Takagi-Sugeno Fuzzy Modeling for Process Control

Kamyar Mehran

Industrial Automation, Robotics and Artificial Intelligence (EEE8005)

School of Electrical, Electronic and Computer Engineering

Newcastle University

· 2008 ·

Contents

Chapter 1

Introduction

Almost all of the physical dynamical systems in real life cannot be represented by linear differential equations and have a nonlinear nature. At the same time, linear control methods rely on the key assumption of small range of operation for the linear model, acquired from linearizing the nonlinear system, to be valid. When the required operation range is large, a linear controller is prone to be unstable, because the nonlinearities in the plant cannot be properly dealt with. Another assumption of linear control is that the system model is indeed linearizable and the linear model is accurate enough for building up the controller. However, the highly nonlinear and discontinuous nature of many, for instance, mechanical and electrical systems does not allow linear approximation. It is also necessary, in the design process of controllers, that the system model is well achievable through a mathematical model and the parameters of the system model are reasonably well-known. Nevertheless, for many nonlinear plans i.e. chemical processes, building a mathematical model is very difficult and only the input-output data yielded from running the process is accessible for an estimation. Many control problems involve uncertainties in the model parameters. A controller based on inaccurate or obsolete values of the model parameters may show significant performance degradation or even instability. There are some complicated approaches like auto-regressive model based on the input-output data to compensate model uncertainties, which usually use to design a process control. However due to the high nonlinearity of the process, the order of the model often becomes very high so that past effects are taken into account, even if that is physically unrealistic.

One way to cope with such difficulty is to develop a nonlinear model composing of a number of sub-models which are simple, understandable, and responsible for respective sub-domains. The idea of multi-model approach [1] is not new, but the idea of fuzzy modeling [2] using the concept of the fuzzy sets theory [3] offers a new technique to build multi-models of the process based on the input-output data or the original mathematical model of the system. Facing complex and nonlinear systems, we have to recognize that modeling is an art and it is important to realize system modeling is generally an act to understand things directly rather than by computer. At most a linear combination like a fuzzy model is clearly understandable.

The fuzzy model proposed by Takagi and Sugeno [2] is described by fuzzy IF-THEN rules which represents local input-output relations of a nonlinear system. The main feature of a Takagi-Sugeno fuzzy model is to express the local dynamics of each fuzzy implication (rule) by a linear system model. The overall fuzzy model of the system is achieved by fuzzy "blending" of the linear system models. In this tutorial, the reader will find, by some examples, that almost all nonlinear dynamical systems can be represented by Takagi-Sugeno fuzzy models to high degree of precision. In fact, it is proved that Takagi-Sugeno fuzzy models are universal approximators of any smooth nonlinear system [4, 5].

Chapter 2

Takagi-Sugeno fuzzy modeling

A fuzzy controller or model uses fuzzy rules, which are linguistic if-then statements involving fuzzy sets, fuzzy logic, and fuzzy inference. Fuzzy rules play a key role in representing expert control/modeling knowledge and experience and in linking the input variables of fuzzy controllers/models to output variable (or variables). Two major types of fuzzy rules exist, namely, Mamdani fuzzy rules and Takagi-Sugeno (TS, for short) fuzzy rules.

Lets first start with the familiar Mamdani fuzzy systems. A simple but representative Mamdani fuzzy rule describing the movement of a car is:

IF *Speed* is *High* AND *Acceleration* is *Small* THEN *Braking* is (should be) *Modest*,

where *Speed* and *Acceleration* are input variables and *Braking* is an output variable. "High," "Small," and "Modest" are fuzzy sets, and the first two are called input fuzzy sets while the last one is named the output fuzzy set.

The variables as well as linguistic terms, such as "High", can be represented by mathematical symbols. Thus, a Mamdani fuzzy rule for a fuzzy controller involving three input variables and two output variables can be described as follows:

$$\text{IF } x_1 \text{ is } M_1 \text{ AND } x_2 \text{ is } M_2 \text{ AND } x_3 \text{ is } M_3 \text{ THEN } u_1 \text{ is } M_4,\ u_2 \text{ is } M_5, \tag{2.1}$$

where x_1, x_2, and x_3 are input variables (e.g., error, its first derivative and its second derivative), and u_1 and u_2 are output variables (e.g., valve openness). In theory, these variables can be either continuous or discrete; practically speaking, however, they should be discrete because virtually all fuzzy controllers and models are implemented using digital computers. M_1, M_2, M_3, M_4, and M_5 are fuzzy sets, and AND are fuzzy logic AND operators. "IF x_1 is M_1 AND x_2 is M_2 AND x_3 is M_3" is called the **rule antecedent**, whereas the remaining part is named the **rule consequent**.

The structure of Mamdani fuzzy rules for fuzzy modeling is the same. The variables involved, however, are different. An example of a Mamdani fuzzy rule for fuzzy modeling is

$$\text{IF } y(n) \text{ is } M_1 \text{ AND } y(n-1) \text{ is } M_2 \text{ AND } y(n-2) \text{ is } M_3 \text{ AND } u(n) \text{ is } M_4 \text{ AND } u(n-1) \text{ is } M_5$$
$$\text{THEN } y(n+1) \text{ is } M_6 \tag{2.2}$$

where M_1, M_2, M_3, M_4, M_5, and M_6 are fuzzy sets, $y(n)$, $y(n-1)$, and $y(n-2)$ are the output of the system to be modeled at sampling time n, $n-1$ and $n-2$, respectively. And, $u(n)$ and $u(n-1)$ are system input at time n and $n-1$, respectively; $y(n+1)$ is system output at the next sampling time, $n+1$.

Now, let us look at the so-called TS fuzzy rules. Unlike Mamdani fuzzy rules, TS rules use functions of input variables as the rule consequent. For fuzzy control, a TS rule corresponding to the Mamdani rule (2.1) is

$$\text{IF } x_1 \text{ is } M_1 \text{ AND } x_2 \text{ is } M_2 \text{ AND } x_3 \text{ is } M_3 \text{ THEN } u_1 = f(x_1, x_2, x_3), \quad u_2 = g(x_1, x_2, x_3),$$

where $f()$ and $g()$ are two real functions of any type. Similarly, for fuzzy modeling, a TS rule analogous to the Mamdani rule (2.2) is in the following form:

$$\text{IF } y(n) \text{ is } M_1 \text{ AND } y(n-1) \text{ is } M_2 \text{ AND } y(n-2) \text{ is } M_3 \text{ AND } u(n) \text{ is } M_4 \text{ AND } u(n-1) \text{ is } M_5$$

$$\text{THEN } y(n+1) = F(y(n), y(n-1), y(n-2), u(n), u(n-1)),$$

where $F()$ is an arbitrary function.

Fuzzy inference systems also known as fuzzy rule-based systems or fuzzy models are schematically shown in Figure 2.1. They are composed of 5 conventional block: a **rule-base** containing a number of fuzzy if-then rules, a *database* which defines the membership functions of the fuzzy sets used in the fuzzy rules, a *decision-making unit* which performs the inference operations on the rules, a *fuzzification interface* which transform the crisp inputs into degrees of match with linguistic values, a *defuzzification interface* which transform the fuzzy results of the inference into a crisp output.

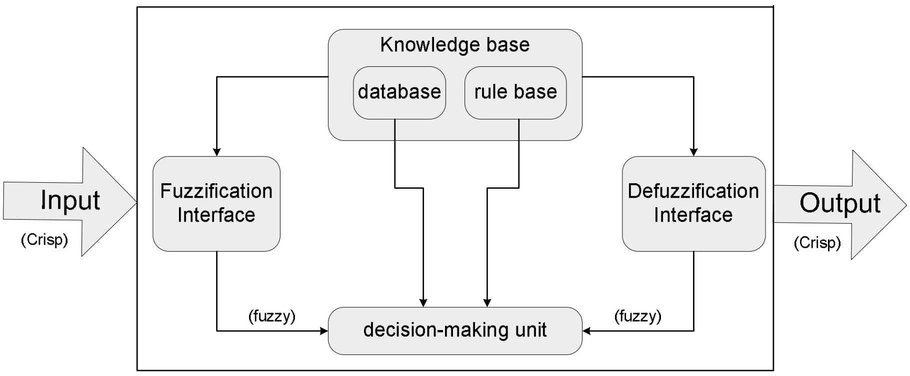


Figure 2.1: Fuzzy inference system.

