Takagi-Sugeno Approximation of Mamdani fuzzy systems and Applications

BY

Salem Boulos Bacha

THESIS

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science awarded
by DigiPen Institute of Technology
Redmond, Washington
United States of America

December

2015

Thesis Advisor: Barnabas Bede

DIGIPEN INSTITUTE OF TECHNOLOGY
GRADUATE STUDIES PROGRAM
DEFENSE OF THESIS

THE UNDERSIGNED VERIFY THAT THE FINAL ORAL DEFENSE OF THE MASTER OF SCIENCE THESIS TITLED

Takagi-Sugeno Approximation of Mamdani Fuzzy Systems and Applications

BY

Salem Boulos Bacha

HAS BEEN SUCCESSFULLY COMPLETED ON December 5, 2015.
MAJOR FIELD OF STUDY: COMPUTER SCIENCE.

APPROVED:

Dimitri Volper date Graduate Program Director	Xin Li date Dean of Faculty
Pushpak Karnick date Department Chair, Computer Science	Claude Comair date President

DIGIPEN INSTITUTE OF TECHNOLOGY
GRADUATE STUDIES PROGRAM
THESIS APPROVAL

DATE: December 5, 2015

BASED ON THE CANDIDATE'S SUCCESSFUL ORAL DEFENSE, IT IS RECOMMENDED THAT THE THESIS PREPARED BY

Salem Boulos Bacha

ENTITLED

Takagi-Sugeno Approximation of Mamdani Fuzzy Systems and Applications

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN COMPUTER SCIENCE
AT DIGIPEN INSTITUTE OF TECHNOLOGY.

Barnabas Bede date	Samir Abou Samra date	Dimitri Volper date
Thesis Committee Chair	Thesis Committee Member	Thesis Committee Member

Pushpak Karnick date	Matt Klassen date
Thesis Committee Member	Thesis Committee Member

To my Parents

ACKNOWLEDGMENTS

My thesis advisor Barnabas Bede, has been incredibly supportive throughout the creation of this thesis. I would like to thank him for being understanding and encouraging me to complete this thesis. I also wish to thank the committee who has been really supportive to me on completing this thesis. Finally, I want to also thank my friends and family for supporting me to reach where I am today.

Table of Contents

Abstract

In the present paper we propose, and investigate different types of constructive approaches towards approximating Mamdani fuzzy systems, using Takagi-Sugeno systems. The Takagi- Sugeno fuzzy systems that we consider will be of different types, such as using a piecewise linear approach, a polynomial approach, and an approach based on cubic splines. Since using a Mamdani system is computationally expensive, this paper will show how the approximation of Mamdani systems using Takagi-Sugeno will be less expensive on the computation side keeping high quality on the performance side. We extend these approaches to fuzzy rule bases with more antecedents. As application we construct a computing with words system using the proposed approach and also use the proposed approach control system in a video game.

Introduction

Approximating a function using fuzzy systems is the problem of identifying a fuzzy model by training it to fit a certain set of data points. Multiple approaches have been used to solve this problem including the ones in **[19], [10],** and **[11]**. Takagi-Sugeno rules and fuzzy inference methods proposed by authors such as **[10]** and **[2]** are utilized in function approximation problems using fuzzy logic. A small number of simple rules can be used to approximate such functions using the Takagi-Sugeno system. The issue that we encounter while using the TS rules is that the consequent part does not cover most of the function space and the rules are hard to interpret, while the Mamdani Fuzzy system **[18]** has a better perspective in presenting the rules and a better interpretability. If we consider a given function space, and if we conduct a good analysis on the function, we are able to generate rules that are specific to that space which will lead to a smooth approximation of the curve. However, this method is expensive in computation, thus this paper explains various ways to make a generalized system that takes information from a Mamdani fuzzy system and constructs a Takagi-Sugeno system that approximates a curve with arbitrary precision. The thesis originally started out as a proof of concept about approximating Mamdani fuzzy system using Takagi-Sugeno. We started testing out this concept by constructing high orders of Takagi-Sugeno in one dimension and two dimensions where by dimensions we mean number of inputs. First, we tested our proposed algorithms on predefined functions and then we extended our results to Mamdani fuzzy systems. For two dimensions' test cases we used two dimensional functions which also started giving us some analogy about the computational complexity of a fuzzy system. After fully testing this side of the concept we designed multiple fuzzy controllers to test on. We used Mamdani systems to solve a given problem system and then we approximated on that system using Takagi-Sugeno. Our results show believe this approach not only approximated the system very well but also changed the time complexity from non linear to linear time.

Chapter 2

Origins & Evolution of Fuzzy Logic

Fuzzy logic, fuzzy sets, and fuzzy systems is a branch of mathematics that deals with rules and reasoning under uncertainty. When something is fuzzy that means we are dealing with uncertain vague values, that we are responsible to defuzzify in order to get the right interpretation. Fuzzy sets allow partial membership with grade range between 0 and 1. The term fuzzy logic was first introduced in 1965 by Lotfi Zadeh **[9]** in the field of fuzzy set theory. Fuzzy logic now is being used to solve real world problems by using fuzzy systems to construct advanced artificial intelligence.

Fuzzy logic

Fuzzy sets are generalization of conventional (Boolean) logic that has been extended to handle the concept of partial truth. Truth values (in fuzzy logic) or membership values (in fuzzy sets) belong to the range [0, 1], with 0 being falseness and 1 being truth. It deals with real world vagueness. Some of the real world applications are ABS Brakes, Expert Systems, Control Units, Bullet train between Tokyo and Osaka, Video Cameras, and

Automatic Transmission. Fuzzy logic is also being used as game AI in order to control agents in the game. The agent using fuzzy logic will have the ability to simulate as if a human is making decisions with a certain threshold. In addition to game AI, fuzzy logic is being used to generate texture mapping on terrains where the fuzzy rules imply on what textures need to be used at a certain height or area on the map with a nice interpolation between the different areas **[13]**.

Crisp (Classical) Sets

Classical subsets as stated in **[1]** are defined by crisp predicates. Crisp predicates classify all elements into two groups or categories group 1 which is elements that make true the predicate and group 2 which is elements that make false the predicate.

Example:

$$\text{Let } E = \mathbb{Z}$$

$$A \subseteq E = \{n \subset E \,\big|\, n = 1 + 2k,{}' \, k \in \mathbb{Z}\}$$

$$\text{and our predicate } \; n \text{ is odd}$$

Crisp Characteristic Functions

The classification of elements can be done using an indicator or characteristic function:

$$\mathrm{A} \ : \mathrm{E} \ \longrightarrow \{0,1\}$$
$$\mathrm{A}\ (x) = \begin{cases} 1, & x \in \mathrm{A} \\ 0, & x \notin \mathrm{A} \end{cases}$$

- Note that:

$$A^{-1}\ (1) = \{\dots, -2\mathrm{k}-1, -3, -1, 1, 3, 2\mathrm{k}+1, \dots\}$$
$$A^{-1}\ (0) = \{\dots, -2\mathrm{k}, -4, -2, 0, 2, 4, 2\mathrm{k}, \dots\}$$

Fuzzy Sets

Human reasoning often uses vague predicates instead of classical ones. Elements cannot be always sharply classified into two groups! (Making a predicate either true or false)

Let us consider for example the set of tall men:

{1.6 m, 1.7m, 1.75m, 1.79m,1.8m ….}

Naturally raises the question: How do we model the concept of tall. As a descriptive term, tall is very subjective and relies on the context in which it is used. Even a 5 foot 7 man can be considered “tall” when he is surrounded by people shorter than he is.

Fuzzy Membership Functions

It is impossible to give a classical model or definition for the subset of tall men. However, we could establish to which degree a man can be considered tall. This can be done using a membership function with a continuum of values between 0 and 1.

$$\mathrm{A} : \mathrm{E} \longrightarrow [0,1]$$

- $\mathrm{A}(x) = y$ has the following interpretation
  - Individual x belongs to some extent ("y") to subset A
  - y is the degree to which the individual x is tall

- $\mathrm{A}(x) = 0$
  - Individual x does not belong to subset A

- $\mathrm{A}(x) = 1$
  - Individual x definitely belongs to subset A

A fuzzy set is defined by its membership function.

