa et al. (2005) applied an IT2 Mamdani FLC to this problem and conducted two tests in order to evaluate the performance of the IT2 Mamdani FLC against its T1 counterpart.

The first test is called a *static ball test* and is one in which the way a "player" reaches the position of the ball is observed. During this test, the ball is positioned at a fixed point, for example, at one of the corners of the ground plane, and a player starts his movement from another point, usually the farthest corner. Figures 1.18b and 1.18c depict five static ball tests using the T1 and IT2 Mamdani FLCs, respectively. Observe that for both kinds of controllers the players' paths are always different (due to uncertainties); however, for the IT2 Mamdani FLC, the player only makes two corrections to reach the ball, whereas for the T1 FLC the player makes three corrections in order to reach the ball. Observe also that the paths followed by the T1 player have larger deviations than those of the T2 player, and that the shapes of those paths varied drastically. On the other hand, the paths followed by the T2 player were more regular. The control surface for the Mamdani IT2 FLC (not shown here) indicated that noisy sensors did not produce significant changes in a player's direction; however, for the T1 FLC, small variations in both the error and change of error produced a considerable change in direction, indicating that the T1 FLC was vulnerable to noise artefacts.

The second test is called a *mobile ball test* and is one in which the ball moves according to a defined trajectory and the player tries to track it. Figueroa et al. (2005) showed that, in all tests, the IT2 Mamdani FLC preserved a smaller average distance between the player and the moving ball. Additionally, they showed that the associated standard deviation was smaller for the IT2 Mamdani FLC than it was for the T1 FLC, which means that the paths followed by the IT2 player were closer to the ball's parabolic trajectory. They concluded, finally, that the IT2 Mamdani FLC was able to cope with uncertainties in a better way than the T1 FLC counterpart and also noted that the IT2 Mamdani FLC used a smaller rule base.

1.8.4 Control of Ambient Intelligent Environments

Ambient intelligence (AmI) provides basic criteria for the development of *ambient intelligent environments* (AIEs) in which intelligent computation that is enabled through simple and intuitive interactions with a user is invisibly embedded into the user's surrounding environments. The user is, therefore, empowered through a digital environment that is aware of her/his context and is sensitive, adaptive, and responsive to her/his needs in an unobtrusive manner.

Ambient intelligent environments rely on ubiquitous computing technologies that implement modular, low-powered devices and distributed high-bandwidth heterogeneous networks of sensors and actuators. They require distributed intelligence that uses modular units of intelligent behavior, such as intelligent agents, in order to create a pervasive distributed "layer of intelligence." Consequently, agents that are embedded in a user's environment (e.g., home, work, car, etc.) provide an intelligent "presence" by being able to recognize the user (or users) and autonomously program themselves to the users' needs by learning from their behaviors. The intelligence mechanisms employed within the agents must have low computational overheads, allowing them to be embedded into small hardware platforms or everyday consumer appliances. It is also important that these intelligent approaches provide their learned decisions in a form that is easily interpreted and analyzed by the end users.

One of the main underlying requirements for determining the kind of intelligent approach to use in the embedded agents is the ability to manage short-term and long-term uncertainties that arise due to changes in the environmental conditions along with changes in user behavior and activities over time. The AIEs face short-term uncertainties (within short-term time intervals) such as slight noise and imprecision associated with the inputs of the FLCs, as well as slight mood changes of the user. The AIEs also face long-term uncertainties because the environmental conditions and associated user activities change over longer durations of time due to:

- Seasonal variations in environmental conditions [e.g., external light level (the difference in the position of the sun can cause a difference between the late afternoon light levels in midsummer and the late afternoon light levels in midwinter), temperature, time of day (morning, afternoon, or evening)].

- People's behavior while occupying these environments because their behaviors, moods, and activities are dynamic, often nondeterministic and are subject to change with external factors such as time and season; there is also the fact that different words mean different things at different times of the year; for example, the values associated with *warm* temperature can vary from winter to summer.
- Changes in an actuator's characteristics as a result of wear and tear that occurs over time.