Figure 2.2 utilizes a two-rule two-input fuzzy inference system to show different types of fuzzy system mentioned above. Type 2 is the widely-used Mamdani type fuzzy system which the output function is determined based on overall fuzzy output; some of them are centroid of area, min of maxima, maximum of maxima, etc. Type 3 is the Takagi-Sugeno type fuzzy system.

In this tutorial, we focus only on fuzzy models that use the T-S rule consequent.

2.1 Construction of Fuzzy Models

Figure 2.3 illustrates the model-based fuzzy control design approach. To design a T-S fuzzy controller, we need a T-S fuzzy model for a nonlinear system. Therefore the construction of a fuzzy model represent an important and basic procedure in this approach. In general there are two approaches for constructing fuzzy models:

1. Identification (fuzzy modeling) using input-output data and
2. Derivation from given nonlinear system equations.

There has been an extensive literature of fuzzy modeling using input-output data following Takagi's, Sugeno's and Kang's excellent work [6, 7]. The procedure mainly consist of two parts: structure identification and parameter identification. The identification approach to fuzzy modeling is suitable for plants that are unable or too difficult to be represented

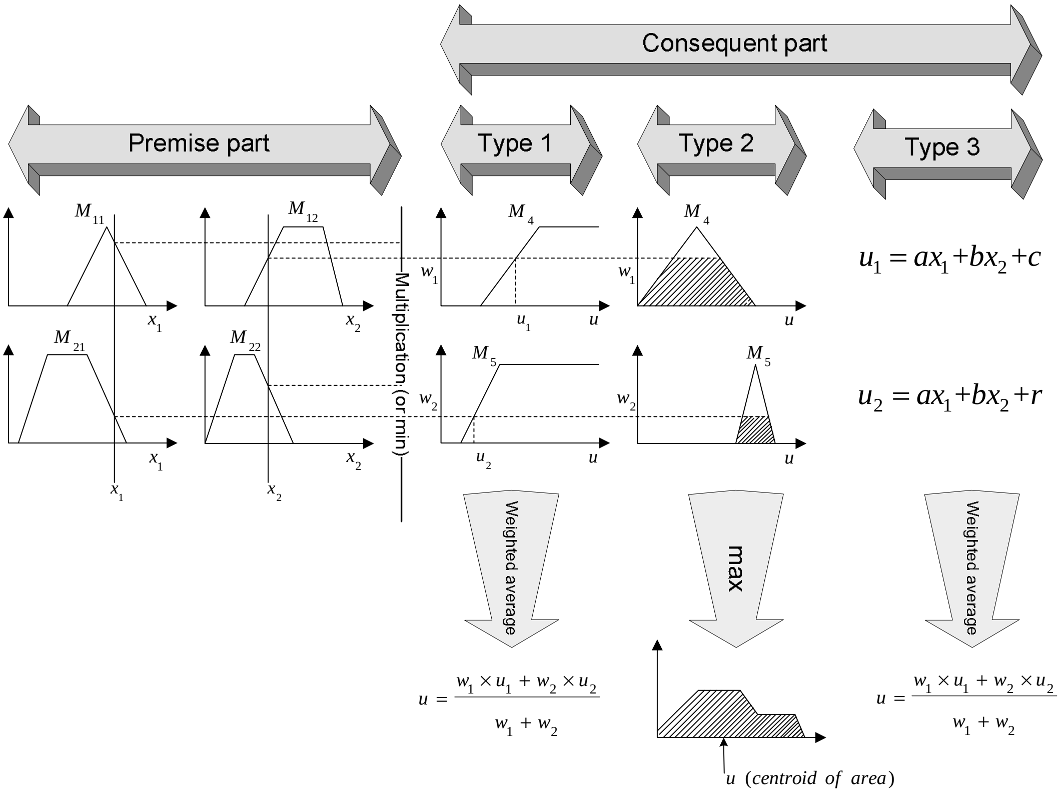


Figure 2.2: Commonly used fuzzy if-then rules and fuzzy mechanism.

analytical and/or physical models. On the other hand, nonlinear dynamical models for mechanical systems can be readily obtained by, for example, the Lagrange method and the Newton-Euler method. In such cases, the second approach, which derives a fuzzy model from given nonlinear dynamical models, is more appropriate. This tutorial focuses on second approach. This approach utilizes the idea of "sector nonlinearity", "local approximation," or a combination of them to construct fuzzy models.

2.1.1 Sector Nonlinearity

The idea of using sector nonlinearity in fuzzy model construction first appeared in [8]. Sector nonlinearity is based on the following idea. Consider a simple nonlinear system $\dot{x} = f(x(t))$, where $f(0) = 0$. The aim is to find the global sector such that $\dot{x} = f(x) \in [a_1 \; a_2]x(t)$. Figure 2.4a illustrates the sector nonlinearity approach. This approach guarantees an exact fuzzy model construction. However, it is sometimes difficult to find global sector for general nonlinear systems. In this case, we consider local sector nonlinearity. This is reasonable as variables of physical systems are always bounded. Figure 2.4b shows the local sector nonlinearity, where two lines become the local sectors under $-d < x(t) < d$. The fuzzy model exactly represents the *local* region, that is, $-d < x(t) < d$. The following two examples illustrate the concrete steps to construct fuzzy models.

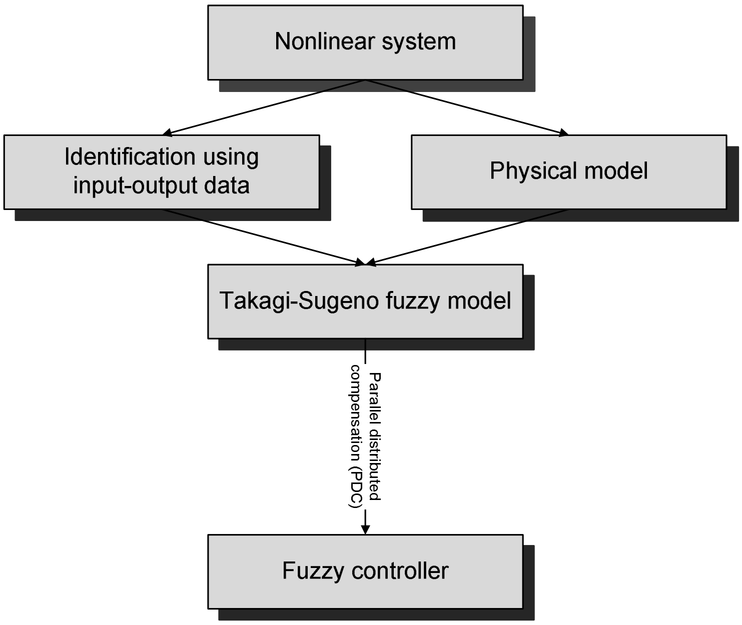


Figure 2.3: Model-based fuzzy control design.

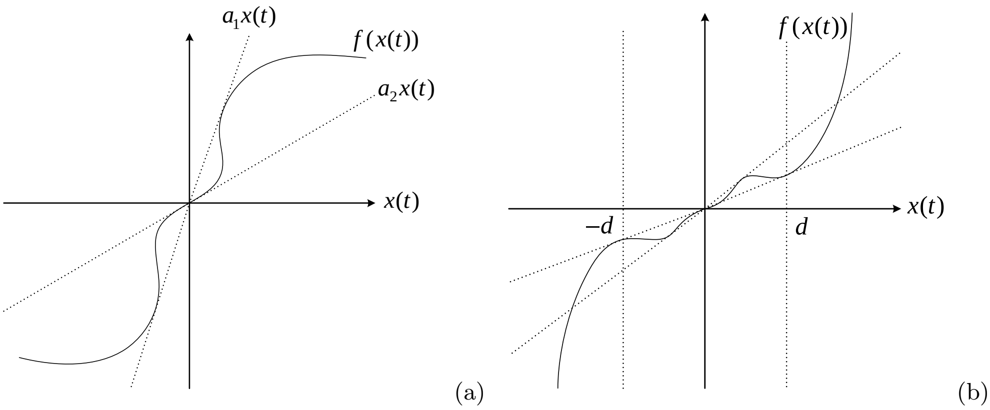


Figure 2.4: (a) Global sector nonlinearity. (b) Local sector nonlinearity.