Types of Membership Functions

A membership function is a curve that is defined by mapping every input point in the input space to a membership value between 0 and 1. The input space is referred to as the universe of discourse. The only condition that a membership function has to satisfy to become a fuzzy set is that all the values must be between 0 and 1. The function itself can be an arbitrary curve where the user defines its shape depending on the point of view of the problem that we are modeling. In the present paper we use among others, the following membership functions.

- Gaussian

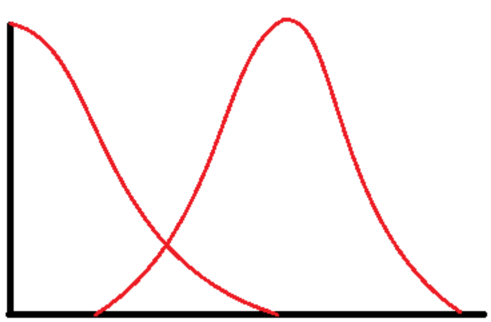


Figure 1.1 this graph shows a membership function of type Gaussian

- Triangular

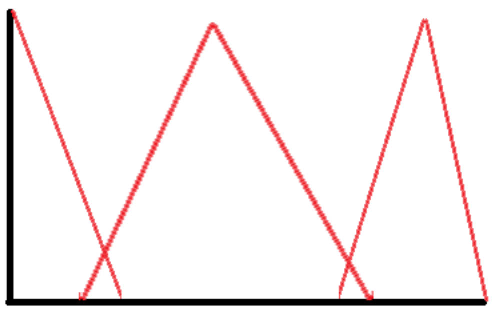


Figure 1.2 this graph shows a membership function of type Triangular

- Trapezoidal

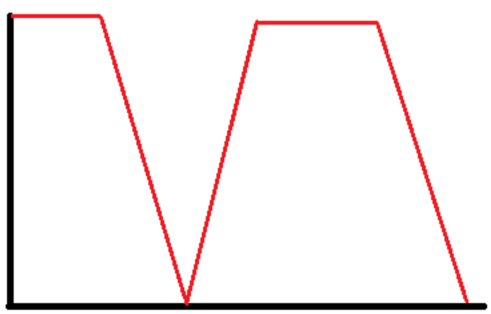


Figure 1.3 this graph shows a membership function of type Trapezoidal

Membership functions represent distributions of possibility rather than probability. For instance, the fuzzy set Young expresses the possibility that a given individual is young. Membership functions often overlap with each other. A given individual may belong to different fuzzy sets (with different degrees). For practical reasons, in many cases the universe of discourse (E) is assumed to be discrete

If we don't have a discrete universe of discourse, we may have a discrete approximation

$$E = \{x_1, x_2, \ldots, x_n\}$$

- The pair $(x, A(x))$, denoted by $A(x)/x$ is called fuzzy singleton
- Fuzzy sets can be described in terms of fuzzy singletons

$$M = \{(A(x)/x)\} = \cup_{i=1}^{n} A(x_i)/x_i$$

Basic Definition over Fuzzy Sets

Empty set: A fuzzy subset A $\subseteq E$ is empty (denoted A=∅) iff

$$A(x) = 0 \,, \forall x \in E$$

Equality: two fuzzy subsets A and B defined over E are equivalent iff

$$A(x) = B(x), \forall x \in E$$

A fuzzy subset A $\subseteq E$ is contained in B $\subseteq E$ iff

$$A(x) \leq B(x), \forall x \in E$$

Normality: A fuzzy subset A $\subseteq E$ is said to be normal iff

$$\max_{x \in E} A\,(x) = 1$$

Support: The support of a fuzzy subset A $\subseteq E$is a crisp set defined as follows

$$S_A = \{x \in E \mid A(x) > 0\} \;\; \emptyset \subseteq S_A \subseteq E$$

Operations over Fuzzy Sets

The basic operations over crisp sets can be extended to suit fuzzy sets

Standard operations:

- Intersection:
  $A \cap B(x) = \min(A(x), B(x))$

- Union:
  $A \cup B(x) = \max(A(x), B(x))$

- Complement:
  $A(x) = 1 - A(x)$

Fuzzy Intersection (t-norms)

A t-norm is an operation on [0,1] that satisfies the following axioms :

$T(x, y) = T(y, x) \quad \forall x, y \in E$ commutativity

$T(T(x, y), z) = T(x, T(y, z)) \ \forall x, y, z \in E$ associativity

$(x \leq y), (w \leq z) \rightarrow T(x, w) \leq T(y, z) \quad \forall x, y, w, z \in E$ monotony

$T(x,0) = 0 \quad \forall x \in E$ absorption

$T(x,1) = x \quad \forall x \in E$ neutrality

Given two fuzzy sets A, B ⊆ E, their intersection can be defined

based on a t-norm as follows:

$A \cap B(x) = T[A(x), B(y)] \ \forall x, y \in B$

Fuzzy Union (t-conorms)

A t-conorm is an operation on [0,1] that satisfies the following axioms :

$S(x,y) = S(y,x) \quad \forall x, y \in E$ commutativity

$S(S(x,y),z) = S(x,S(y,z)) \quad \forall x, y, z \in E$ associativity

$(x \leq y), (w \leq z) \rightarrow S(x,w) \leq S(y,z) \quad \forall x, y, w, z \in E$
monotony

$S(x,1) = 1 \quad \forall x \in E$ absorption

$S(x,0) = x \quad \forall x \in E$ neutrality

Given two fuzzy sets A, B ⊆ E, their union can be defined based on a t-conorm as follows:

$$A \cup B(x) = S[A(x), B(y)] \quad \forall x,\ y \in E$$

Linguistic Variables

Linguistic variables are used to associate a term from natural language to a fuzzy set, used as a mathematical model for the given linguistic expression **[1]**:

(X, T, U, G, M)

Where

X is the name of the variable

T is the set of linguistic terms which can be values of the variable

U is the universe of discourse

G is a collection of syntax rules, grammar, that produces correct expressions in T.

M is a set of semantic rules that map T into fuzzy sets in U.

Fuzzy Inference

We can see from the previous definition and as defined **[1]** that a linguistic variable works as a dictionary that translates linguistic terms into fuzzy sets. Often we use the term linguistic variable for a given value of the linguistic variable and also, if confusion is avoided, we use the same term for the fuzzy set that is associated to it, i.e., if A is a fuzzy set that is associated through a semantic rule to an instance of a linguistic variable, then we say that A is a linguistic variable.

Fuzzy rules

Fuzzy rules (fuzzy if-then rules) as explained in **[1]** and **[4]** are able to model expert opinion or commonsense knowledge often expressed in linguistic terms. The intuitive association that exists between given typical input data and typical output data is hard to be described in a mathematically correct way, because of the uncertain, often subjective nature of this information. Fuzzy rules are tools that are able to model and use such knowledge.

A fuzzy rule is a triplet (A, B, R) that consists of an antecedent A $\in \mathcal{F}(X)$, a consequence B $\in \mathcal{F}(X)$, that are linguistic variables, linked through a fuzzy relation R $\in \mathcal{F}(X \text{x} Y)$ **[1]**.

