Fuzzy Logic and Fuzzy Control
Fuzzy Logic & Fuzzy Control

• Introduction
• What is Fuzzy Logic?
• Introductory Example: Fuzzy vs. Non-Fuzzy
• Control
  – Overview
  – Conventional Control System Design
  – Fuzzy Control System Design
• Fuzzy Control: The Basics
  – Introduction
  – Choosing Fuzzy Controller Inputs and Outputs
  – Putting Control Knowledge into Rule Bases
  – Fuzzy Quantification of Knowledge
  – Matching: Determining Which Rules to Use
  – Inference Step: Determining Conclusions
  – Converting Decisions into Actions
  – Tuning via Scaling Universes of Discourse
  – Tuning Membership Functions
  – The Nonlinear Surface of the Fuzzy Controller
  – Summary: Basic Design Guidelines
References

• Fuzzy Control
  – Kevin Passino and Stephen Yurkovich
  – Addison Wesley, 1998

• Fuzzy Logic Toolbox
  – The Math Works, 2001
Introduction

- There has been a rapid growth in the number and variety of applications of fuzzy logic:
  - Consumer Products, e.g., cameras, camcorders, washing machines, microwave ovens
  - Industrial Process Control
  - Medical Instrumentation
- Fuzzy Logic is synonymous with the theory of fuzzy sets, a theory which relates to classes of objects with unsharp boundaries in which membership is a matter of degree.
• The basic concept underlying Fuzzy Logic is that of a linguistic variable, i.e., a variable whose values are words rather than numbers.

• Much of Fuzzy Logic may be viewed as a methodology for computing with words rather than numbers. Although words are inherently less precise than numbers, their use is closer to human intuition. Furthermore, computing with words exploits the tolerance for imprecision and thereby lowers the cost of solution.

• Another basic concept in Fuzzy Logic is that of a fuzzy if-then rule, or simply, fuzzy rule, and the calculus of fuzzy rules.
• Fuzzy Logic, neurocomputing, and genetic algorithms may be viewed as the principal constituents of soft computing, which unlike hard computing, is aimed at an accommodation with the pervasive imprecision of the real world.

• The guiding principle of soft computing is: Exploit the tolerance for imprecision, uncertainty, and partial truth to achieve tractability, robustness, and low solution cost.

• Soft computing is likely to play an increasingly important role in the conception and design of systems whose machine IQ is much higher than that of systems designed by conventional methods.
What is Fuzzy Logic?

• Fuzzy Logic is all about the *relative importance of precision*. How important is it to be exactly right when a rough answer will do?

• Relevant Quotes:
  – “Vagueness is no more to be done away with in the world of logic than friction in mechanics.” – C. Peirce
  – “So far as the laws of mathematics refer to reality, they are not certain. And so far as they are certain, they do not refer to reality.” – A. Einstein
  – “As complexity rises, precise statements lose meaning and meaningful statements lose precision.” – L. Zadeh
• Fuzzy Logic does a good job of trading off between significance and precision – something that humans have been managing for a very long time.

• Fuzzy Logic is both old and new! The modern and methodical science of fuzzy logic is still young, yet the concepts of fuzzy logic reach right down to our bones.
• Fuzzy Logic is a *convenient* way to map an input space to an output space, with emphasis on the word convenient.

• **Example:** You tell me how good your service was at a restaurant, and I’ll tell you what the tip should be.

• What could go in the black box? Fuzzy system, linear system, differential equations, lookup tables, neural networks, …
• Of the dozens of ways to make the black box work, it turns out that fuzzy is often the very best way! Why? “In almost every case you can build the same product without fuzzy logic, but fuzzy is faster and cheaper.” – L. Zadeh

• **Why use Fuzzy Logic?**
  – Fuzzy logic is conceptually easy to understand. It’s natural!
  – Fuzzy logic is flexible. It is easy to change without starting from scratch.
  – Fuzzy logic is tolerant of imprecise data. This is built into the process, rather than added onto the end.
– Fuzzy logic can model nonlinear functions of arbitrary complexity.
– Fuzzy logic can be built on top of the experience of experts. It lets you rely on the experience of people who already understand the system.
– Fuzzy logic can be blended with conventional control techniques. Fuzzy systems do not necessarily replace conventional control methods. In many cases, fuzzy systems augment them and simplify their implementation.
– Fuzzy logic is based on natural language, that which is used by ordinary people on a daily basis and which is convenient and efficient. The basis for fuzzy logic is the basis for human communication. As a result, fuzzy logic is easy to use.
• **When Not to Use Fuzzy Logic?**
  
  – Fuzzy logic is not a cure-all.
  
  – Fuzzy logic is a convenient way to map an input space to an output space. If you find that it is not convenient, try something else. If a simpler solution exists, use it.
  
  – Fuzzy logic is the codification of common sense – use common sense when you implement it and you will probably make the right decision.
  
  – Fuzzy logic can be a very powerful tool for dealing quickly and efficiently with imprecision and nonlinearity.
Introductory Example: Fuzzy vs. Non-Fuzzy

• Let’s look at two different approaches to the same problem: linear and fuzzy.

• Consider the restaurant tipping problem:
  – What is the right amount to tip your waitperson in a restaurant?
  – Given a number between 0 and 10 (where 10 is excellent) that represents the quality of service at a restaurant, what should the tip be?
  – The average tip for a meal in the U.S. is 15%.
• **Non-Fuzzy Approach**
  – Suppose that the tip always equals 15% of the total bill.
– This doesn’t take into account the quality of service. Let’s have the tip go linearly from 5% if the service is bad to 25% if the service is excellent.

\[
tip = \frac{0.20}{10} \text{ service} + 0.05
\]
Now, we may want the tip to reflect the quality of the food as well. Given two sets of numbers between 0 and 10 (where 10 is excellent) that respectively represent the quality of the service and the quality of the food at the restaurant, what should the tip be? Here is a formula to handle this situation:

\[
tip = \frac{0.20}{20} (\text{service} + \text{food}) + 0.05
\]
These results don’t seem quite right. Suppose you want the service to be a more important factor than food quality. Let’s say that the service will account for 80% of the overall tipping “grade” and the food will make up the other 20%. Here is a formula that handles this situation:

\[
\text{ServiceRatio} = 0.8
\]

\[
\text{tip} = \text{ServiceRatio} \left( \frac{0.20}{10} \text{ service} + 0.05 \right) + (1 - \text{ServiceRatio}) \left( \frac{0.20}{10} \text{ food} + 0.05 \right)
\]
A 3D graph showing the relationship between food and service quality and tip amount.
The response is still too uniformly linear. Suppose you want more of a flat response in the middle, i.e., you want to give a 15% tip in general, and will depart from this plateau only if the service is exceptionally good or bad. This, in turn, means that those nice linear mappings no longer apply. We can use a piecewise linear construction.