2.2 Basic Fuzzy Mathematics for Modeling

Lets consider the nonlinear system below:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_1^2 + x_2^2 + u \end{cases} \tag{2.3}$$

The goal is to derive a T-S fuzzy model from the above given nonlinear system equations by the sector nonlinearity approach as if the response of the T-S fuzzy model in the specified domain exactly match with the response of the original system with the same input u.

The following steps should be taken to derive the T-S fuzzy model of (2.3). For simplicity, we assume that $x_1 \in [0.5, 3.5]$ and $x_2 \in [-1, 4]$. Here x_1 and x_2 are nonlinear terms in the equations in the last equations so we make them as our fuzzy variables. Generally they are denoted as z_1, z_2 and are known as premise variables that may be functions of state variables, input variables, external disturbances and/or time. Therefore

$z_1 = x_1$ and $z_2 = x_2$. Equation (2.3) can be written as

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ x_1 & x_2 \end{bmatrix} x(t),$$

where $x(t) = [x_1(t)\ x_2(t)]^T$. The first step for any kind of fuzzy modeling is to determine the fuzzy variables and fuzzy sets or so-called membership functions. Although there is no general procedure for this step and it can be done by various methods predominantly trial and error, in exact fuzzy modeling using sector nonlinearity, it is quite routine. It is assumed in this tutorial that the premise variables are just functions of the state variables for the sake of simplicity. This assumption is needed to avoid a complicated defuzzification process of the fuzzy controllers [9].

To acquire membership functions, we should calculate the minimum and maximum values of $z_1(t)$ and $z_2(t)$ which under $x_1 \in [0.5, 3.5]$ and $x_2 \in [-1, 4]$, they are obviously obtained as follows:

$$\max z_1(t) = 3.5, \qquad \min z_1(t) = 0.5,$$
$$\max z_2(t) = 4, \qquad \min z_2(t) = -1.$$

Therefore x_1 and x_2 can be represented by for membership functions M_1, M_2, N_1 and N_2 as follows:

$$z_1(t) = x_1(t) = M_1(z_1(t)) \cdot 3.5 + M_2(z_1(t)) \cdot 0.5,$$
$$z_2(t) = x_2(t) = N_1(z_2(t)) \cdot 4 + N_2(z_2(t)) \cdot (-1),$$

and because M_1, M_2, N_1 and N_2 are actually fuzzy sets according to fuzzy mathematics

$$M_1(z_1(t)) + M_2(z_1(t)) = 1,$$
$$N_1(z_2(t)) + N_2(z_2(t)) = 1.$$

We name the membership functions "Positive", "Negative," "Big," and "Small," respectively. Figure 2.5 shows these membership functions. Here, we can generalize that the i^{th}

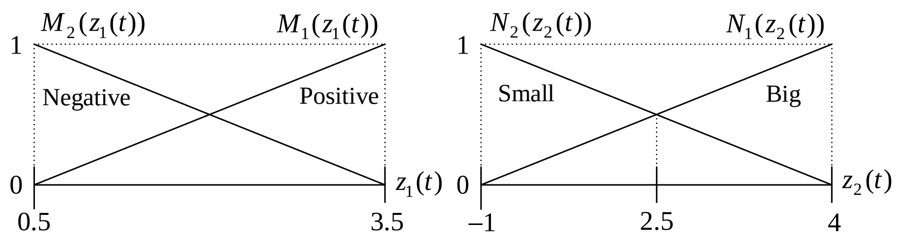


Figure 2.5: Membership functions $M_1(z_1(t))$, $M_2(z_2(t))$, $N_1(z_2(t))$ and $N_2(z_2(t))$.

rule of the continuous T-S fuzzy models are of the following forms:

Model Rule i:

IF $z_1(t)$ is M_{i1} and ... and $z_p(t)$ is M_{ip},

$$\text{THEN} \begin{cases} \dot{x} = A_i x(t) + B_i u(t), & i=1,2,\ldots,r; \\ y(t) = C_i x(t), & i=1,2,\ldots,r. \end{cases} \tag{2.4}$$

Here, M_{ij} is the fuzzy set and r is the number of model rules; $x(t)$ is the state vector, $u(t)$ is the input vector, $y(t)$ is the output vector, A_i is the square matrix with real elements and $z_1(t), \ldots, z_p(t)$ are known premise variables as mentioned before. Each linear consequent equation represented by $A_i x(t) + B_i u(t)$ is called a *subsystem*.

Therefore, the nonlinear system (2.3) is modeled by the following fuzzy rules (we don't consider input $u(t)$ in this stage):

Model Rule 1: IF $z_1(t)$ is "Positive" and $z_2(t)$ is "Big," THEN $\dot{x}(t) = A_1x(t)$.

Model Rule 2: IF $z_1(t)$ is "Positive" and $z_2(t)$ is "Small," THEN $\dot{x}(t) = A_2x(t)$.

Model Rule 3: IF $z_1(t)$ is "Negative" and $z_2(t)$ is "Big," THEN $\dot{x}(t) = A_3x(t)$.

Model Rule 4: IF $z_1(t)$ is "Negative" and $z_2(t)$ is "Small," THEN $\dot{x}(t) = A_4x(t)$.

where the subsystems are determined as:

$$A_1 = \begin{bmatrix} 0 & 1 \\ \max\limits_{z_1 \in \text{Positive}} z_1 & \max\limits_{z_2 \in \text{Big}} z_2 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 1 \\ \max\limits_{z_1 \in \text{Positive}} z_1 & \max\limits_{z_2 \in \text{Small}} z_2 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 0 & 1 \\ \max\limits_{z_1 \in \text{Negative}} z_1 & \max\limits_{z_2 \in \text{Big}} z_2 \end{bmatrix}, \quad A_4 = \begin{bmatrix} 0 & 1 \\ \max\limits_{z_1 \in \text{Negative}} z_1 & \max\limits_{z_2 \in \text{Small}} z_2 \end{bmatrix},$$

which is

$$A_1 = \begin{bmatrix} 0 & 1 \\ 3.5 & 4 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 1 \\ 3.5 & -1 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 0 & 1 \\ 0.5 & 4 \end{bmatrix}, \quad A_4 = \begin{bmatrix} 0 & 1 \\ 0.5 & -1 \end{bmatrix},$$

Now, $\dot{x}$ can be derived out of defuzzification process as:

$$\dot{x}(t) = h_1(z(t))A_1x(t) + h_2(z(t))A_2x(t) + h_3(z(t))A_3x(t) + h_4(z(t))A_4x(t)$$

where

$$\begin{aligned} h_1(z(t)) &= M_1(z_1(t)) \times N_1(z_2(t)), \\ h_2(z(t)) &= M_1(z_1(t)) \times N_2(z_2(t)), \\ h_3(z(t)) &= M_2(z_1(t)) \times N_1(z_2(t)), \\ h_4(z(t)) &= M_2(z_1(t)) \times N_2(z_2(t)). \end{aligned}$$

This T-S fuzzy model can exactly represents the nonlinear system in the region $[0.5, 3.5] \times [-1, 4]$ on the $x_1 - x_2$ space. To have a clear picture of the fuzzy modeling procedure above, we calculate the final output of $\dot{x}$ for the specific given values of x_1 and x_2.

According to the model rules above

Model Rule 1: IF $z_1(t)$ is "Positive" and $z_2(t)$ is "Big," THEN $\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = 3.5x_1 + 4x_2 \end{cases}$.

Model Rule 2: IF $z_1(t)$ is "Positive" and $z_2(t)$ is "Small," THEN $\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = 3.5x_1 - x_2 \end{cases}$.

Model Rule 3: IF $z_1(t)$ is "Negative" and $z_2(t)$ is "Big," THEN $\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = 0.5x_1 + 4x_2 \end{cases}$.

Model Rule 4: IF $z_1(t)$ is "Negative" and $z_2(t)$ is "Small," THEN $\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = 0.5x_1 - x_2 \end{cases}$.