Using fuzzy sets, a fuzzy rule is written as follows:

If x is A then y is B

Definition: (Mamdani Assilian **[4]**) We define the fuzzy rule

If x is A then y is B

As a fuzzy relation as follows:

Mamdani rule: $R_m(x, y) = \mathrm{A}(x) \wedge \mathrm{B}(y)$

Remark: In many applications a fuzzy rule will have several antecedents that are used in conjunction to build our fuzzy rule. For example, a more complex fuzzy rule can be considered

If x is A and y is B then z is C

In this case the antecedents are naturally combined into a fuzzy relation

$\mathrm{D}\ (x, y) = \mathrm{A}(x) \wedge \mathrm{B}(y)$, that is regarded as a fuzzy set on its own. Then the fuzzy rule uses the antecedent D. For example, in this case the Mamdani rule will be

$$R_m\ (x, y, z) = \mathrm{A}(x) \wedge \mathrm{B}(y) \wedge \mathrm{C}(z).$$

Definition: (Mamdani-Assilian) We define the fuzzy rule base

$$\text{If } x \text{ is } \mathrm{A}_i \text{ then } y \text{ is } B_i,\ i = 1, \ldots, \mathrm{n}$$

as a fuzzy relation as follows:

Mamdani rule base:

$$R_m\ (x, y) = \bigvee_{i=1}^{n} A_i(x) \wedge B_i(y)$$

Fuzzy Inference

Fuzzy inference is the process of obtaining a conclusion for a given input that was possibly never encountered before. The basic rule (law) for a fuzzy inference system is the compositional rule of inference (Zadeh). It is based on the fuzzy version of the classical rule of Modus Ponens. Let us recall first the classical Modus Ponens of Boolean logic:

premise: if p then q

fact: p

conclusion: q

Given a fuzzy rule or a fuzzy rule base R ∈ $\mathcal{F}(X \text{ x } Y)$, the compositional rule of inference is a function $\mathcal{F}: \mathcal{F}(X) \longrightarrow \mathcal{F}(Y)$ determined through a composition B' = $\mathcal{F}$(A')= A' * R, with * : $\mathcal{F}(x)$ x $\mathcal{F}(X \text{ x } Y) \rightarrow \mathcal{F}(Y)$ being a composition of fuzzy relations.

The composition rule of inference consists of a

premise: if x is A_i then y is B_i, i = 1, …, n

fact: x is A'

conclusion: y is B'

Definition: (Mamdani- Assilian **[4]**)

We define a fuzzy inference based on a composition law as follows:

Mamdani Inference:

$$B'(\text{y}) = A' \circ R(x, y) = \bigvee_{x \epsilon X} A'(x) \wedge R(x, y)$$

where $R(x, y) = \bigvee_{i=1}^{n} A_i(x) \wedge B_i(y)$

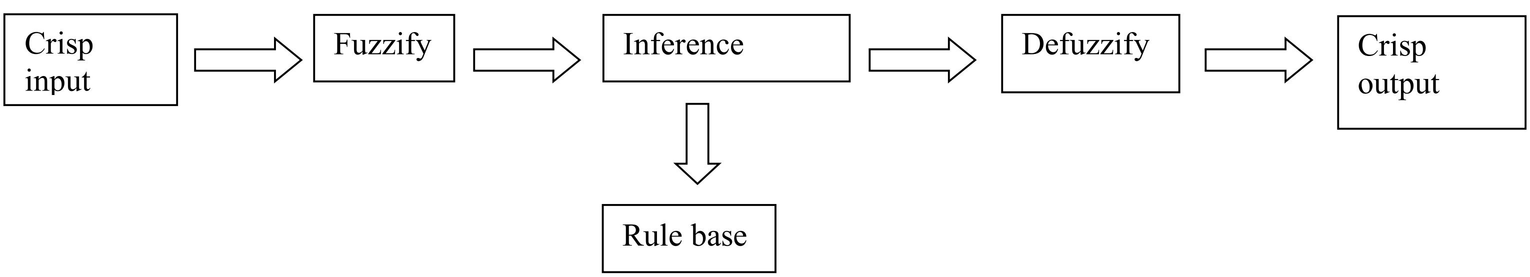
Chapter 3

Structure of a SISO Fuzzy System

A single input single output fuzzy system (SISO Fuzzy System) uses a crip input, fuzzifies it, maps it through a fuzzy inference system and the fuzzy output that is obtained and defuzzified to get a crisp output. Often SISO Fuzzy Systems are used in a control problem in which case they are called fuzzy controllers **[1]**.

The diagram of a SISO fuzzy is represented bellow:

[The components of a fuzzy controller]



We will discuss in what follows each of the above components.

Fuzzification

Most of the systems use the most basic fuzzifier that is the canonical inclusion. If $x_0 \in X$ is a crisp input then the fuzzy set associated with it is the singleton fuzzy set x_0, given by the characteristic function

$$A'(x) = \mathrm{X}_{\{x_0\}}(x) = \begin{cases} 1 & if \quad x = x_0 \\ 0 & if \quad x \neq x_0 \end{cases}$$

Fuzzy Rule Base

The fuzzy rule base can be described as a fuzzy relation :

$$R(x, y) = \bigvee_{i=1}^{n} A_i(x) \wedge B_i(y)$$

Consider a linguistic variable A with linguistic values close, distant, far and another linguistic variable B with linguistic values slow, steady, fast and a consequence linguistic variable C with linguistic values slowdown, keep distance, brake.

Then we can write rules using the relation we stated above:

Example:

If A is close and B is slow then C is brake.

Using the following example, we can populate a fuzzy rule base with the antecedents linguistic variables and the consequence linguistic variables.

Fuzzy Inference Mamdani

The fuzzy inference system that we consider can be of any type that have been discussed before. Mamdani Inference gives

$$B'(\mathrm{y}) = (\mathrm{R} \circ A')\,(x) = \bigvee_{x \epsilon X} A'\,(x) \wedge R(x,\ y)$$

where the fuzzy relation R is the fuzzy rule base.

Defuzzification

Defuzzification as explained in **[1]** is the final step in a fuzzy control algorithm. Based on the output of a fuzzy controller one has to give an estimate of the crisp quantity (a representation crisp element) for the output value of the SISO fuzzy system. In this case one has to use a defuzzification. There are many different defuzzification methods and based on the given application that we are working on, we can select a suitable defuzzification. In the present paper we will use the center of gravity as our defuzzification method. Center of Gravity (COG). The value selected is the center of gravity of the fuzzy set

u ∈ $\mathcal{F}(X)$. The defuzzification value is the x coordinate of the center mass of the fuzzy set.

$$\text{COG}\ (u) = \frac{\int_W x \,.\, u\,(x)dx}{\int_W u\,(x)dx}\ ,$$

Where W = supp(u).

Fuzzy Inference and Rule Base for a SISO Fuzzy System

Mamdani inference has simple expression on par with great computational and intuitive properties. These were historically the systems used in the first fuzzy controllers. Also we can obtain a simplified expression for the output of the fuzzy controller. If $A'(x) = \chi_{\{x_0\}}(x)$ is a crisp input of a fuzzy inference system with a given rule base R(x, y), then the output of a Mamdani fuzzy system is $R'(x, y) = \bigvee_{i=1}^{n} A_i(x) \wedge B_i(y)$

Indeed, if we consider

$$A'(x) = \begin{cases} 1 \ \ if\ x = x_0 \\ 0\ if\ x \neq x_0 \end{cases}$$

then for Mamdani inference we have

$$B'(y) = \bigvee_{x \epsilon X} A'(x) \wedge R(x,\ y) = A'(x_0) \wedge R(x_0, y)$$

Takagi-Sugeno Fuzzy Systems

Takagi-Sugeno fuzzy systems **[1][5] [3]** and **[8]** are intrinsically single input single output systems. A Takagi-Sugeno system has the rule base consisting of antecedents of linguistic type and conclusions that are piecewise linear crisp outputs. This makes the defuzzification step redundant.

$$\text{premise: if } x \text{ is } A_i \text{ then } z_i = a_i x + b_i,\ i = 1, \ldots, \text{n}$$
$$\text{fact: } x \text{ is } x_0$$
$$\text{conclusion: } z \text{ is } \text{z}_0$$