MatLab Code

```matlab
x=0:.5:10;
for i=1:length(x)
    if x(i)<3
        tip(i)=(0.10/3)*x(i)+0.05;
    elseif x(i)<7,
        tip(i)=0.15;
    elseif x(i)<=10,
        tip(i)=(0.10/3)*(x(i)-7)+0.15;
    end
end
```
If we extend this to two dimensions, where we take food into account again, we get the following results:

```matlab
x = service
y = food

x=0:.5:10;
y=0:.5:10;
for i=1:length(x)
  for j=1:length(y)
    if x(i)<3
      tip(i,j)=((0.1/3)*x(i)+0.05)*0.8+0.2*(0.2/10*y(j)+0.05);
    elseif x(i)<7,
      tip(i,j)=0.15*0.8+0.2*(0.2/10*y(j)+0.05);
    else,
      tip(i,j)=((0.10/3)*(x(i)-7)+0.15)*0.8+0.2*(0.2/10*y(j)+0.05);
    end
  end
end
```
• Wow! The plot looks good, but the function is surprisingly complicated.
• It was tricky to code this correctly, and it’s definitely not easy to modify this code in the future.
• Moreover, it’s even less apparent how the algorithm works to someone who didn’t witness the original design process.
• **Fuzzy Approach**
  – It would be nice if we could just capture the essentials of this problem, leaving aside all the factors that could be arbitrary.
  – If we make a list of what really matters in this problem, we might end up with the following rule descriptions:
    • If service is poor, then tip is cheap
    • If service is good, then tip is average
    • If service is excellent, then tip is generous
    • If food is rotten, then tip is cheap
    • If food is delicious, then tip is generous
  – The order of the rules does not matter.
  – These rules can be combined into one tight list of three rules:
• If service is poor or the food is rotten, then tip is cheap
• If service is good, then tip is average
• If service is excellent or food is delicious, then tip is generous

– These three rules are the core of our solution. These are the rules for a fuzzy logic system. If we give mathematical meaning to the linguistic variables, we would have a complete fuzzy inference system.

– Of course, there is a lot left to the methodology of fuzzy logic, e.g.,
  • How are the rules all combined?
  • How do I define mathematically what an “average” tip is?

– The details of the method do not change much from problem to problem – the mechanics of fuzzy logic are not terribly complex. Fuzzy is adaptable, simple, and easily applied.
Fuzzy Logic Solution to the Tipping Problem
Control

• **Overview**

• When confronted with a control problem for a complicated physical process, there is a relatively systematic control design procedure:
  – Gain an intuitive understanding of the plant’s dynamics and establish the design objective.
  – Develop a physical and mathematical model of the plant dynamics.
  – Use the mathematical model, or a simplified version of it, to design a controller.
– Use the mathematical model of the closed-loop system or simulation-based analysis to study the system performance (possibly leading to redesign).
– Implement the controller via a microprocessor, for example, and evaluate the performance of the closed-loop system (again, possibly leading to redesign).

• The procedure is concluded when the engineer has demonstrated that the control objectives have been met, and the controller is approved for manufacturing and distribution.

• **Fuzzy Control Design Methodology** can be used to construct fuzzy controllers for challenging real-world applications.
– In *conventional control* approaches (PID, lead-lag, state feedback) the focus is on modeling and the use of this model to construct a controller described by differential equations.

– In *fuzzy control*, we focus on gaining an intuitive understanding of how to best control the process, then we load this information directly into the *fuzzy controller*.

– Differential equations are the language of conventional control; heuristics and rules about how to control the plant are the language of fuzzy control.

– Are differential equations needed in the fuzzy control methodology? We will see how “conventional” the fuzzy control methodology really is and how many ideas from conventional control can be quite useful in the analysis of fuzzy control systems.
• Conventional Control System Design

• Feedback Control System

What are the steps taken to design the controller?

– Physical and Mathematical Modeling
– Performance Objectives and Design Constraints
– Controller Design
– Performance Analysis and Evaluation
• Physical and Mathematical Modeling
  – Remember! There is never a perfect model for the plant!
  – How does one generate a model?
    • First Principles of Physics
    • System Identification via the use of real plant data to produce a model of the system.
    • Combined Approach: Write down a general differential equation that we believe represents the plant behavior, and then perform experiments on the plant to determine certain model parameters or functions.
    • Truth Model: One that is developed to be as accurate as possible so it can be used in simulation-based evaluations of control systems.
    • Design Model: A low-order model for control design that may satisfy certain assumptions (e.g., linearity, certain nonlinearities) yet still capture the essential plant behavior.
• There are certain properties of the plant that the control engineer often seeks to identify early in the design process, e.g., stability of the plant, effects of certain nonlinearities, controllability of the plant, observability of the plant. These properties have a fundamental impact on our ability to design effective controllers.

• In addition, the engineer will try to make a general assessment of how the plant behaves under various conditions, how the plant dynamics may change over time, and what random effects are present.

  – The goal is a fundamental understanding of plant dynamics.
• Performance Objectives and Design Constraints

  – Controller design entails constructing a controller to meet specifications.
    • Open Loop vs. Closed Loop: Do not develop a feedback controller just because you are used to developing feedback controllers; you may want to consider an open-loop controller since it may provide adequate performance.