Therefore, if $z_1 = x_1 = 2.75$ and $z_2 = x_2 = 0.25$, according to (2.6), the T-S fuzzy modeling implication can be derived as:

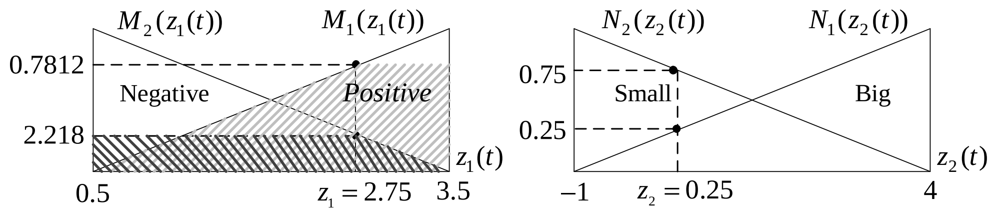


Figure 2.6: Given the value of $z_1 = 2.75$ and $z_2 = 0.25$ to the membership functions

Implication	Premise	Consequence	Truth value
Rule 1	$\mathrm{M}_1(z_1) = 0.7812, N_1(z_2) = 0.25$	$\begin{cases} \dot{x}_1 = 0.25 \\ \dot{x}_2 = 3.5 \times 2.75 + 4 \times 0.25 \end{cases}$	$0.7812 \wedge 0.25 = 0.25$
Rule 2	$\mathrm{M}_1(z_1) = 0.7812, N_2(z_2) = 0.75$	$\begin{cases} \dot{x}_1 = 0.25 \\ \dot{x}_2 = 3.5 \times 2.75 - 0.25 \end{cases}$	$0.7812 \wedge 0.75 = 0.75$
Rule 3	$\mathrm{M}_2(z_1) = 0.218, N_1(z_2) = 0.25$	$\begin{cases} \dot{x}_1 = 0.25 \\ \dot{x}_2 = 0.5 \times 2.75 + 4 \times 0.25 \end{cases}$	$0.218 \wedge 0.25 = 0.218$
Rule 4	$\mathrm{M}_2(z_1) = 0.218, N_2(z_2) = 0.75$	$\begin{cases} \dot{x}_1 = 0.25 \\ \dot{x}_2 = 0.5 \times 2.75 - 0.25 \end{cases}$	$0.218 \wedge 0.75 = 0.218$

Now, the final values for $\dot{x}_1$ and $\dot{x}_2$, in T-S fuzzy defuzzification process, can be calculated as:

$$\begin{cases} \dot{x}_1 = \dfrac{0.25 \times 0.25 + 0.25 \times 0.75 + 0.25 \times 0.218 + 0.25 \times 0.218}{0.25 + 0.75 + 0.218 + 0.218} = 0.25 \\ \dot{x}_2 = \dfrac{10.625 \times 0.25 + 9.375 \times 0.75 + 2.375 \times 0.218 + 1.125 \times 0.218}{0.25 + 0.75 + 0.218 + 0.218} = 7.2775 \end{cases}$$

Comparing, the values of $\dot{x}_1 = 0.25$ and $\dot{x}_2 = 7.6225$, we can see that the T-S fuzzy approximation does the good job and small value difference of $\dot{x}_2$, actually comes from rounding error of the premise fuzzy variables.

Generally, given a pair of $(x(t), u(t))$ for the model rule (3.6), the final outputs of the fuzzy model for the Continuous Fuzzy System (for the general rule of Discrete Fuzzy Systems see the appendix) are inferred as follows:

$$\dot{x} = \frac{\sum_{i=1}^{r} w_i(z(t))\{A_i x(t) + B_i u(t)\}}{\sum_{i=1}^{r} w_i(z(t))}$$

$$= \sum_{i=1}^{r} h_i(z(t))\{A_i x(t) + B_i u(t)\}, \tag{2.5}$$

$$y(t) = \frac{\sum_{i=1}^{r} w_i(z(t)) C_i x(t)}{\sum_{i=1}^{r} w_i(z(t))}$$

$$= \sum_{i=1}^{r} h_i(z(t)) C_i x(t). \tag{2.6}$$

where

$$z(t) = [z_1(t) z_2(t) \dots z_p(t)],$$

$$w_i(z(t)) = \prod_{j=1}^{p} M_{ij}(z_j(t)),$$

and weighting functions w_i should be normalized as

$$h_i(z(t)) = \frac{w_i(z(t))}{\sum_{i=1}^{r} w_i(z(t))} \tag{2.7}$$

for all t. The term $M_{ij}(z_j(t))$ is the grade of membership of $z_j(t)$ in M_{ij}. Since

$$\begin{cases} \sum_{i=1}^{r} w_i(z(t)) > 0, \\ w_i(z(t)) \geq 0, \qquad\qquad i\text{=1,2,}\ldots\text{,}r, \end{cases} \tag{2.8}$$

we have

$$\begin{cases} \sum_{i=1}^{r} h_i(z(t)) = 1, \\ h_i(z(t)) \geq 0, \qquad\qquad i\text{=1,2,}\ldots\text{,}r, \end{cases} \tag{2.9}$$

for all t.

Example 1 Consider the following nonlinear system:

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \begin{pmatrix} -x_1(t) + x_1(t)x_2^3(t) \\ -x_2(t) + (3 + x_2(t))x_1^3(t) \end{pmatrix} \tag{2.10}$$

For simplicity, we assume that $x_1 \in [-1, 1]$ and $x_2 \in [-1, 1]$. Of course, we can assume any range for $x_1(t)$ and $x_2(t)$ to construct a fuzzy model. Equation (2.10) can be written as

$$\dot{x}(t) = \begin{bmatrix} -1 & x_1(t)x_2^2(t) \\ (3 + x_2(t))x_1^2(t) & -1 \end{bmatrix} x(t),$$

where $x(t) = [x_1(t)\ x_2(t)]^T$ and $x_1(t)x_2^2(t)$ and $(3 + x_2(t))x_1^2(t)$ are nonlinear terms. For the nonlinear terms, define $z_1(t) \equiv x_1(t)x_2^2(t)$ and $z_2(t) \equiv (3+x_2(t))x_1^2(t)$. Then, we have

$$\dot{x}(t) = \begin{bmatrix} -1 & z_1(t) \\ z_2(t) & -1 \end{bmatrix} x(t).$$

Next, we should calculate the minimum and maximum values of $z_1(t)$ and $z_2(t)$ under $x_1(t) \in [-1, 1]$ and $x_2(t) \in [-1, 1]$. They are obtained as follows:

$$\max_{x_1(t),x_2(t)} z_1(t) = 1, \qquad \min_{x_1(t),x_2(t)} z_1(t) = -1,$$

$$\max_{x_1(t),x_2(t)} z_2(t) = 4, \qquad \min_{x_1(t),x_2(t)} z_2(t) = 0.$$

From the maximum and minimum values, $z_1(t)$ and $z_2(t)$ can be represented by

$$z_1(t) = x_1(t)x_2^2(t) = M_1(z_1(t)) \cdot 1 + M_2(z_1(t)) \cdot (-1),$$

$$z_2(t) = (3 + x_2(t))x_1^2(t) = N_1(z_2(t)) \cdot 4 + N_2(z_2(t)) \cdot 0,$$

where

$$M_1(z_1(t)) + M_2(z_1(t)) = 1,$$
$$N_1(z_2(t)) + N_2(z_2(t)) = 1.$$

Therefore the membership functions can be calculated as

$$M_1(z_1(t)) = \frac{z_1(t) + 1}{2}, \qquad M_2(z_1(t)) = \frac{1 - z_1(1)}{2},$$

$$N_1(z_2(t)) = \frac{z_2(t)}{4}, \qquad N_2(z_2(t)) = \frac{4 - z_2(t)}{4},$$

We name the membership functions "Positive," "Negative," "Big," and "Small," respectively. Then, the nonlinear system (2.10) is represented by the following fuzzy model.

Model Rule 1: IF $z_1(t)$ is "Positive" and $z_2(t)$ is "Big," THEN $\dot{x}(t) = A_1 x(t)$.

Model Rule 2: IF $z_1(t)$ is "Positive" and $z_2(t)$ is "Small," THEN $\dot{x}(t) = A_2 x(t)$.

Model Rule 3: IF $z_1(t)$ is "Negative" and $z_2(t)$ is "Big," THEN $\dot{x}(t) = A_3 x(t)$.

Model Rule 4: IF $z_1(t)$ is "Negative" and $z_2(t)$ is "Small," THEN $\dot{x}(t) = A_4 x(t)$.