Hagras et al. (2007) describe an agent's architecture for the control of AIEs that uses an IT2 Mamdani FLC and a one-pass (noniterative) method to learn the user's particular behaviors and preferences in an online nonintrusive and seamless manner. The system learned the user's behavior by learning his/her particular rules and T2 membership functions. These rules and membership functions could then be adapted incrementally in a life-long learning mode to suit the changing environmental conditions and user's preferences. They developed a T2 agent architecture suitable for the embedded platforms used in AIEs, which have limited computational and memory capacities.

The agent based on IT2 Mamdani FLC was evaluated in the Essex Intelligent Dormitory (iDorm), depicted in Fig. 1.19a. The iDorm is a multiuser inhabited space that is fitted with a plethora of embedded sensors, actuators, processors, and heterogeneous networks that are cleverly concealed (buried in the walls and underneath furniture) so that the user is unaware of the hidden intelligent infrastructure of the room. It looks and feels like an ordinary study/bedroom environment, containing a mix of furnishings such as a bed, work desk, and wardrobe, which split the room into different areas of activity such as sleeping, working, and entertaining. Any networked embedded computer that can run a standard Java process can directly access and control the devices in the iDorm. The IT2 Mamdani FLC-based agent was embedded in an Internet Fridge (iFridge) computer.

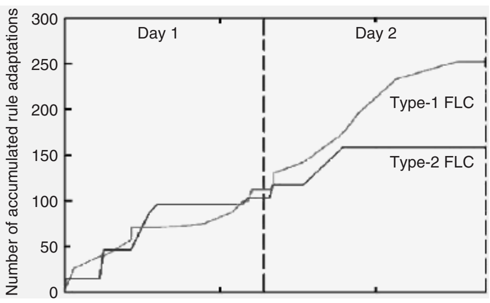


Figure 1.19 (a) iDorm and (b) number of accumulated online user adaptations (Hagras et al., 2007; © 2007, IEEE).

Experiments were conducted with various users during an extended period (spanning the course of a year) over which it was possible to evaluate and demonstrate how the agent could adapt in a life-long learning mode and could handle short- and long-term uncertainties. The agents based on IT2 Mamdani FLC were compared with T1-FLC-based agents regarding their ability to model a user's behavior while also handling long-term uncertainties. Results demonstrated that the IT2 FLC was better able to model a user's behavior and handle the short- and long-term uncertainties, and it used fewer rules than the T1 FLC.

Further experiments were conducted in the iDorm where *user satisfaction* was measured by monitoring how well the agents adjusted the iDorm environment to the user's preferences such that *user intervention* (which can be used as a measure of a user's satisfaction) was reduced over time. Figure 1.19b shows, for a two-day experiment, the number of rules that were adapted online every time the user had to override the agent's decision. Observe that agent based on the IT2 Mamdani FLC required significantly less user interaction than did the T1 agent. The curve for the T2 agent shows that user intervention initially was high but that it stabilized on the second day; therefore, the T2 agent only required the very short online tuning period of approximately one day. This is because the T2 agent better modeled user behavior and handled the short- and long-term uncertainties better than did the T1 agent. The curve for the T2 agent also shows it to be more stable (i.e., flat and not increasing with time) than the T1 agent in controlling the environment between the points when the user had to intervene in the agent's decisions to adapt the rules, that is, the curve for the T1 agent shows that user intervention continues to increase and does not properly stabilize by the end of the second day.

In conclusion, Hagras et al. (2007) show that T2 agents can adapt to user behaviors and that they generated fewer rules as compared with T1 agents. Fewer rules led to faster processing and more efficient memory usage. More specifically, the T2 agent was able to outperform the T1 agent achieving a 60% increase in processing speed as a result of a 50% reduction in the size of the rule base, thus reducing memory usage.

1.9 BOOK RATIONALE

Fuzzy control using familiar T1 FSs and logic has been extensively studied and applied to practical problems since 1974 and is considered a matured field. As mentioned above, fuzzy logic control relying on T2 FSs has now gained the attention of the fuzzy systems community, and the number of publications about it is growing rapidly.

Because of a lack of basic calculation methods in the early days of T2 FSs and logic, T2 FLCs have not emerged in popularity until recently. Now, T2 calculations can be done in real time.