  – Assuming that you use feedback control, the closed-loop specifications (performance objectives) can involve the following factors:
    • Disturbance-rejection properties
    • Insensitivity to plant parameter variations
    • Stability
    • Rise-time
    • Overshoot
• Settling time
• Steady-state error

While these factors are used to characterize the technical conditions that indicate whether or not a control system is performing properly, there are other issues that must be considered that are often of equal or greater importance:

• Cost
• Computational complexity
• Manufacturability
• Reliability
• Maintainability
• Adaptability
• Understandability
• Politics
– It is important that the engineer has all these issues in mind early in the design process.
Controller Design

- Conventional control has provided numerous methods for constructing controllers for dynamic systems:
  - **PID control**: Over 90% of the controllers in operation today are PID controllers. This approach is often viewed as simple, reliable, and easy to understand. Often, like fuzzy controllers, heuristics are used to tune PID controllers (e.g., the Zeigler-Nichols tuning rules).
  - **Classical Control**: lead-lag compensation, Bode and Nyquist methods, root-locus design, …
  - **State-Space Methods**: state feedback, observers, …
  - **Optimal Control**: linear quadratic regulator, …
  - **Robust Control**: $H_2$ and $H_{\infty}$ methods, loop shaping, …
  - **Nonlinear Methods**: feedback linearization, sliding mode control, backstepping, …
• Adaptive Control: model reference adaptive control, self-tuning regulators, …

– These conventional approaches to control system design offer a variety of ways to utilize information from mathematical models on how to do good control. Sometimes they do not take into account certain heuristic information early in the design process, but use heuristics when the controller is implemented to tune it. Tuning is invariably needed!

– Engineers can sometimes become removed from the control problem and this can lead to unrealistic control laws. Sometimes in conventional control, useful heuristics are ignored because they do not fit into the proper mathematical framework. This can cause problems!
• **Performance Analysis and Evaluation**
  
  – We need performance evaluation to test that the control system that we design does in fact meet the closed-loop specifications. This is most important in safety-critical applications.

  – Basically, there are *three general ways* to verify that a control system is operating properly:

    • Mathematical analysis based on the use of formal models
    • Simulation-based analysis that most often uses formal models
    • Experimental investigations on the real system
– Mathematical Analysis

• In mathematical analysis you may seek to prove that the system is stable, controllable, or that other closed-loop specifications such as disturbance rejection, rise time, overshoot, settling time, and steady-state errors have been met.

• Clearly, however, there are several limitations to mathematical analysis.

• First, it always relies on the accuracy of the mathematical model, which is never a perfect representation of the plant, so conclusions that are reached from the analysis are in a sense only as accurate as the model that they are developed from. Mathematical analysis proves that properties hold for the mathematical model, not for the real physical system.

• Second, existing theory is somewhat lacking for the analysis of complex nonlinear control systems, particularly when there are significant nonlinearities, a large number of inputs and outputs, and stochastic effects.
- Simulation-Based Analysis

• In simulation-based analysis we seek to develop a simulation model of the physical system. This can entail using physics to develop a mathematical model and perhaps real data can be used to specify some of the parameters of the model (e.g., via system identification or direct parameter measurement). The simulation model can be made quite accurate and this “truth model” will be more complex than the model used for control design.

• There are always limitations on what can be achieved in simulation-based analysis.

• First, as with the mathematical analysis, the model that is developed will never be perfectly accurate.

• Second, some properties simply cannot be fully verified via simulation studies, e.g., asymptotic stability of an ordinary differential equation (simulations can only run for a finite amount of time and only a finite number of initial conditions can be tested for these finite-length trajectories).
• However, simulation-based studies can enhance our confidence that properties of the closed-loop system hold, and can offer valuable insights into how to redesign the control system before you spend time implementing the control system.

– **Experimental Investigations**

• To conduct an experimental investigation of the performance of a control system, you implement the control system for the plant and test it under various conditions.
• This can require significant resources, and for some plants you would not even consider doing an implementation until extensive mathematical and simulation-based investigations have been performed.
• Experimental evaluation does shed some light on issues like cost, reliability, and maintainability.
• Limitations include problems with repeatability of experiments and variations in physical components.
• Getting the system to actually work does enhance one’s confidence!
– Keep in mind that there are two basic reasons to do such analysis:

• First, we seek to verify that the designed control system will perform properly.
• Second, if it does not perform properly, then we hope that the analysis will suggest a way to improve the performance so that the controller can be redesigned and the closed-loop specifications met.
• **Fuzzy Control System Design**
  
  – What is the motivation for turning to fuzzy control?
  
  – Modeling and simulating complex real-world systems for control systems development is a difficult task.
  
  – Even if a relatively accurate model of a dynamic system can be developed, it is often too complex to use in controller development, especially for conventional control design procedures.
  
  – In practice, conventional controllers are developed via simple models of plant behavior and via ad hoc tuning of relatively simple linear or nonlinear controllers.
Heuristics enter the conventional control design process as long as you are concerned with the actual implementation of the control system and approaches that use appropriate heuristics to tune the design have been relatively successful.

**Questions?**

- How much of the success can be attributed to the use of modeling and conventional control design, and how much should be attributed to the clever heuristic tuning that the control engineer uses upon implementation?
- If we exploit the use of heuristic information throughout the entire design process, can we obtain higher performance control systems?
• Fuzzy control provides a formal methodology for representing, manipulating, and implementing a human’s heuristic knowledge about how to control a system.

• The fuzzy controller has four main components:
  – The rule-base holds the knowledge, in the form of a set of rules, of how best to control the system.
  – The inference mechanism evaluates which control rules are relevant at the current time and then decides what the input to the plant should be.
  – The fuzzification interface simply modifies the inputs so that they can be interpreted and compared to the rules in the rule base.
– The defuzzification interface converts the conclusions reached by the inference mechanism into inputs to the plant.

• The *fuzzy controller* is an artificial decision maker that operates in a closed-loop system in real time. It gathers plant output data $y(t)$, compares it to the reference input $r(t)$, and then decides what the plant input $u(t)$ should be to ensure that the performance objectives will be met.

• Where does the information come from to design the fuzzy controller?
  – A human decision maker that performs the control task
– The control engineer’s understanding of the plant dynamics

• **The rules basically say:**
  – If the plant output and reference input are behaving in a certain manner, then the plant input should be some value.