Here,

$$A_1 = \begin{bmatrix} -1 & 1 \\ 4 & -1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} -1 & 1 \\ 0 & -1 \end{bmatrix},$$
$$A_3 = \begin{bmatrix} -1 & -1 \\ 4 & -1 \end{bmatrix}, \quad A_4 = \begin{bmatrix} -1 & -1 \\ 0 & -1 \end{bmatrix},$$

Figures 2.7a and 2.7b illustrates the above membership functions. The defuzzification is

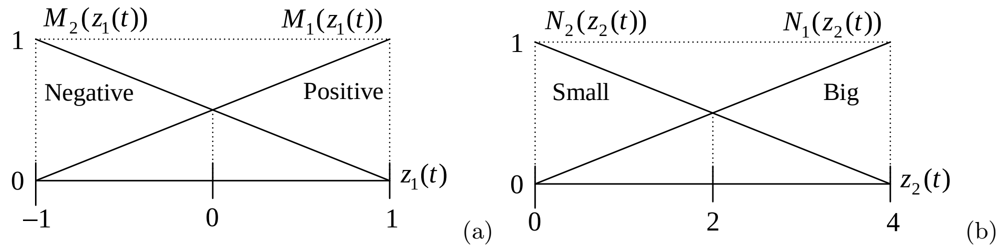


Figure 2.7: Membership functions $M_1(z_1(t))$, $M_2(z_1(t))$, $N_1(z_2(t))$ and $N_2(z_2(t))$.

carried out as

$$\dot{x}(t) = \sum_{i=1}^{4} h_i(z(t)) A_i x(t),$$

where

$$h_1(z(t)) = M_1(z_1(t)) \times N_1(z_2(t)),$$
$$h_2(z(t)) = M_1(z_1(t)) \times N_2(z_2(t)),$$
$$h_3(z(t)) = M_2(z_1(t)) \times N_1(z_2(t)),$$
$$h_4(z(t)) = M_2(z_1(t)) \times N_2(z_2(t)).$$

This fuzzy model exactly represents the nonlinear system in the region $[-1, 1] \times [-1, 1]$ on the $x_1 - x_2$ space.

Figure 2.8 shows the implementation of the above fuzzy model in Matlab/Simulink. As it is evident in Figure 2.9a, the time responses of the fuzzy model can exactly follow the responses of the original differential equations, which means the fuzzy model can exactly represents the original system in the pre-specified domains. It is also clear from Figure 2.9b that even outside of the boundaries of x_1 and x_2, the above approach can accurately represents the original system (2.10).

2.2.1 Local Approximation in Fuzzy Partition Spaces

Another approach to obtain T-S fuzzy models is the so-called local approximation in Fuzzy Partition Spaces. The spirit of the approach is to approximate nonlinear terms by judiciously chosen linear terms. This procedure leads to reduction of the number of model rules. For instance, if we try to exactly represents the inverted pendulum by T-S fuzzy model as the way in the previous section, it ends up with 16 rules. In comparison, using

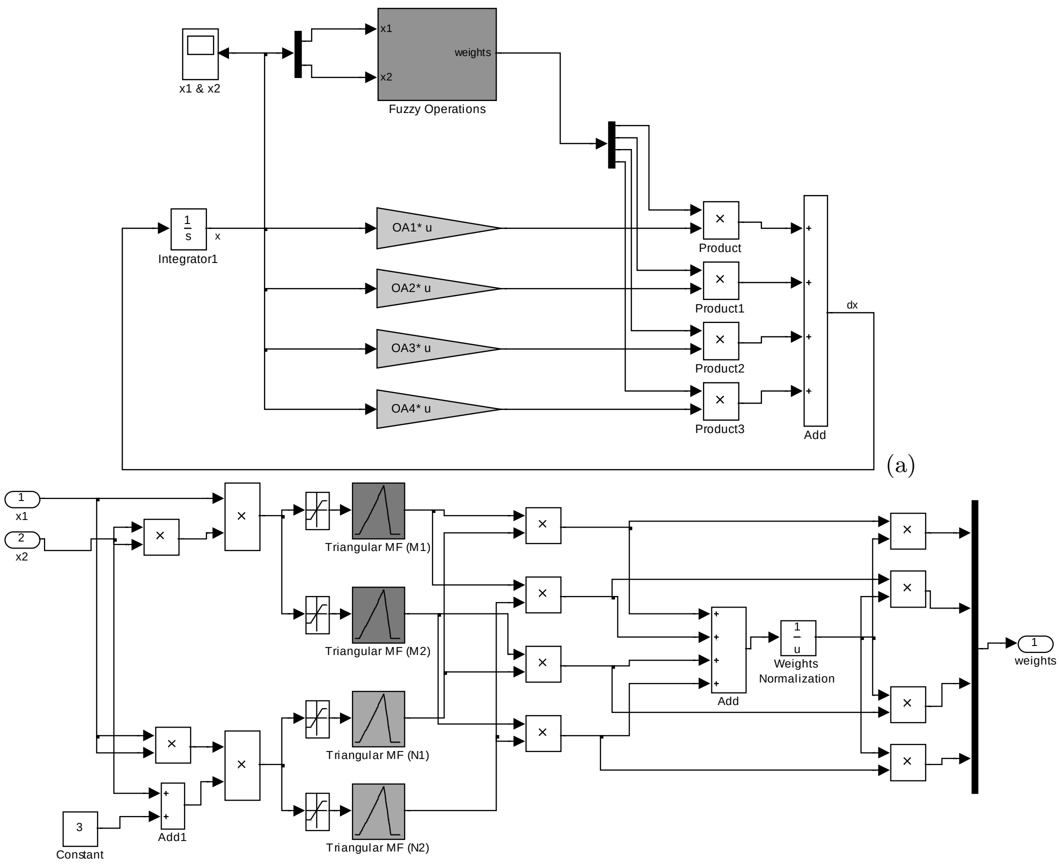


Figure 2.8: (a) Simulink implementation of the differential equations (2.10) modeled with T-S fuzzy. (b) fuzzy operations block.

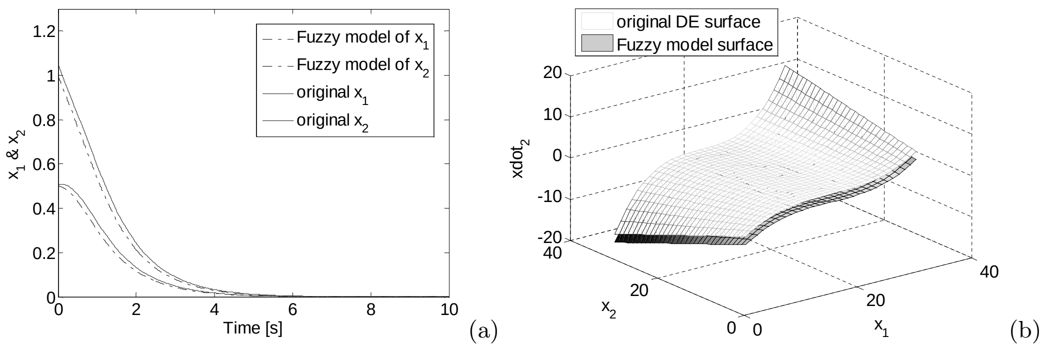


Figure 2.9: (a) Time response of the fuzzy model and the original system. (b) Function surface of $\dot{x}_2$, real and exact fuzzy modeling.

local approximation, a T-S model with 4 or 2 rules can be constructed. The number of model rules is directly related to the complexity of analysis and design LMI control laws for the T-S fuzzy controller. This is because the number of model rules for the overall T-S fuzzy control system is basically the combination of the model rules and control rules (for further reading on Model-based T-S fuzzy Controllers based on LMI design see [10, 11]).

Example 2 The equations of motion for the inverted pendulum [12] are

$$\begin{cases} \dot{x}_1(t) = x_2(t), \\ \dot{x}_2(t) = \frac{g\sin(x_1(t)) - amlx_2^2(t)sin(2x_1(t))/2 - a\cos(x_1(t))u(t)}{4l/3 - aml\cos^2(x_1(t))}, \end{cases} \tag{2.11}$$

where $x_1(t)$ denotes the angle (in radians) of the pendulum from the vertical and $x_2(t)$ is the angular velocity; $g = 9.8\ m/s^2$ is the gravity constant, m is the mass of the pendulum, M is the mass of the cart, $2l$ is the length of the pendulum, and u is the force applied to the cart (in newtons);$a = 1/(m + M)$.