Takagi-Sugeno (TS) fuzzy controllers do not use an inference system as the Mamdani system described in the previous section, instead they use the firing strength of each fuzzy rule in the computation of the conclusions. TS fuzzy controller has crisp inputs, singleton fuzzifier and practically it does not have a defuzzifier. The fuzzy rules for TS fuzzy system are of the form if x is A_i and y is B_i then $z_i = a_i x + b_i y + c_i$. The output is computed as

$$\mathrm{TS}(x,y) = \frac{\sum_{i=1}^{n} A_i(x).B_i(y).(a_i x + b_i y + c_i)}{\sum_{i=1}^{n} A_i(x).B_i(y)}$$

The control algorithm for a TS fuzzy system of this type is given as follows:

Algorithm:

1. Input the crisp values $x_0,\ y_0$.

2. Calculate the firing strengths of each fuzzy rule

$$\delta_i = A_i(x_0) \wedge B_i\ (y_0)$$

3. Calculate the rule outputs

$$z_i = a_i x_0 + b_i y_0 + c_i$$

4. The output is

$$z_0 = \frac{\sum_{i=1}^{n} \delta_i\ .\, z_i}{\sum_{i=1}^{n} \delta_i}$$

Remark. Designing the rule base for a TS fuzzy system requires knowledge of the parameters $a_{i,}, b_i, c_i$, i = 1, …,n. The values of the parameters $a_{i,}, b_i,$ and c_i can be given in advance or they can be obtained using adaptive techniques that will be discussed later.

In the present paper we propose a simple method to calculate parameters $a_{i,}, b_i,$ and c_i from a Mamdani fuzzy system. Obtaining this way, a Takagi-Sugeno approximation of a fuzzy system **[7]**.

Chapter 4

How to Approximate an Arbitrary Function using a Mamdani Fuzzy System?

We consider the problem of approximation of a function $y = f(x)$ using a Mamdani fuzzy system.

Step 1:

We consider sample values from the function.

Let $f(x_i) = y_i$, $i = 0$, … , $n + 1$.

Step 2:

We construct fuzzy sets A_i & B_i which will represent the antecedents and consequences respectively.

To construct A_i & B_i we use membership functions such as triangle, trapezoid, Splines, etc..

The antecedent is constructed using a value range

t_j $j = 1, \dots, m$ with a given step size and sample points $x_i, i = 1, \dots, n$ as a support for each fuzzy set.

For example : $A_i = (t, x_{i-1,} x_i, x_{i+1})$.

The fuzzy sets of the consequence parts are considered integrable and are constructed such as the value range is $l = \min(y), \dots, \max(y)$ with a predefined step size. The support of each fuzzy set is considered to be based on the sample points $y_i, i = 1, \dots, n$. For example:

$B_i = (\ l\ , \min\{y_{i-1}\ ,\ y_i,\ y_{i+1}\}, y_i\ , \max\{y_{i-1}\ ,\ y_i,\ y_{i+1}\}\ )$

So our fuzzy rule base in this case becomes

if x is A_i then $y\ is\ B_i$, i= 1, …, n

Which is consistent with our intuitive knowledge on function approximation.

Step 3:

The approximation part of the function is done here where we combine the SISO fuzzy system with a Mamdani fuzzy rule base and COG defuzzification.

Let us recall that the fuzzy output of the Mamdani system is calculated as follows:

$B'(y) = (R \circ A')(x) = \bigvee_{x \epsilon X} A'(x) \wedge R(x,\ y) = \bigvee_{i=1}^{n} A_i(x) \wedge B_i(y)$

and the COG as:

$$\text{COG}(\text{B}') = \frac{\int_c^d \text{B}'(y)\,.y.\ dy}{\int_c^d \text{B}'(y)\ \ dy}$$

Combining both relations we can write a SISO fuzzy system as:

$$\text{F}(\text{f}, \text{x}) = \frac{\int_c^d \vee_{i=1}^{n}(\,A_i\ (x) \wedge\ B_i\ (y)\,).y.dy}{\int_c^d \vee_{i=1}^{n}(\,A_i\ (x) \wedge\ B_i\ (y)\,).dy}$$

Since A_i is continuous and B_i is integrable then we can approximate the given integrals as a summation. (Riemann Sum)

The pseudo code is as follows:

```
for k = 1 : t
  top  =0;
  bottom =0;
  for j = 1:  l
     temp =0;
     for i = 2 : n – 1
         tmp1 = tnorm( A(i,k), B(i,j)  );
         temp =max( [tmp1, mtmp] );
     end
     top  = top  + temp * l(j);
     bottom = bottom + temp;
  end
  out(k)= top   / bottom;
end
```

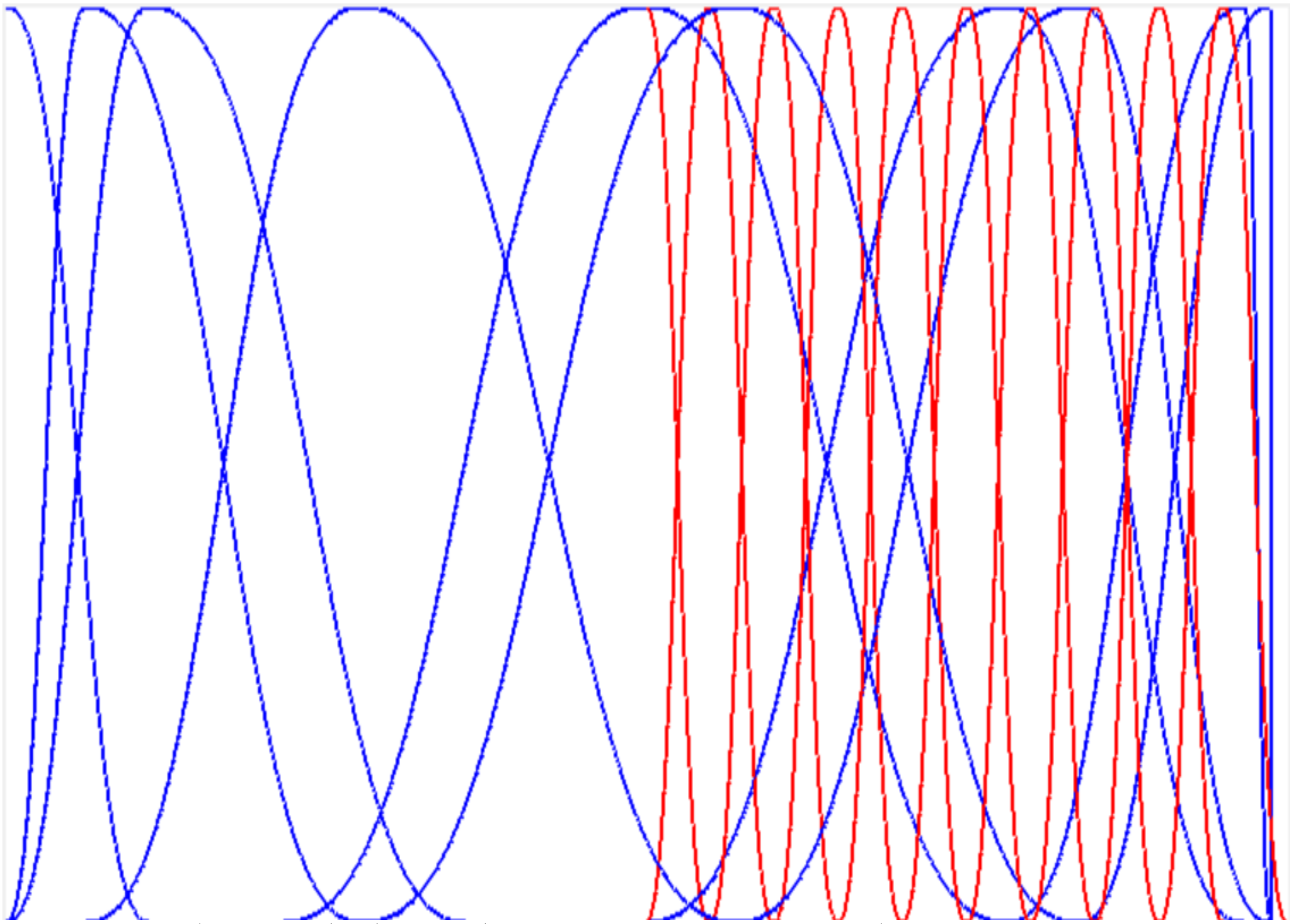


Figure 2.0 This graph shows the Mamdani fuzzy sets that were computed to approximate the function curve. The red sets represent the antecedent part and the blue sets represent the consequence part.