• A whole set of such **If-Then rules** is loaded into the rule base and an **inference strategy** is chosen. The system is then ready to be tested to see if the closed-loop specifications are met.
• **Modeling Issues and Performance Objectives**
  – Is a model needed to develop a fuzzy controller? Definitely YES!
  – A proper understanding of the plant dynamics cannot be obtained without applying first principles of physics to develop a mathematical model followed by a simulation-based evaluation.
  – When you carefully consider the possibility of ignoring the information frequently available in a mathematical model, it is clear that it will be unwise to do so.
  – So the role of modeling in fuzzy control design is quite similar to its role in conventional control system design.
  – In fuzzy control there is more significant emphasis on the use of heuristics.
In fuzzy logic there is a focus on the use of rules to represent how to control the plant rather than ordinary differential equations. The representation of knowledge in rules seems more lucid and natural to some people. There is simply a language difference between fuzzy and conventional control: ODE’s are the language of conventional control and rules are the language of fuzzy control.

The performance objectives and design constraints are the same as the ones for conventional control. We still want to meet the same types of closed-loop specification and the fundamental limitations that the plant provides still affect our ability to achieve high-performance control.
• **Fuzzy Controller Design**
  
  – Fuzzy controller design essentially amounts to:
    • Choosing the fuzzy controller inputs and outputs
    • Choosing the preprocessing that is needed for the controller inputs and possibly postprocessing that is needed for the outputs
    • Designing each of the four components of the fuzzy controller: fuzzification, rule base, inference mechanism, and defuzzification
  
  – There are standard choices for the fuzzification and defuzzification interfaces and most often the designer settles on an inference mechanism and uses it for many different processes. The main part of the fuzzy controller we focus on for design is the rule base.
– The rule base represents a human expert “in the loop.” The human expert could be someone who has spent a long time learning how best to control the process or could be the control engineer who studied the plant dynamics using modeling and simulation and wrote down a set of control rules that make sense.

– Generally speaking, if we load very detailed expertise into the rule base, we enhance our chances of obtaining better performance.
• **Performance Evaluation**
  - The ideas presented on performance evaluation for conventional controllers applies here as well, because the *fuzzy controller is a nonlinear controller*, so many conventional modeling, analysis, and design ideas apply directly.
  - **What value does fuzzy logic control have relative to conventional methods?** Detailed comparative analyses are few, and, moreover, most work in fuzzy control to date has focused only on its advantages and has not taken a critical look at what possible disadvantages there could be to using it.
  - The following questions are cause for concern when you employ a strategy of gathering heuristic control knowledge:
• Will the behaviors that are observed by a human expert and used to construct the fuzzy controller include all situations that can occur due to disturbances, noise, or plant parameter variations?
• Can the human expert realistically and reliably foresee problems that could arise from closed-loop system instabilities or limit cycles?
• Will the human expert be able to effectively incorporate stability criteria and performance objectives into a rule base to ensure that reliable operation can be obtained?
  – These questions are more troublesome in safety-critical situations or if the human expert’s knowledge is not as expert as we would hope.
  – So there is a need for a methodology to develop, implement, and evaluate fuzzy controllers to ensure that they are reliable in meeting their performance specifications.
• **Application Areas**

• **In engineering potential application areas include:**
  
  – **Aircraft/Spacecraft**
    
    • Flight control, engine control, failure diagnosis, navigation, satellite attitude control
  
  – **Automated Highway Systems**
    
    • Automatic steering, braking, and throttle control for vehicles
  
  – **Automobiles**
    
    • Brakes, transmission, suspension, and engine control
  
  – **Autonomous Vehicles**
    
    • Ground and underwater
  
  – **Manufacturing Systems**
    
    • Scheduling and deposition process control
– **Power Industry**
  - Motor control, power control/distribution, load estimation

– **Process Control**
  - Temperature, pressure, and level control, failure diagnosis, distillation column control, and desalination processes

– **Robotics**
  - Position control and path planning

- This is only representative of the range of possible applications of fuzzy control.
Fuzzy Control: The Basics

• **Introduction**
  
  The primary goal of control engineering is to distill and apply knowledge about how to control a process so that the resulting control system will reliably and safely achieve high-performance operation.

• Fuzzy Logic provides a methodology for representing and implementing our knowledge about how best to control a process.
Fuzzy Controller Architecture

Fuzzification → Inference Mechanism → Rule Base → Defuzzification → Process

Reference Input: r(t) → u(t) → y(t)

Inputs: r(t) → u(t)

Outputs: y(t)
• The fuzzy controller is composed of the following four elements:
  – Rule base (set of if-then rules) which contains a fuzzy logic quantification of the expert’s linguistic description of how to achieve good control.
  – Inference mechanism which emulates the expert’s decision making in interpreting and applying knowledge about how best to control the plant.
  – Fuzzification interface which converts controller inputs into information that the inference mechanism can easily use to activate and apply rules.
  – Defuzzification interface which converts the conclusions of the inference mechanism into actual inputs for the process.
**Simple Example: Balance an Inverted Pendulum on a Cart**

The goal is to balance the pendulum in the upright position when it initially starts with some nonzero angle off the vertical. We will use this system to illustrate the design and basic mechanics of the operation of a fuzzy control system.

Motor Parameters:
- \( J, B, T_f, K_T, K_A \)

Motor Operates in Torque Mode
Choosing Fuzzy Controller Inputs and Outputs

The fuzzy controller is to be designed to automate how a human expert who is successful at this task would control the system.

The expert will use the error and the rate of change of the error as the variables on which to base decisions. This choice makes good intuitive sense.

The controlled variable is the force that moves the cart – the only variable we are allowed to control.
• In general, you want to make sure that the controller will have the proper information available to make good decisions and have proper control inputs to be able to steer the system in the directions needed to be able to achieve high-performance operation.

• While in some academic problems, you may be given the plant inputs and outputs, in many practical situations you may have some flexibility in their choice. These choices affect what information is available for making on-line decisions about the control of a process and hence affect how we design a fuzzy controller.
• You must next determine what the reference input is. Here it is clear: $r(t) = 0$.

• The choice of the inputs and outputs of the controller places certain constraints on the remainder of the fuzzy control design process.