When $x_1(t)$ is near zero, the nonlinear equations can be simplified as

$$\dot{x}_1(t) = x_2(t), \tag{2.12}$$

$$\dot{x}_2(t) = \frac{gx_1(t) - au(t)}{4l/3 - aml} \tag{2.13}$$

When $x_1(t)$ is near $\pm\pi/2$, the nonlinear equations can be simplified as

$$\dot{x}_1(t) = x_2(t), \tag{2.14}$$

$$\dot{x}_2(t) = \frac{2gx_1(t)/\pi - a\beta u(t)}{4l/3 - aml\beta^2}, \tag{2.15}$$

where $\beta = cos(88°)$ Just remind that (2.12)-(2.15) are now linear systems. We arrive at the following fuzzy model based on linear subsystems:

Model Rule 1: IF $x_1(t)$ is about 0 THEN $\dot{x}(t) = A_1x(t) + B_1u(t)$.

Model Rule 2: IF $x_1(t)$ is about $\pm\pi/2(|x_1| < \pi/2)$ THEN $\dot{x}(t) = A_2x(t) + B_2u(t)$.

Here,

$$A_1 = \begin{bmatrix} 0 & 1 \\ \frac{g}{4l/3 - aml} & 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 \\ -\frac{a}{4l/3 - aml} \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 0 & 1 \\ \frac{2g}{\pi(4l/3 - aml\beta^2)} & 0 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 \\ -\frac{a\beta}{4l/3 - aml\beta^2} \end{bmatrix},$$

and $\beta = cos(88°)$. Membership functions for *Rule 1* and *Rule 2* can be simply defined as shown in figure 2.10.

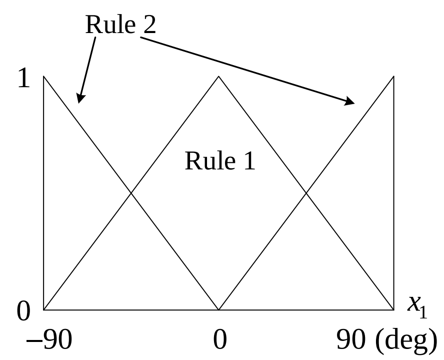


Figure 2.10: Membership functions of two-rule model.

An important and natural question arises in the construction using local approximation in fuzzy partition spaces or simplification before sector nonlinearity is that "Is it possible to approximate any smooth nonlinear systems with Takagi-Sugeno fuzzy models (3.1) having no consequent constant term?". The answer is fortunately *Yes* if we consider the problem in C^0 or C^1 context. That is, the original vector field plus its first-order derivative can be accurately approximated(for further reading please see [10]).

Now suppose the pendulum on the cart system is built in such a way that the work space of the pendulum is the full circle $[-\pi, \pi]$. In this subsection, we extends the result to the range of $x_1 \in [-\pi, \pi]$ except for a thin strip near $\pm\pi/2$. Balancing the pendulum for the angle range of $\pi/2 < |x_1| \leq \pi$ is referred to as a swing-up control of the pendulum. Recall that for $x_1 = \pm\pi/2$ the system is uncontrollable. We add two more rules (Rules 3 and 4) to the fuzzy model.

Model Rule 1: IF $x_1(t)$ is about 0 THEN $\dot{x}(t) = A_1x(t) + B_1u(t)$.

Model Rule 2: IF $x_1(t)$ is about $\pm\pi/2(|x_1| < \pi/2)$ THEN $\dot{x}(t) = A_2x(t) + B_2u(t)$.

Model Rule 3: IF $x_1(t)$ is about $\pm\pi/2(|x_1| > \pi/2)$ THEN $\dot{x}(t) = A_3x(t) + B_3u(t)$.

Model Rule 4: IF $x_1(t)$ is about π THEN $\dot{x}(t) = A_4x(t) + B_4u(t)$.

Here A_1, B_1, A_2 and B_2 are the same as above and

$$A_3 = \begin{bmatrix} 0 & 1 \\ \frac{2g}{\pi(4l/3 - aml\beta^2)} & 0 \end{bmatrix}, \quad B_3 = \begin{bmatrix} 0 \\ -\frac{a\beta}{4l/3 - aml\beta^2} \end{bmatrix},$$

$$A_4 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B_4 = \begin{bmatrix} 0 \\ -\frac{a}{4l/3 - aml} \end{bmatrix}.$$

The membership functions of this four-rule fuzzy model are shown in Figure 2.12. Figures

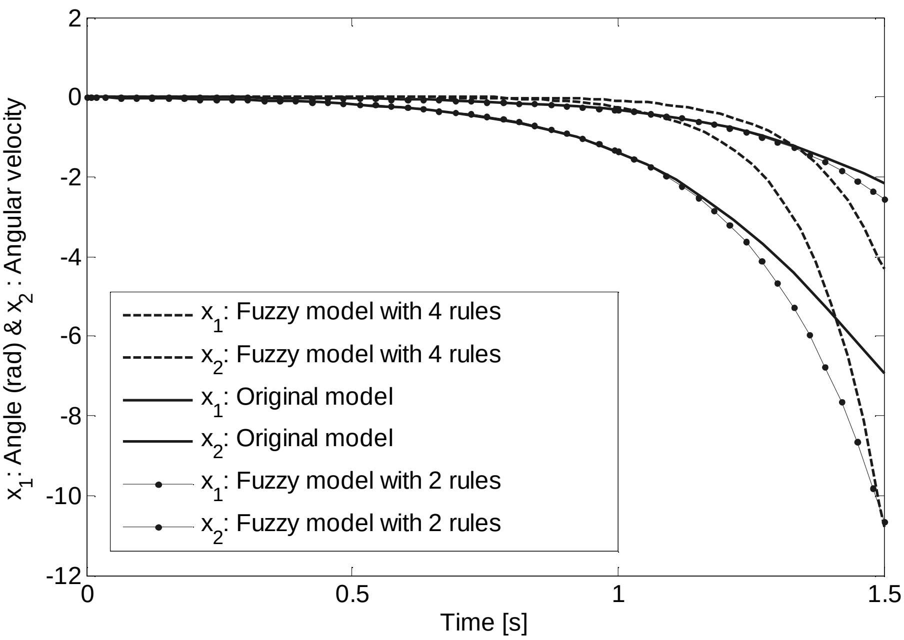


Figure 2.11: Time responses of original Inverted Pendulum system and their fuzzy approximations.

2.13 and 2.14 shows the construction of fuzzy models with 2 and 4 rules respectively Matlab/Simulink. Figure 2.11 compares the time response of the original system with

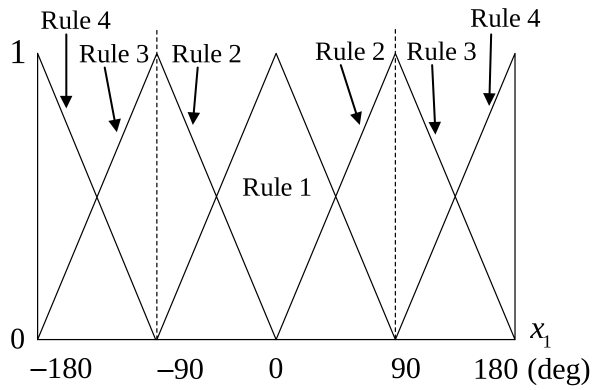


Figure 2.12: Membership functions of four-rule model.