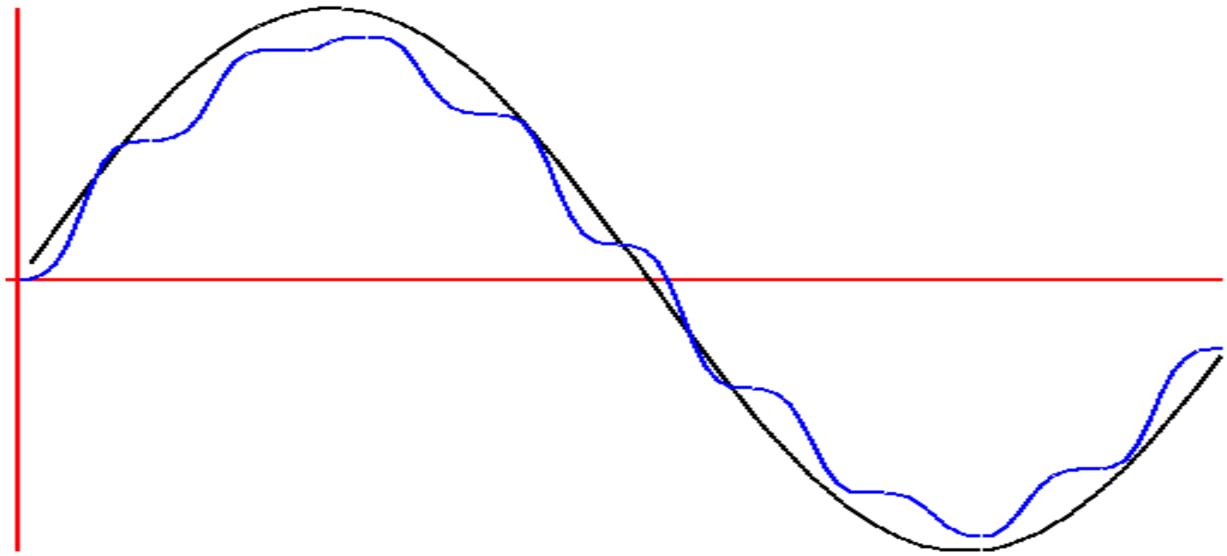


Figure 2.1 This graph shows the approximation of a curve using Mamdani SISO system with 11 fuzzy rule set. The black line is the original function and the blue line is the approximation.

The time complexity to approximate the function using a Mamdani fuzzy system is $O(n.l)$ where n is the number of rules and l is the number of steps of integration with a certain precision.

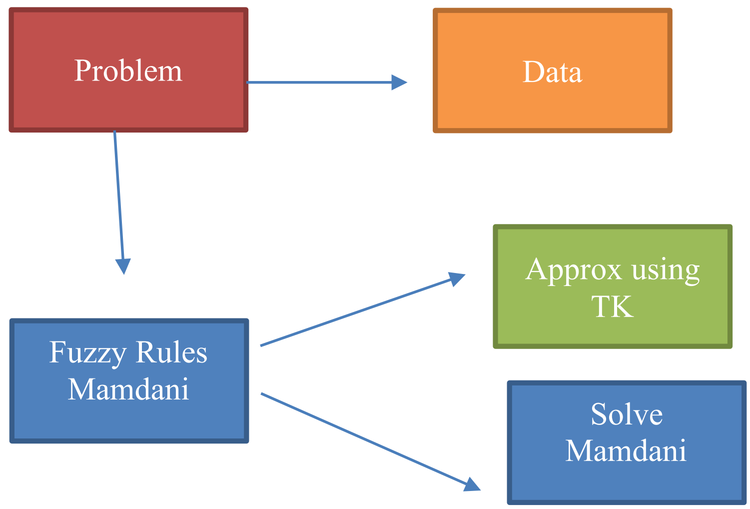
How to Approximate a Mamdani Fuzzy System using TS Controller?

In this section we will construct the TS fuzzy system that approximates a Mamdani fuzzy system. Since Takagi-Sugeno fuzzy systems can approximate any continuous function with random accuracy, **[3]**, the existence of such Takagi-Sugeno fuzzy systems is theoretically ensured. Given a Mamdani system, we can construct a Takagi-Sugeno fuzzy system that approximates its output with any arbitrary accuracy. This gives us the advantage of low computational complexity compared to the numerical integration in Mamdani fuzzy systems.

We will discuss several construction approaches.

The first approach is with a single input and single output dimension Mamdani fuzzy system, using multiple orders of a Takagi-Sugeno fuzzy system as an approximation.

The diagram of the whole process is as follows:



Consider a Mamdani System with the fuzzy rules as:

if x is A_i then y is B_i , i= 1, …, n

The Takagi-Sugeno transforms these fuzzy rules to:

if x is A_i then $y_i = a_i x + b_i$, i= 1, …, n Where the TS system is defined to be of the 1st order. (linear case)

Knowing the fuzzy rules and given the Mamdani system's antecedents and consequences

We can calculate coefficients as:

$$a_i = \frac{y_{i+1} - y_{i-1}}{x_{i+1} - x_{i-1}} \text{ and } b_i = y_i - a_i x_i$$

We consider A_i as $A_i = (t, x_{i-1,} x_i, x_{i+1})$ similar to what we considered in the construction of the Mamdani system.

Knowing this information, we obtain the Takagi-Sugeno output of the form:

$$\mathrm{TS}(x) = \frac{\sum_{i=1}^{n} A_i \; . \, (\mathrm{a_i} x + \mathrm{b_i})}{\sum_{i=1}^{n} A_i}$$

We can prove that we can approximate a Mamdani system using Takagi-Sugeno system in a similar way to **[3]**.

Theorem:

Let us consider a Mamdani fuzzy system satisfying the properties :

i) A_i - continuous with supp(A_i) = (x_{i-1} , x_{i+1})

ii) B_i - integrable with supp(B_i) = (y_{i-1} , y_{i+1})

Then M(x) can be approximated by the Takagi – Sugeno fuzzy system

$$\mathrm{TS}(x) = \frac{\sum_{i=1}^{n} A_i \; . \, (\mathrm{a_i} x + \mathrm{b_i})}{\sum_{i=1}^{n} A_i} \; ,$$

where $a_i = \frac{y_{i+1} - y_{i-1}}{x_{i+1} - x_{i-1}}$ and $b_i = y_i - a_i x_i$

with arbitrary accuracy. More over the following error estimate holds true

$|M(x) - TS(x)| \leq 3\delta_1 + c\delta_2$ with $\delta_1 = \max_{i=1 \dots n} | y_{i+1} - y_i|$

$\delta_2 = \max\limits_{i=1\ldots n} |x_{i+1} - x_i|$ and c $= \max\limits_{i=1\ldots n} |a_i|$

Proof:

Given $M(x)$ is the Mamdani system considered, we have

$$|M(x) - TS(x)| = \left| M(x) - \frac{\sum_{i=1}^{n} A_i(x) \cdot p_i(x)}{\sum_{i=1}^{n} A_i(x)} \right|$$
$$= \left| \frac{\sum_{i=1}^{n} A_i(x) \cdot M(x)}{\sum_{i=1}^{n} A_i(x)} - \frac{\sum_{i=1}^{n} A_i(x) \cdot p_i(x)}{\sum_{i=1}^{n} A_i(x)} \right|$$
$$\leq \frac{\sum_{i=1}^{n} A_i(x) \cdot |M(x) - p_i(x)|}{\sum_{i=1}^{n} A_i(x)}$$

since we have two overlapping consequences $x \in (A_j) \cup (A_{j+1})$ then we get :

$$|M(x) - TS(x)| \leq \frac{\sum_{i=j}^{j+1} A_i(x) \cdot |M(x) - p_i(x)|}{\sum_{i=j}^{j+1} A_i(x)}$$

Now we will estimate $|M(x) - p_k(x)|, k \in \{j, j+1\}$.