• The choice of the controller inputs and outputs is a fundamentally important part of the control design process.
• **Putting Control Knowledge into Rule Bases**

• Here we seek to take the linguistic description of how best to control the plant and load it into the fuzzy controller.

• **Linguistic Descriptions can be broken down into several parts:**
  – Linguistic Variables that describe each of the time-varying fuzzy controller inputs and outputs
    • “Error” \( e(t) \)
    • “Rate-of-change in the error” \( \frac{de(t)}{dt} \)
    • “Force” \( u(t) \)
  – The choice of the linguistic description of variables has no impact on the way the fuzzy controller operates.
Just as $e(t)$ takes on a numerical value, e.g., $e(2) = 0.1$, linguistic variables assume “linguistic values,” i.e., the values that linguistic variables take on over time change dynamically.

For the inverted pendulum system, the error, rate-of-change in the error, and force take on the following values:

- negative large or neglarge or -2
- negative small or negsmall or -1
- zero or 0
- positive small or possmall or +1
- positive large or poslarge or +2

The last type of linguistic variable is called a “linguistic-numeric value.”
The linguistic variables and values provide a language for the expert to express his/her ideas about the control decision-making process in the context of the framework established by our choice of fuzzy controller inputs and outputs.

For the inverted pendulum, the following statements quantify a different configuration of the pendulum:

- error is poslarge
- error is negsmall
- error is zero
- error is poslarge and rate-of-change in error is possmall
- error is negsmall and rate-of-change in error is possmall

We see that to quantify the dynamics of the process we need to have a good understanding of the physics of the underlying process we are trying to control.
We use the linguistic quantification to specify a set of linguistic rules (a rule base) that captures the experts knowledge about how to control the plant. 

For example:
- If error is neglarge and rate-of-change in error is neglarge Then force is poslarge
- If error is zero and rate-of-change in error is possmall Then force is negsmall
- If error is poslarge and rate-of-change in error is negsmall Then force is negsmall

A linguistic rule is formed from linguistic variables and values – neither the values nor the rules are precise; the rules are simply abstract ideas.
• The general form of the **linguistic rule** is:
  – IF premise (or antecedent) THEN consequent (or action)
  – The antecedents are associated with the fuzzy controller inputs.
  – The consequents are associated with the fuzzy controller outputs.
  – Each premise or consequent can be composed of the conjunction of several terms.
  – There does not need to be a premise (consequent) term for each input (output) in each rule, although often there is.
• Since we only specify a finite number of linguistic variables and values, there is only a finite number of possible rules.
• For the inverted pendulum system, there are two inputs and five linguistic values for each of these. Hence, there are at most $5^2 = 25$ possible rules.

• Shown is a tabular representation or rule table.

<table>
<thead>
<tr>
<th>force</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>-1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
</tr>
</tbody>
</table>

rate-of-change in error
• The table shows, for example, that:
  – IF error is poslarge AND rate-of-change in error is negsmall THEN force is negsmall

• Notice the pattern of rule consequents that appears in the body of the table. Notice the diagonal of zeros and viewing the body of the table as a matrix we see that it has a certain symmetry to it. This symmetry that emerges when the rules are tabulated is actually a representation of abstract knowledge about how to control the pendulum and is due to a symmetry in the system’s dynamics. This symmetry can be exploited in implementing fuzzy controllers.
• **Fuzzy Quantification of Knowledge**

- How do we fully quantify the meaning of linguistic descriptions so that we may automate, in the fuzzy controller, the control rules specified by the expert?

- We use **Membership Functions** to quantify (make less fuzzy) the meaning of the linguistic values.

• **For Example:** Possmall Membership Function

\[
\mu(\theta) = \begin{cases} 
1.0 & \text{if } \theta = -\pi / 2 \\
0.5 & \text{if } \theta = \pi / 8 \\
0.5 & \text{if } \theta = \pi / 4 \\
1.0 & \text{if } \theta = \pi / 2 \\
0 & \text{otherwise}
\end{cases}
\]

\( \mu \) is the certainty (degree of truth) that \( e(t) \) can be classified linguistically as “possmall.”
• The membership function quantifies, in a continuous manner, whether values $e(t)$ belong to (are members of) the set of values that are “possmall,” and hence it quantifies the meaning of the linguistic statement “error is possmall.”

• Depending on the application and the designer, many different choices of membership functions are possible. The definition of a membership function is subjective rather than objective. That is, we simply quantify it in a manner that makes sense to us, but others may quantify it in a different manner.
Membership Functions:

- The only condition a membership function must really satisfy is that it must vary between 0 and 1. The function itself can be an arbitrary curve whose shape we can define as a function that suits us from the point of view of simplicity, convenience, speed, and efficiency.

- The simplest membership functions are formed using straight lines. The simplest is the \textit{triangular} membership function – a collection of three points forming a triangle.

- The \textit{trapezoidal} membership function has a flat top and really is just a truncated triangle curve.

- These straight-line membership functions have the advantage of simplicity.
Straight-Line Membership Functions

trimf

trapmf

Gaussian and Bell Membership Functions

gaussmf

gauss2mf

gbellmf
Sigmoidal Membership Functions

sigmf

dsigmf

psigmf

Polynomial Membership Functions

zmf

pimf

smf

Fuzzy Logic & Fuzzy Control
Summary: Membership Functions
- *Gaussian* and *bell* membership functions are popular methods for specifying fuzzy sets. Both of these curves have the advantage of being smooth and nonzero at all points, however, they are unable to specify asymmetric membership functions, which are important in certain applications.

- *Sigmoidal* membership functions are either open left or open right. Asymmetric and closed (i.e., not open to the left or right) membership functions can be synthesized using two sigmoidal functions.

- *Polynomial-based curves* also serve as membership functions.