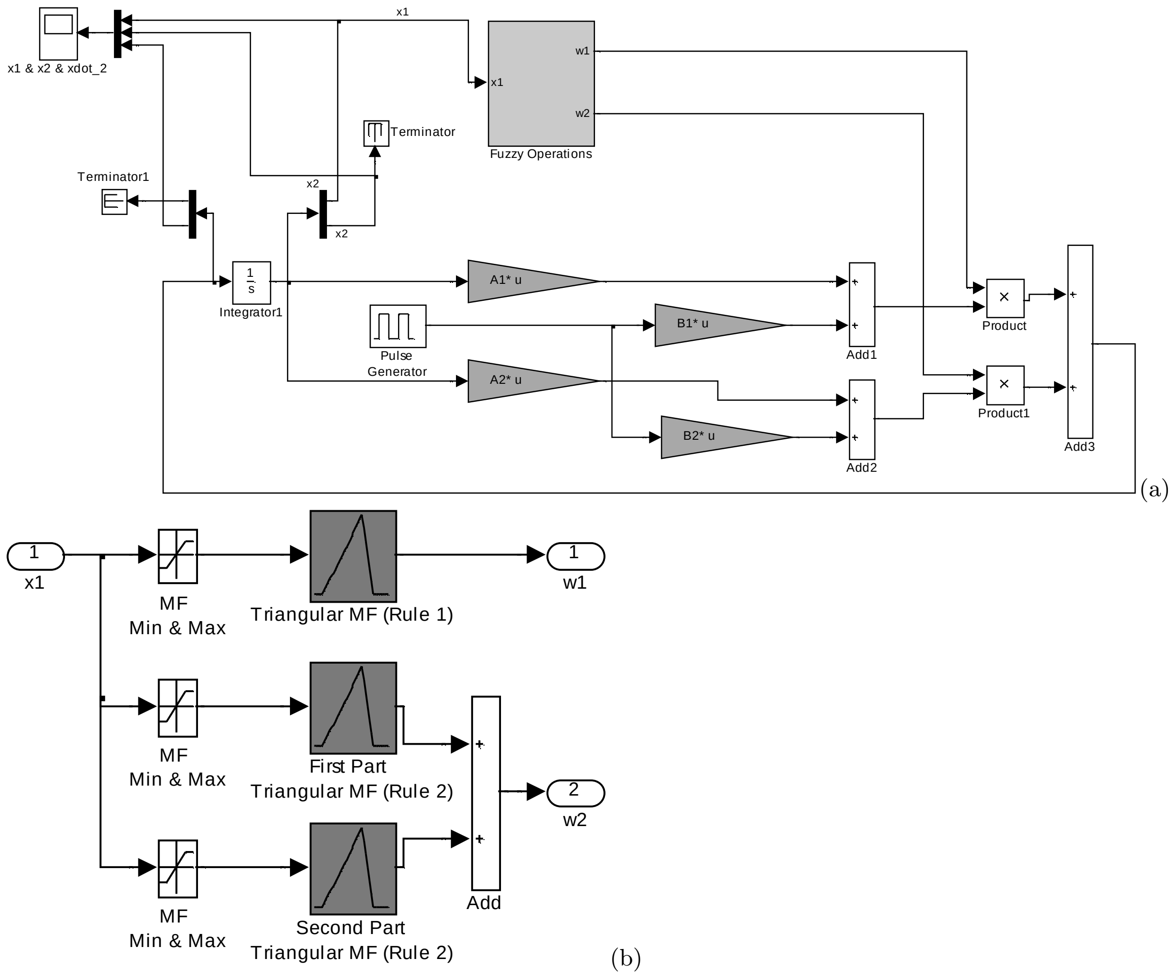


Figure 2.13: (a) Simulink implementation of Inverted Pendulum modeled with T-S Fuzzy with 2 rules (b) Fuzzy operation block.

fuzzy models. Figure 2.15 depicts the error between the real and fuzzy approximation. One can see that the 4-rule fuzzy model is a better approximation of real system comparing to 2-rule fuzzy model.

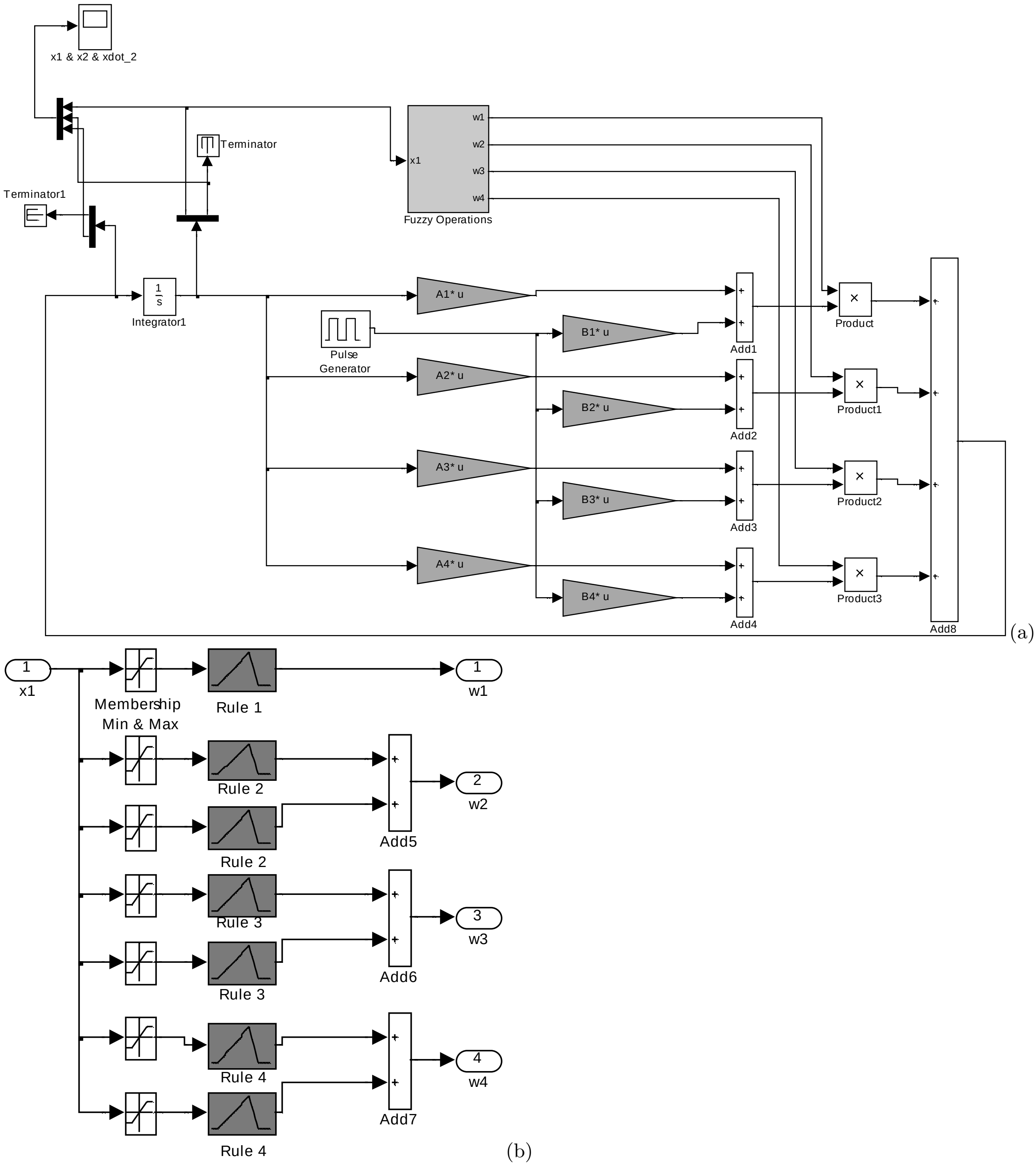
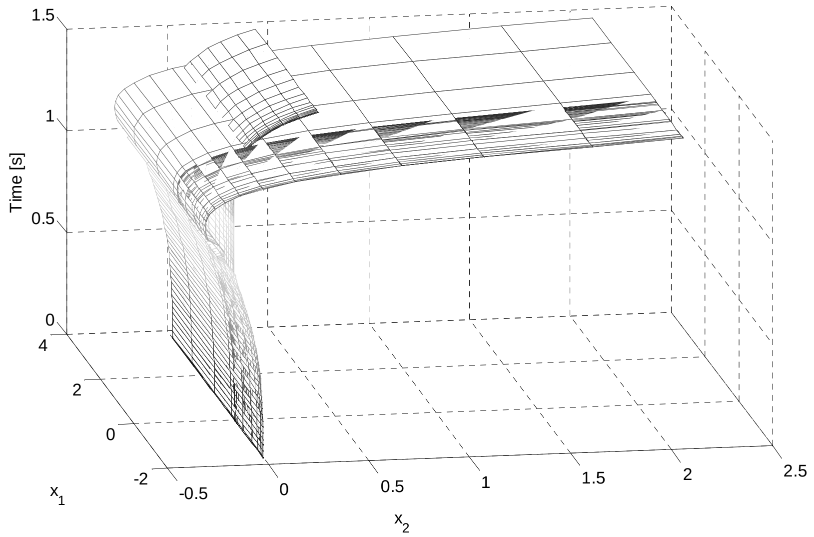
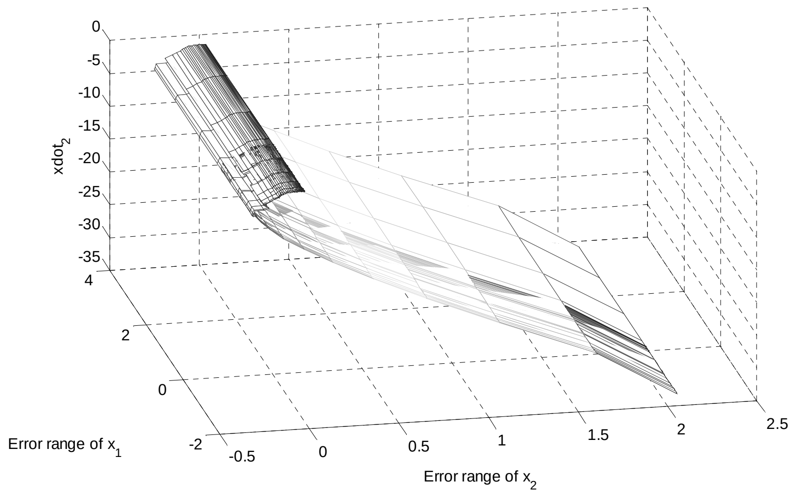