We have

$$|M(x) - p_k(x)| = \left| \frac{\int_c^d \bigvee_{i=1}^{n} (A_i(x) \wedge B_i(y)).y.dy}{\int_c^d \bigvee_{i=1}^{n} (A_i(x) \wedge B_i(y)).dy} - p_k(x) \right|$$

$$= \left| \frac{\int_c^d \vee_{i=1}^{n} (A_i \ (x) \ \wedge \ \ B_i \ (y) \ ) \, . \, (y - p_k(x)) \, . \, dy}{\int_c^d \vee_{i=1}^{n} (A_i \ (x) \ \wedge \ \ B_i \ (y) \ ) \, . \, dy} \right|$$

$$\leq \frac{\int_c^d \vee_{i=\mathrm{j}}^{j+1} (A_i \ (x) \ \wedge \ \ B_i \ (y) \ ) \, . \, |y - p_k(x)| \, . \, dy}{\int_c^d \vee_{i=\mathrm{j}}^{j+1} (A_i \ (x) \ \wedge \ \ B_i \ (y) \ ) \, . \, dy}$$

Knowing that we have only two overlapping active rules and

$y_j = (\mathrm{a_j} x + \mathrm{b_j})$ we get

$$\left| y - (\mathrm{a_j} x + \mathrm{b_j}) \right| \leq \left| y - y_j \right| + \left| y_j - (\mathrm{a_j} x + \mathrm{b_j}) \right|$$

$$\leq 3\delta_1 + \mathrm{a_j} \left| x_\mathrm{j} - x \right| = 3\delta_1 + \mathrm{a_j}\delta_2 \leq 3\delta_1 + \mathrm{c}\delta_2$$

where all j – values

$\delta_1 = \max\limits_{i=1\dots n} | y_{i+1} - y_i| \ \ \delta_2 = \max\limits_{i=1\dots n} | x_{i+1} - x_i|$ and $\mathrm{c} = \max\limits_{i=1\dots n} | a_i|$

We let

If $\delta_1 \rightarrow 0$ and $\delta_2 \rightarrow 0$ then we get

$\lim\limits_{\delta_1, \delta_2 \rightarrow 0} M(x) = TS(x)$, uniformly in x.

So the approximation property is true with an error estimation based on the rule overlapping and on the step size where the system is interpolating.

The pseudo code for calculating $TS(x) = \frac{\sum_{i=1}^{n} A_i \ . (a_i x + b_i)}{\sum_{i=1}^{n} A_i}$ is as follows:

```
top  =0;
bottom =0;
for k = 1 : t
  top  =0;
  bottom =0;
    for i = 1 : n
    top  = top  + Ai(k) * (ai*x(k)+bi);
    bottom = bottom + Ai(t);
  end
   out(k)= top  / bottom;
end
```

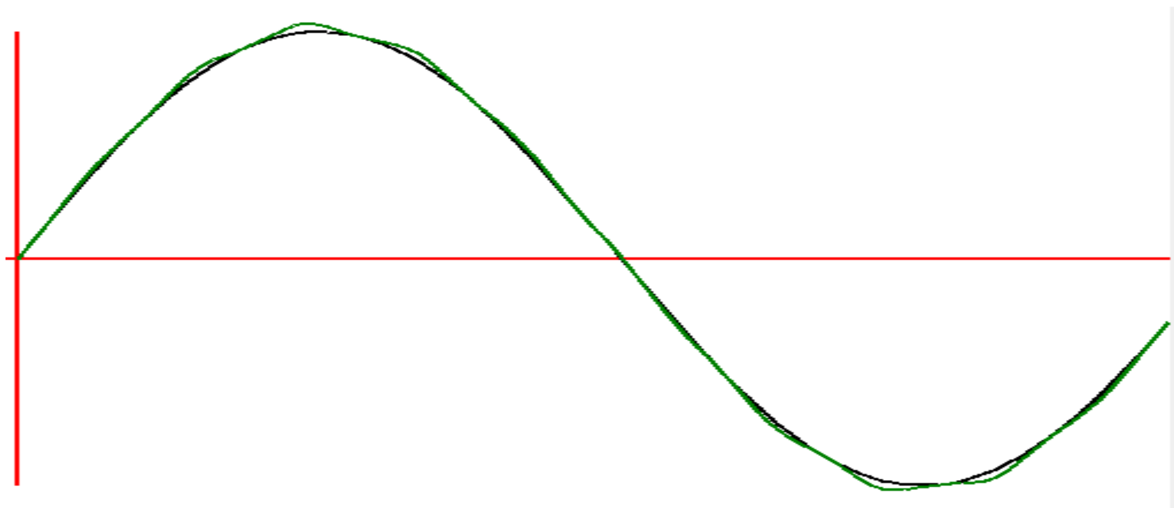


Figure 2.2 This graph shows the approximation of a curve using Mamdani fuzzy rules and approximated by Takagi-Sugeno system using linear function as the consequence.The black line is the original function and the green line is the approximation.

We can see that the time complexity and computational complexity is more efficient than using a Mamdani fuzzy system, the estimate in big O notation of the time complexity in this approach is $\boldsymbol{O}(\boldsymbol{n})$ where n is the number of rules.

Our second approach for a better TS approximation of a Mamdani System was implemented through quadratic interpolation.

Takagi-Sugeno are expandable from the point of view of the degree fo the output.

So for using a high order polynomial like a quadratic, we replace $y_i = a_i x + b_i$, with a quadratic $y_i = a_i x^2 + b_i x + c$.

In order to calculate the unknowns a, b, c we used polynomial interpolation.

For the quadratic polynomial interpolation we use the following approach:

Consider this equation:

$Ax = B$

Where A is a 3x3 Vandermonde matrix that contains the following:

$$A = \begin{bmatrix} (x_{i-1})^2 & x_{i-1} & 1 \\ (x_i)^2 & x_i & 1 \\ (x_{i+1})^2 & x_{i+1} & 1 \end{bmatrix},$$

and B is 3x1 matrix:

$$B = \begin{bmatrix} y_{i-1} \\ y_i \\ y_{i+1} \end{bmatrix}.$$

In order to calculate the coefficients a, b, and c we will find the inverse matrix of A and then multiply it with B as $x = A^{-1}B$,

where x is the following matrix:

$$x = \begin{bmatrix} a \\ b \\ c \end{bmatrix}.$$

The result was more accurate in its precision on approximating the curve.