- You can probably get along very well with just one or two types of membership functions.
Summary of Membership Functions:

- Fuzzy sets describe vague concepts (fast runner, hot weather, weekend days).
- A fuzzy set admits the possibility of partial membership in it. (Friday is sort of a weekend day, the weather is rather hot).
- The degree an object belongs to a fuzzy set is denoted by a membership value between 0 and 1. (Friday is a weekend day to the degree 0.8).
- A membership function associated with a given fuzzy set maps an input value to its appropriate membership value.
• The set of values that is described by $\mu$ as being “positive small” is called a “fuzzy set.”
  – Fuzzy logic starts with the concept of a fuzzy set. A fuzzy set is a set without a crisp, clearly defined boundary. It can contain elements with only a partial degree of membership.
  – A classical set is a container that wholly includes or wholly excludes any given element.

Classical Set: Days of the Week

- Monday
- Thursday
- Saturday
- Liberty
- Shoe Polish
- Butter
– Everything falls into either one group or the other. There is no thing that is both a day of the week and not a day of the week.

– Now consider the set of days comprising the weekend.

Shoe Polish
Monday
Sunday
Saturday
Friday
Butter
Thursday
Liberty

Fuzzy Set: Days of the Weekend

– Friday feels like part of the weekend, but technically it should be excluded. Friday sits on the fence – fence sitting is part of life.
– We are entering the realm where sharp-edged yes-no logic stops being helpful. Fuzzy reasoning becomes valuable exactly when we are talking about how people really perceive the concept “weekend” as opposed to a simple-minded classification useful for accounting purposes only.

– *In fuzzy logic, the truth of any statement becomes a matter of degree.*

– Any statement can be fuzzy. The tool that fuzzy reasoning gives is the ability to reply to a yes-no question with a not-quite-yes-or-no answer.

– Reasoning in fuzzy logic is just a matter of generalizing the familiar yes-no (Boolean) logic. If True = 1 and False = 0, values like 0.73 and 0.22 are permitted.
Weekend-ness

Days of the Weekend: Two-Valued Membership

Days of the Weekend: Multi-Valued Membership

Membership Function
• **Another Example**: Consider the Set of Tall People
Sharp-Edged Membership Function for Tall

Continuous Membership Function for Tall
We can now specify the membership functions for all 15 linguistic values of our inverted pendulum example.
– Note that for the inputs $e(t)$ and $\frac{de(t)}{dt}$ the outermost membership functions “saturate” at a value of one. This makes intuitive sense as at some point the human expert would just group all large values together in a linguistic description such as “poslarge.”

– For the output $u$, the membership functions at the outermost edges cannot be saturated for the fuzzy system to be properly defined since we seek to take actions that specify an exact value for the process input. We do not generally indicate to a process actuator, “any value bigger than say, 10, is acceptable.”

– We see that the membership functions quantify the meaning of linguistic statements that describe time-varying signals.
• The rule base of the fuzzy controller holds the linguistic variables, linguistic values, their associated membership functions, and the set of all linguistic rules. Next we describe the fuzzification process.

• **Fuzzification**

• For most fuzzy controllers the fuzzification process is very simple and can be virtually ignored. It is simply the act of obtaining a value of an input variable (e.g., $e(t)$) and finding the numeric values of the membership function(s) that are defined for that variable.
• For example, if $e(t) = \pi/4$ and $de(t)/dt = \pi/16$, the fuzzification process amounts to finding the values of the input membership functions for these.

\[
\mu_{\text{possmall}}[e(t)] = 1
\]

\[
\mu_{\text{zero}}\left[\frac{d}{dt}e(t)\right] = \mu_{\text{possmall}}\left[\frac{d}{dt}e(t)\right] = 0.5
\]

• Some think of the membership function values as an “encoding” of the fuzzy controller numeric input values.

• The encoded information is then used in the fuzzy inference process that starts with “matching.”
Matching: Determining Which Rules to Use

How does the inference mechanism operate?

The inference process generally involves two steps:

– The premises of all the rules are compared to the controller inputs to determine which rules apply to the current situation. This “matching” process involves determining the certainty that each rule applies, and typically we will more strongly take into account the recommendations of rules that we are more certain apply to the current situation.

– The conclusions are determined using rules that have been determined to apply at the current time and are characterized with a fuzzy set (or sets) that represents the certainty that the input should take on various values.
• **Step 1: Premise Quantification via Fuzzy Logic**
  
  – Quantify each of the rules with fuzzy logic. First quantify the meaning of the premises of the rules that are composed of several terms, each of which involves a fuzzy controller input.

  – **For example:** IF error is zero AND rate-of-change in error is possmall THEN force is negsmall

```
“error is zero and change-in-error is possmall”
```

```
quantified with

0 “zero”

\( \mu_{\text{zero}} \)

\( e(t), \text{(rad.)} \)

\( \frac{d}{dt} e(t), \text{(rad/sec)} \)
```
– How do we quantify the AND operation?
– We know what’s fuzzy about fuzzy logic, but what about logic?
– Fuzzy logic reasoning is a superset of standard Boolean logic. If we keep the fuzzy values at their extreme of 1 (completely true) and 0 (completely false), standard logical operations will hold.
– Consider the standard truth tables.
– What function will preserve the results of the AND truth table and also extend to all real numbers between 0 and 1? One answer is the \textit{min} function. That is, resolve the statement A AND B, where A and B are limited to the range (0,1), by using the function \( \text{min}(A,B) \).

– Using the same reasoning, we can replace the OR operation with the \textit{max} function, so that A OR B becomes equivalent to \( \text{max}(A,B) \).
– Notice how the truth table is completely unchanged by this substitution.
– Moreover, since there is a function behind the truth table rather than just the truth table itself, we can now consider values other than 1 and 0.
– Returning to the inverted pendulum example, suppose:

\[
e(t) = \frac{\pi}{8} \quad \mu_{\text{zero}} \left[ e(t) \right] = 0.5
\]

\[
\frac{d}{dt} e(t) = \frac{\pi}{32} \quad \mu_{\text{possmall}} \left[ \frac{d}{dt} e(t) \right] = 0.25
\]

– What for these values is the certainty of the statement: “error is zero AND rate-of-change in error is possmall”?
– There are actually several ways to define it. We have already considered one way.

  • MINIMUM: Define $\mu_{\text{premise}} = \min\{0.5, 0.25\} = 0.25$, that is, using the minimum of the two membership values.
  • PRODUCT: Define $\mu_{\text{premise}} = (0.5)(0.25) = 0.125$, that is, using the product of the two membership values.