As an emerging field, many different aspects of T2 fuzzy logic control need to be investigated in order to advance this new and powerful technology. This is the first book to bring together some of the latest developments on T2 fuzzy logic control in one place, so that interested researchers and practitioners can participate in this field. This book can be used to quickly understand the fundamentals of T2 fuzzy logic control and the latest theoretical developments about some important aspects of this new technology.

The central themes of any control methodology, fuzzy or conventional, are analysis and design. Analysis includes (1) describing the mathematical structure of T2 FLCs, (2) examining the stability of T2 fuzzy logic control systems, and (3) studying the robustness of T2 fuzzy logic control systems. Design means designing a T2 FLC (Mamdani or TSK) to control a given system to achieve user-desired performance, including stability. This book focuses on both topics for T2 FLCs and T2 fuzzy logic control systems, and also explains and demonstrates how to apply T2 fuzzy logic control to some important applications.

1.10 SOFTWARE AND HOW IT CAN BE ACCESSED

Software for Examples 4.1 and 4.6 and the examples in Chapter 6 can be accessed at http://booksupport.wiley.com/, and software for Appendix A, that supports T1, IT2 and GT2 FLCs, is available at http://juzzy.wagnerweb.net.

1.11 COVERAGE OF THE OTHER CHAPTERS

Chapter 2 provides background materials about IT2 FSs that are used in the rest of the book. To begin, T1 FSs are reviewed because T2 FSs build upon T1 FSs. Then a lot of information about interval T2 FSs is covered because this is needed in the rest of this book. Finally, general T2 FSs are introduced because such sets are the wave of the future.

Chapter 3 provides short reviews of T1 Mamdani and TSK FLCs so as to set the stage for the complete descriptions of IT2 Mamdani and TSK FLCs. These important IT2 FLCs are then developed in great detail, but using only T1 mathematics. The Wu–Mendel uncertainty bounds, which have let IT2 Mamdani FLCs run in real time, are stated; however, their derivations are included in Appendix 3A for completeness. Finally, some design methods for IT2 FLCs are described.

Chapter 4 describes techniques for rigorously deriving the precise mathematical relationships between the input and output of a variety of IT2 Mamdani and TSK FLCs. This is a relatively young area that started a few years ago. Some of the T2 FLCs are of the PI or PD type, and their derived relationships reveal them to be nonlinear variable PI or PD controllers that have variable proportional gain and integral gain (or derivative gain) plus variable control offset. Since many T1 fuzzy PI and PD controllers are already known to possess such structures, the structural characteristics of the T2 fuzzy PI controller can be (and are) compared to those of the corresponding T1 fuzzy PI controller. This chapter uses the derived relationships and structure characteristics analyses for insightfully understanding and studying the T2 FLCs and for developing their design guidelines.

Chapter 5 also focuses on the properties of IT2 proportional plus derivative (PD) and proportional plus integral (PI) FLCs. First, a class of IT2 PD/PI FLCs that has lower computational requirements, but still retains the properties previewed in Section 1.7, is introduced. The key idea is to only replace some critical T1 FSs by T2 FSs. Experimental results are presented that demonstrate the proposed simplified T2 FLC has the potential to be as robust as a conventional T2 FLC, while lowering the computational cost. Next, a methodology is presented, which is useful for theoretical studies, for deriving the analytical structure of IT2 PI/PD FLCs that have a symmetrical rule base. The methodology extends the analytical structure technique for T1 FLCs by leveraging a property of the Karnik–Mendel (KM) type reducer (which is derived and explained in Chapter 2) that constrains switch points to the locations of the consequent sets. Finally, examples are presented that illustrate how this framework may be applied to analyze IT2 FLCs.

Chapter 6 focuses on IT2 TSK FLCs. Its approach is based on rigorous mathematical analyses for both FLC analysis and design. It includes stability analysis and systematic methodologies for the design of adaptive and robust control, and introduces and provides some design approaches for practical control designs of such FLCs. Finally it includes several examples as well as an industrial application for modular and reconfigurable robotic systems.

Chapter 7 examines the future for T2 FLCs. Each of its sections has been written by one or more of the authors of this book and has a futuristic flavor.