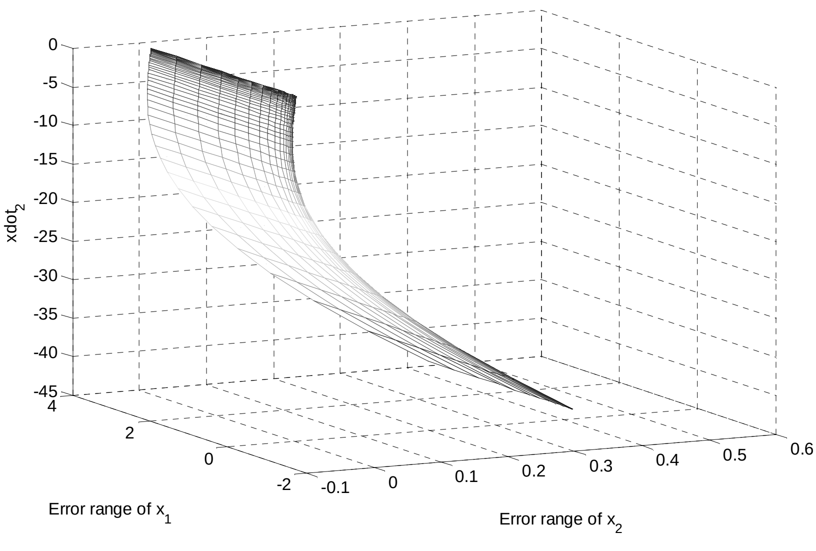
Figure 2.14: (a) Simulink implementation of Inverted Pendulum modeled with T-S Fuzzy with 4 rules. (b) Fuzzy operation block.



(a)



(b)



(c)

Figure 2.15: Error surface (a) with respect to time (b) of 2-rule fuzzy approximation with respect to $\dot{x}_2$ (c) of 4-rule fuzzy approximation with respect to $\dot{x}_2$.

Chapter 3

Appendix

Discrete Fuzzy System: DFS

Model Rule i:

IF $z_1(t)$ is M_{i1} and ... and $z_p(t)$ is M_{ip},

$$\text{THEN} \begin{cases} x(t+1) = A_i x(t) + B_i u(t), & i=1,2,\ldots,r; \\ y(t) = C_i x(t), & i=1,2,\ldots,r. \end{cases}$$

Here, M_{ij} is the fuzzy set and r is the number of model rules; $x(t)$ is the state vector, $u(t)$ is the input vector, $y(t)$ is the output vector, A_i is the square matrix with real elements and $z_1(t), \ldots, z_p(t)$ are known premise variables as mentioned before. Each linear consequent equation represented by $A_i x(t) + B_i u(t)$ is called a *subsystem*.

Given a pair of $(x(t), u(t))$, the final outputs of the fuzzy model are inferred as follows:

$$x(t+1) = \frac{\sum_{i=1}^{r} w_i(z(t))\{A_i x(t) + B_i u(t)\}}{\sum_{i=1}^{r} w_i(z(t))}$$

$$= \sum_{i=1}^{r} h_i(z(t))\{A_i x(t) + B_i u(t)\}, \tag{3.1}$$

$$y(t) = \frac{\sum_{i=1}^{r} w_i(z(t)) C_i x(t)}{\sum_{i=1}^{r} w_i(z(t))}$$

$$= \sum_{i=1}^{r} h_i(z(t)) C_i x(t). \tag{3.2}$$

where

$$z(t) = [z_1(t) z_2(t) \ldots z_p(t)],$$

$$w_i(z(t)) = \prod_{j=1}^{p} M_{ij}(z_j(t)),$$

$$h_i(z(t)) = \frac{w_i(z(t))}{\sum_{i=1}^{r} w_i(z(t))} \tag{3.3}$$

for all t. The term $M_{ij}(z_j(t))$ is the grade of membership of $z_j(t)$ in M_{ij}. Since

$$\begin{cases} \sum_{i=1}^{r} w_i(z(t)) > 0, & \\ w_i(z(t)) \geq 0, & i=1,2,\ldots,r, \end{cases} \tag{3.4}$$

we have

$$\begin{cases} \sum_{i=1}^{r} h_i(z(t)) = 1, & \\ h_i(z(t)) \geq 0, & \textit{i=1,2,\ldots,r,} \end{cases} \tag{3.5}$$

for all t.

Example Assume in the DFS that

$$p = n, z_1(t) = x(t),\ z_1(t) = x(t-1), \ldots,\ z_n(t) = x(t-n+1).$$

Then, the model rule can be represented as follows:

Model Rule i:

IF $x(t)$ is M_{i1} and ... and $x(t-n+1)$ is M_{in},

$$\text{THEN } \begin{cases} x(t+1) = A_i x(t) + B_i u(t), & \textit{i=1,2,\ldots,r;} \\ y(t) = C_i x(t), & \textit{i=1,2,\ldots,r.} \end{cases} \tag{3.6}$$

where $x(t) = [x(t)\ x(t-1) \cdots x(t-n+1)]^T$.

Bibliography

[1] Z. Binder, "About a multimodel control methodology, algorithm, multi-processors, implementation and application.," in *8th IFAC World Congress*, (Kyoto), pp. 981–986, 1991.

[2] T. TAKAGI and M. SUGENO, "Fuzzy identification of systems and its applications to modeling and control," *IEEE transactions on systems, man, and cybernetics*, vol. 15, no. 1, pp. 116–132, 1985.

[3] L. Zadeh, "Fuzzy sets," *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers*, 1996.

[4] C. Fantuzzi and R. Rovatti, "On the approximation capabilities of the homogeneous Takagi-Sugenomodel," in *Fuzzy Systems, 1996., Proceedings of the Fifth IEEE International Conference on*, vol. 2, 1996.

[5] J. Buckley, "Universal fuzzy controllers," *Automatica (Journal of IFAC)*, vol. 28, no. 6, pp. 1245–1248, 1992.

[6] M. Sugeno, *Fuzzy Control*. North-Holland, 1988.

[7] M. Sugeno and G. Kang, "Structure identification of fuzzy model," *Fuzzy Sets and Systems*, vol. 28, no. 1, pp. 15–33, 1988.

[8] S. Kawamoto, K. Tada, A. Ishigame, and T. Taniguchi, "An approach to stability analysis of second order fuzzy systems," in *Fuzzy Systems, 1992., IEEE International Conference on*, pp. 1427–1434, 1992.

[9] K. Tanaka, "A Theory of Advanced Fuzzy Control," 1994.

[10] H. Ying, *Fuzzy Control and Modeling: Analytical Foundations and Applications*. Wiley-IEEE Press, 2000.

[11] J. Li, H. Wang, D. Niemann, and K. Tanaka, "Dynamic parallel distributed compensation for Takagi–Sugeno fuzzy systems: An LMI approach," *Information Sciences*, vol. 123, no. 3-4, pp. 201–221, 2000.

[12] R. Cannon, *Dynamics of Physical Systems*. Dover Publications, 2003.