– Do these quantifications make sense? They both indicate that you can be no more certain about the conjunction of two statements than you can about the individual terms that make them up.

– Remember that $e(t)$ and $de(t)/dt$ change dynamically over time. When this occurs the values of $\mu_{\text{premise}}$ for each rule change, and hence the applicability of each rule in the rule base for specifying the force input to the cart, changes with time.
Multidimensional Membership Function of the Premise for a Single Rule

Represents how certain we are that a rule is applicable for specifying the force input to the plant.
• **Which Rules Are On?**
  
  - Determining the applicability of each rule is called “matching.” We say that a rule is “on at time t” if its premise membership function $\mu_{\text{premise}} > 0$.
  
  - The inference mechanism seeks to determine which rules are on to find out which rules are relevant to the current situation.

  - Suppose that: $e(t) = 0 \quad \frac{d}{dt} e(t) = \frac{\pi}{8} - \frac{\pi}{32} = 0.294$

  - We see that:
    \[
    \begin{align*}
    \mu_{\text{zero}} [e(t)] &= 1 \\
    \mu_{\text{zero}} \left[ \frac{d}{dt} e(t) \right] &= 0.25 \\
    \mu_{\text{possmall}} \left[ \frac{d}{dt} e(t) \right] &= 0.75
    \end{align*}
    \]
Input Membership Functions with Input Values
- All other membership functions are off.
- Rules are on that have the premise terms:
  - error is zero
  - rate-of-change in error is zero
  - rate-of-change in error is possmall
- Which rules are these? Check our table.

```
<table>
<thead>
<tr>
<th>force</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>-1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
</tr>
</tbody>
</table>
```

<table>
<thead>
<tr>
<th>( e(t) )</th>
<th>( de(t)/dt )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>-2</td>
<td></td>
</tr>
<tr>
<td>-3</td>
<td></td>
</tr>
<tr>
<td>-4</td>
<td></td>
</tr>
<tr>
<td>-5</td>
<td></td>
</tr>
</tbody>
</table>
• **Inference Step: Determining Conclusions**

• How do we determine which conclusions should be reached when the rules that are on are applied to deciding what the force input to the cart should be?

• First consider the recommendations of each rule independently. Then we will combine all the recommendations from all the rules to determine the force input to the cart.

• **Recommendation from Rule 1:**
  
  - IF error is zero AND rate-of-change in error is zero
  
  THEN force is zero

  - \( \mu_{\text{premise}(1)} = \min\{0.25, 1\} = 0.25 \)
(a) Consequent Membership Function for Rule 1
(b) Implied Fuzzy Set with Membership Function $\mu_{(1)}(u)$ for Rule 1

$$\mu_{(1)}(u) = \min\{0.25, \mu_{\text{zero}}(u)\}$$
– The justification for the use of the minimum operator to represent the implication is that we can be no more certain about our consequent than our premise. We could use the product operation to represent the implication also.

– Notice that the membership function $\mu_{(1)}(u)$ is a function of $u$ and that the minimum operation will generally “chop off the top” of the $\mu_{\text{zero}}(u)$ membership function to produce $\mu_{(1)}(u)$.

– We see that $\mu_{(1)}(u)$ is in general a time-varying function that quantifies how certain Rule 1 is that the force input $u$ should take on certain values.

– It is most certain that the force input should lie in a region around zero, and it indicates that it is certain that the force input should not be too large, either $+$ or $-$. 
– The membership function $\mu_{(1)}(u)$ quantifies the conclusion reached by only Rule 1 and only for the current $e(t)$ and $de(t)/dt$.

• **Recommendation from Rule 2:**
  – IF error is zero AND rate-of-change in error is possmall THEN force is negsmall
  – $\mu_{\text{premise}(2)} = \min\{0.75, 1\} = 0.75$
  – We are 0.75 certain that this rule applies to the current situation.
  – Notice that we are much more certain that Rule 2 applies to the current situation than Rule 1.
  – Rule 2 is quite certain that the control output should be a small negative value.
(a) Consequent Membership Function for Rule 2
(b) Implied Fuzzy Set with Membership Function $\mu_{(2)}(u)$ for Rule 2

$$\mu_{(2)}(u) = \min\{0.75, \mu_{\text{negsmall}}(u)\}$$
– As Rule 2 has a premise membership function that has a higher certainty than for Rule 1, we see that we are more certain of the conclusion reached by Rule 2.
– This completes the operations of the inference mechanism.
– While the input to the inference process is the set of rules that are on, its output is the set of implied fuzzy sets that represent the conclusions reached by all the rules that are on.
• **Converting Decisions into Actions**

• This is the defuzzification operation, the final component of the fuzzy controller.

• Defuzzification operates on the implied fuzzy sets produced by the inference mechanism and combines their effects to provide the “most certain” controller output (plant input).

• Some think of defuzzification as “decoding” the fuzzy set information produced by the inference process into numeric fuzzy controller outputs.

• First, draw all the implied fuzzy sets on one axis.
Implied Fuzzy Sets
- We want to find the one output, which we denote as $u^{\text{crisp}}$ that best represents the conclusions of the fuzzy controller that are represented with the implied fuzzy sets.

- There are actually many approaches to defuzzification. We will consider the "center of gravity" (COG) defuzzification method for combining the recommendations represented by the implied fuzzy sets from all the rules.