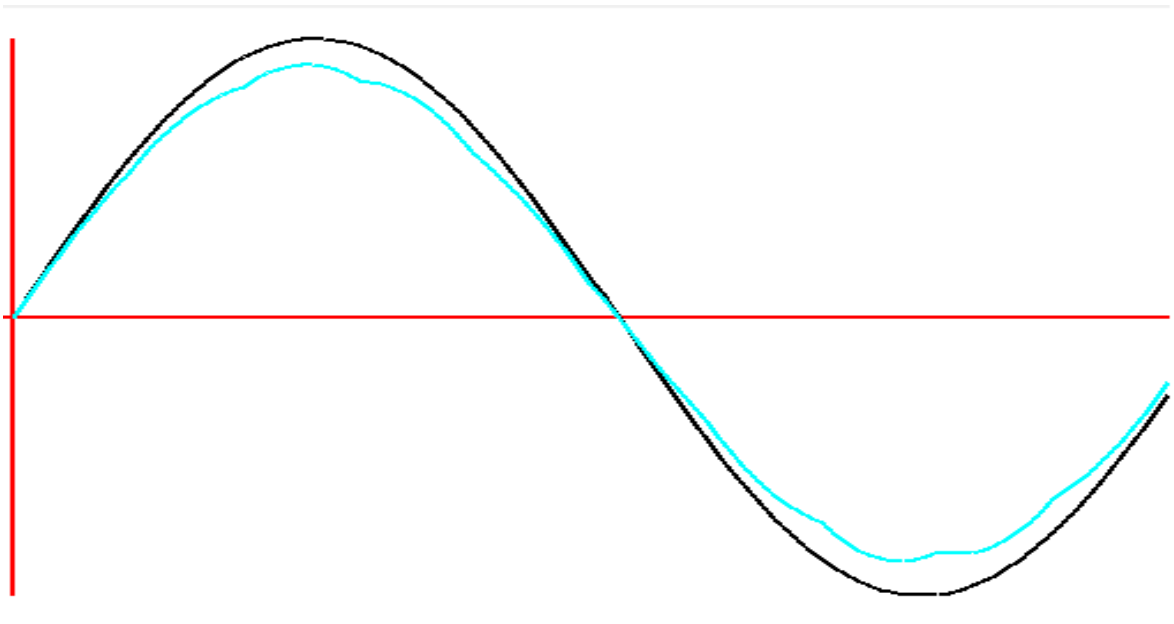


Figure 2.3 This graph shows the approximation of a curve using Mamdani fuzzy rules and approximated by Takagi-Sugeno system using quadratic polynomial interpolation as a consequence function. The black line is the original function and the cyan line is the approximation.

Our third approach for better TS approximation using Mamdani System was implemented through cubic B spline interpolation.

To add more to this functionality and more accuracy, we used B-splines as fuzzy rule outputs which transformed our output to:

$$\text{TS}\ (x)\ = \frac{\sum_{i=1}^{n} A_i(x).\sum_{i=0}^{n} Q_i(x)\ .}{\sum_{i=1}^{n} A_i(x)}$$

Where $Q_i(x) = (1-t)y_{i-1} +\ ty_i + t(1-t)(a_i(1-t) +\ b_i t)$,

$t = \frac{x - x_{i-1}}{x_i - x_{i-1}}$ and

$$a_i = k_{i-1}(x_i - x_{i-1}) - (y_i - y_{i-1})$$

$$b_i = -k_i(x_i - x_{i-1}) + (y_i - y_{i-1})$$

For this example, we used 3 sample points to construct the B-Spline.

The values of k_0, k_1, k_2 are found by solving the tridiagonal linear equation system.

$$\begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & a_{23} \\ 0 & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} k_0 \\ k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

where:

$$a_{11} = \frac{2}{x_1 - x_0}, \; a_{12} = \frac{1}{x_1 - x_0}, \; a_{21} = \frac{1}{x_1 - x_0}, \; a_{22} = 2\left(\frac{1}{x_1 - x_0} + \frac{1}{x_2 - x_1}\right)$$

$$a_{23} = \frac{1}{x_2 - x_1}, \; a_{32} = \frac{1}{x_2 - x_1}, \; a_{33} = \frac{2}{x_2 - x_1}$$

$$b_1 = 3\frac{y_1 - y_0}{(x_1 - x_0)^2}, \; b_2 = 3\left(\frac{y_1 - y_0}{(x_1 - x_0)^2} + \frac{y_2 - y_1}{(x_2 - x_1)^2}\right), \; b_3 = 3\frac{y_2 - y_1}{(x_2 - x_1)^2}$$

The B-spline approach gave the best approximation with respect to the original function but the quadratic approach gave a better approximation with respect to Mamdani fuzzy system.

In addition to that, B-splines are also known for their approximation properties. This can be considered as a future work combining and understanding how a B-spline is directly related to fuzzy systems.

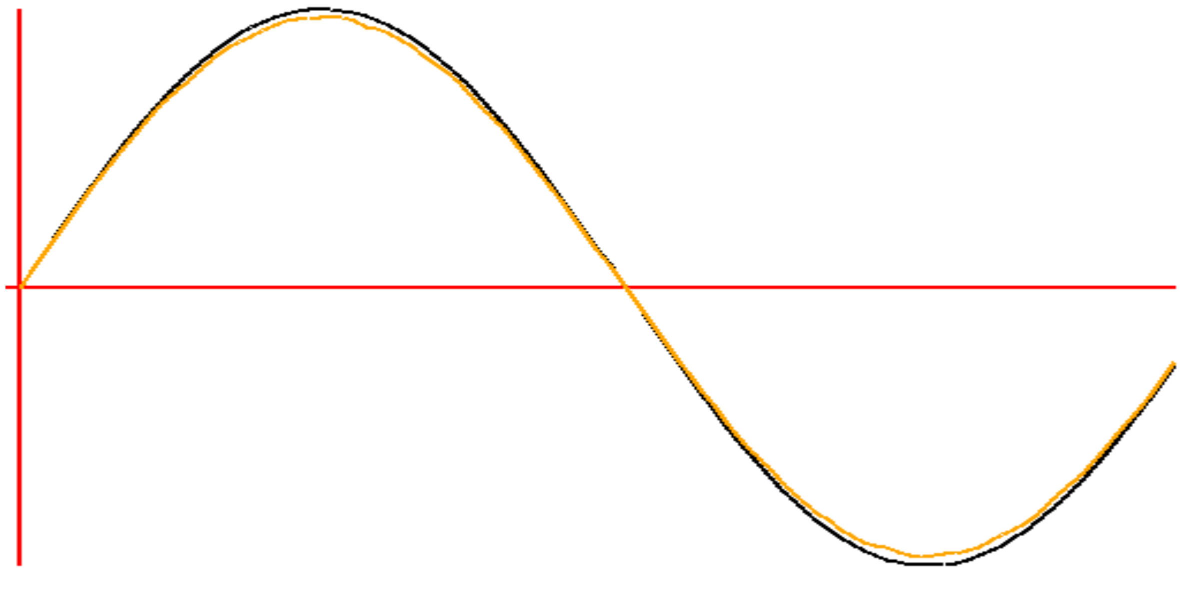


Figure 2.4 This graph shows the approximation of a curve using Mamdani fuzzy rules and approximated by Takagi-Sugeno system using Cubic Bsplines interpolation as a consequence function. The black line is the original function and the orange line is the approximation.

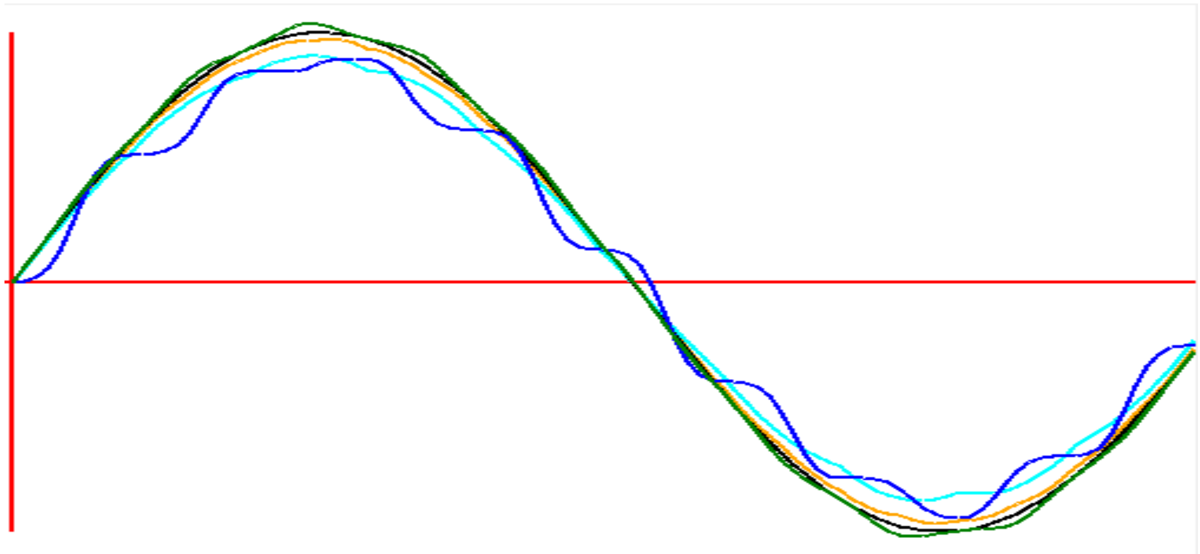


Figure 2.5 This graph shows the approximation of a curve using Mamdani fuzzy rules and approximated by Takagi-Sugeno system of various orders and the Mamdani approximation of the original curve. These systems were based on 11 rules for approximation.