- Let $b_i$ denote the center of the membership function of the consequent rule (i). Here $b_1 = 0.0$ and $b_2 = -10$. Then

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

where $\int \mu_{(i)}(x)$ is the area under the membership function $\mu_{(i)}$. 
Defuzzification Methods

Methods:
- centroid of area
- bisector of area
- mean of maximum
- smallest of maximum
- largest of maximum

Fuzzy Logic & Fuzzy Control

K. Craig  110
– **Three items are important to note:**

- We cannot have output membership functions that have infinite area.
- Be careful to define the input and output membership functions so that the sum in the denominator is not equal to zero no matter what the inputs to the fuzzy controller are. Essentially, this means that we must have some sort of conclusion for all possible control situations we may encounter.
- The area is easy to compute for our example. For the case where we have symmetric triangular output membership functions that peak at one and have a base width of \( w \), simple geometry can be used to show that the area under a triangle “chopped off” at a height \( h \) is equal to:

\[
w \left( h - \frac{h^2}{2} \right)
\]
The input to the cart is then:

\[ u_{\text{crisp}} = \frac{(0)(4.375) + (-10)(9.375)}{4.375 + 9.375} = -6.81 \text{ N} \]

Does this result make sense?
Summary:
Graphical Depiction of Fuzzy Controller Operations
• **Tuning via Scaling Universes of Discourse**
  – We can use standard ideas from conventional control engineering to try to tune the fuzzy logic controller.
  – To do this we include gains on the inputs, as well as a gain between the fuzzy controller and the plant.
  – In the case of the inverted pendulum, we are having the same effect as varying the proportional and derivative gains of a classical PD controller.
  – The change in the scaling gains at the input and output of the fuzzy controller can have a significant impact on the performance of the resulting fuzzy control system, and hence they are often a convenient parameter for tuning.
  – What happens when these scaling gains are tuned?
– **Input Scaling Gains**

– The choice of scaling gain $g$ results in a scaling of the horizontal axis of the membership function.
  
  - If $g = 1$, there is no effect on the membership functions
  
  - If $g < 1$, the membership functions are uniformly spread out by a factor of $1/g$; the membership function is now characterized by larger numbers
  
  - If $g > 1$, the membership functions are uniformly contracted by a factor of $1/g$; the membership function is now characterized by smaller numbers

– The expansion and contraction of the horizontal axes by the input scaling gains is like the operation of an accordion, especially for triangular membership functions.

– So we can either use input scaling gains or simply redefine the horizontal axis of the membership functions.
We see that the input scaling factors have an inverse relationship in terms of their ultimate effect on scaling. Larger $g$ that is $>1$ corresponds to changing the meaning of the linguistics so that they quantify smaller numbers.

As we will see, just the opposite effect is seen for the output scaling gain.

**Output Scaling Gains**

Similarly, you can collapse the output scaling gain $h$ into the definition of the membership functions on the output.

- If $h = 1$, there is no effect on the output membership functions
- If $h < 1$, there is the effect of contracting the output membership functions and hence making the meaning of their associated linguistics quantify smaller numbers
• If $h > 1$, there is the effect of spreading out the output membership functions and hence making the meaning of their associated linguistics quantify larger numbers

– Overall, the tuning of scaling gains for fuzzy systems is often referred to as “scaling a fuzzy system.” Scaling gains can be used to normalize a fuzzy controller, i.e., the effective universes of discourse for all inputs and outputs are the same, e.g., $[-1, +1]$. 
• **Tuning Membership Functions**

  – Scaling gains are not the only parameters that can be tuned to improve the performance of the fuzzy control system.

  – Sometimes it is the case that for a given rule base and membership functions you cannot achieve the desired performance by tuning only the scaling gains. **Often what is needed is a more careful consideration of how to specify additional rules or better membership functions.**

  – The problem is that there are often too many parameters to tune, e.g., membership function shapes, positioning, and the number and type of rules, and often there is not a clear connection between the design objectives and a rationale and method that should be used for tuning.
– One method that has been found to be very useful for real implementations of fuzzy control systems for challenging systems is output membership function tuning.

– For example, making the output membership function centers near the origin be more closely spaced than the membership functions farther out on the horizontal axis will have the effect of making the “gain” of the fuzzy controller smaller when the signals are small and larger as the signals grow larger (up to the point where there is saturation, of course).

– This can help counter large disturbances while keeping the gains small when the input signals are small so as not to amplify noise in a real implementation.
• **The Nonlinear Surface of the Fuzzy Controller**
  
  – Ultimately the goal of tuning is to shape the nonlinearity that is implemented by the fuzzy controller.
  
  – This nonlinearity, sometimes called the control surface, is affected by all the main fuzzy controller parameters.
  
  – The surface represents in a compact way all the information in the fuzzy controller. Of course, this representation is limited in that if there are more than two inputs it becomes difficult to visualize the surface.
  
  – Note that a simple PD controller is a plane in three dimensions. With the proper choice of the PD gains, the linear PD controller can easily be made to have the same shape as the fuzzy controller near the origin. However, there is no way the linear PD controller can achieve a nonlinear control surface.
Inverted Pendulum Fuzzy Controller
Inverted Pendulum Fuzzy Controller

Fuzzy Logic & Fuzzy Control  K. Craig  122
Summary: Basic Design Guidelines

- These are basic design guidelines that are generic to all fuzzy controllers.
- Begin by trying a simple conventional PID controller. If this is successful, do not even try a fuzzy controller. The PID is computationally simpler and very easy to understand.
- Perhaps you should also try some other conventional control approaches, e.g., lead-lag or state feedback, if it seems that these may offer a good solution.
- For a variety of reasons, you may choose to try a fuzzy controller. Choose the proper inputs to the fuzzy controller.
– Try tuning the fuzzy controller using scaling gains.
– Try adding or modifying rules and membership functions so that you more accurately characterize the best way to control the plant. This can require significant insight into the physics of the plant.
– Try to incorporate higher-level ideas about how best to control the plant.
– If there is unsmooth or chattering behavior, you may have a gain set too high on an input to the fuzzy controller, or perhaps the output gain is too high. Setting the input gain too high makes it so that the membership functions saturate for very low values, which can result in oscillations.
– Sometimes the addition of more membership functions and rules can help. These can provide for a “finer” control, which can sometimes reduce chattering or oscillations.

– Sometimes it is best to first design a linear controller, then choose the scaling gains, membership functions, and rule base so that near the operating point the slope of the control surface will match the slope of the linear controller. Then we are incorporating all the good ideas that have gone into the design of the linear controller (about an operating point) into the design of the fuzzy controller. After this, the designer should seek to shape the nonlinearity for the case where the input signals are not near the operating point using insights about the plant.
– As with conventional control design, a process of trial and error is generally needed.

– If you are having trouble coming up with a good fuzzy controller, you probably need to gain a better understanding of the physics of the process you are trying to control, and then you need to get the knowledge of how to properly affect the plant dynamics into the fuzzy controller.