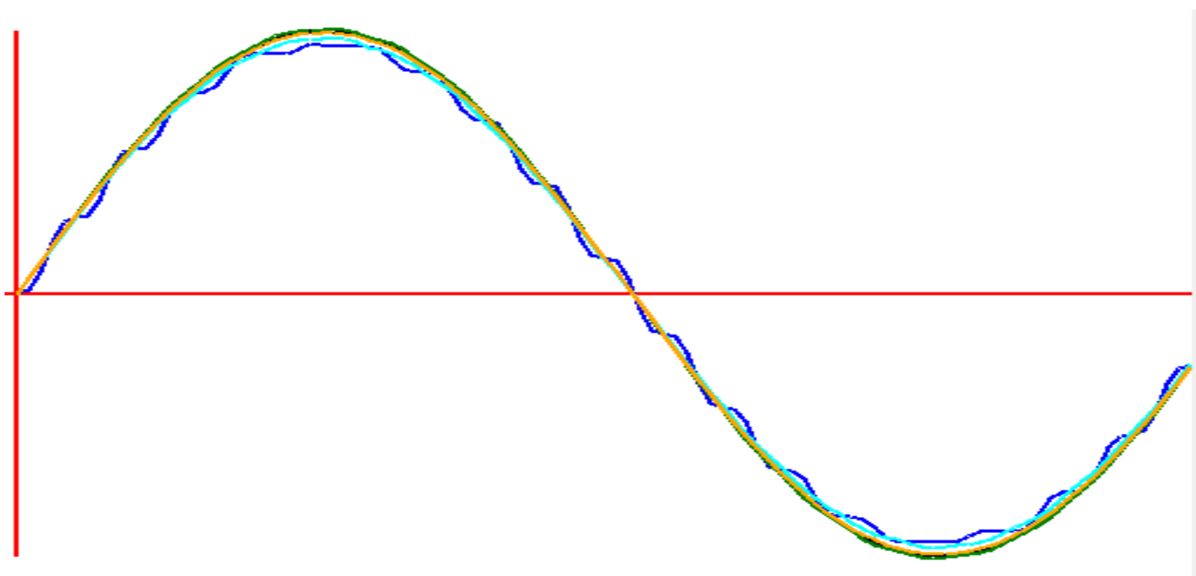


Figure 2.6 This graph shows the approximation of a curve using Mamdani fuzzy rules and approximated by Takagi-Sugeno system of various orders and the Mamdani approximation of the original curve. These systems were based on 21 rules for approximation.

The Blue line represents the mamdani fuzzy system approximation of the curve. The green line represents the TS fuzzy system using the linear approach. The cyan line represents the TS fuzzy system using the quadratic approach. The orange line represents the TS fuzzy system using the cubic spline approach.

Rules/Approach	11 Rules	21 Rules	31 Rules
Linear	0.22	0.08	0.024
Quadratic	0.14	0.04	0.022
Cubic Spline	0.17	0.06	0.023

Table showing the error of approximation of TS with Mamdani fuzzy systems

How to Approximate a Mamdani Fuzzy System using a Two Dimensional TS System?

Since our approach was successful we have to make sure it can work with different degrees and dimensions in order to fully cover the basis of approximation. A Mamdani fuzzy system can have several input values and several outputs. We solved in the previous section the base case which handled one input and one output. In order to provide enough evidence for the advantages of the proposed theory we extended the base case to an advanced case where the Mamdani system is constructed based on two inputs and one output. This is a two dimensional system where each input corresponds to a dimension and the output represents the axis perpendicular to the plane formed by the input axis.

Similar to the implementation of the 1^{st} approach we need to construct a Mamdani fuzzy system to approximate a function with 2 variables since the general method has already been proven and our theory will be constructed based on the same ideas.

Consider a Mamdani fuzzy system with a fuzzy rule base as:

if x is A_i and y is B_j then z is $C_{i,j}$, i, j= 1, …, n

Step 1:

We consider sample values from the function.

Let $f(x_i, y_j) = z_{i,j}$, $i = 0, \dots, n+1, j = 0, \dots, m+1$.

Step 2:

We construct A_i & B_j which represent the antecedent fuzzy sets on the x axis and y axis respectively.

To construct A_i & B_j we use membership functions such as triangle, trapezoid, Splines, etc..

The antecedent is constructed using value ranges t, s with a step size of the user's choice, and the sample points $x_i, i = 1, \dots, n$ and $y_j, i = 1, \dots, m$ as a support for each fuzzy set.

For example:

$A_i = (t, x_{i-1,} x_i, x_{i+1})$ & $B_j = (s, y_{j-1,} y_j, y_{j+1})$ triangular fuzzy sets.

In order to compute consequence of the fuzzy rules, we use our output dimensional values that we obtain from sampling the function.

The fuzzy sets of the consequence parts are considered integrable and are constructed such as the value range is $l = \min(z), \dots, \max(z)$ with a step size of the user's choice . To compute the support for each fuzzy set we use all the sample points from the matrix

$$\begin{bmatrix} Z_{i-1,j-1} & \cdots & Z_{i-1,j+1} \\ \vdots & \ddots & \vdots \\ Z_{i+1,j-1} & \cdots & Z_{i+1,j+1} \end{bmatrix}$$

to get the minimum and maximum points as support and we consider that the center $Z_{i,j}$ is the core of the fuzzy set that defines the consequence. So we get:

$C_{i,j} = (l , \min, Z_{i,j} , \max)$ which represents a fuzzy set. In the most general situation we have $m.n$ fuzzy rules and consequences which is consistent with our intuitive knowledge on function approximation.

Step 3:

The approximation of the function is done at this step, by combining the SISO fuzzy system with a Mamdani fuzzy rule base with the COG defuzzification. The combination will result into a summation function:

$$M\ (\mathrm{f},\ x,y) = \frac{\int_{w}^{-w} \vee_{\boldsymbol{i=1}}^{\boldsymbol{n}} \vee_{\boldsymbol{j=1}}^{\boldsymbol{n}} (\ \boldsymbol{A_i}\ \ (\boldsymbol{x}) \wedge\ \boldsymbol{B_i}\ \ (\boldsymbol{y})\ \wedge \boldsymbol{C_{ij}}\ \ (\boldsymbol{z}))\boldsymbol{z}\ dz}{\int_{w}^{-w} \vee_{\boldsymbol{i=1}}^{\boldsymbol{n}} \vee_{\boldsymbol{j=1}}^{\boldsymbol{n}} (\ \boldsymbol{A_i}\ \ (\boldsymbol{x}) \wedge\ \boldsymbol{B_i}\ \ (\boldsymbol{y})\ \wedge \boldsymbol{C_{ij}}\ \ (\boldsymbol{z}))}$$

We approximate the integral with a Riemann sum.

The pseudo code is as follows:

```
for k1 = 1 : t
  for k2 = 1 : t
     top  =0;
     bottom =0;
         for L = 1:  l
              temp =0;
                    for i = 2 : n – 1
                     for j= 2 : m – 1
                              tmp1 = tnorm( A(i,k1), B(j,k2),C(l,i,j)  );
                              temp =max( [tmp1, mtmp] );
                      end
                     end
              top  = top  + temp * l(j);
              bottom = bottom + temp;
          end
      out(k)= top  / bottom;
    end
end
```

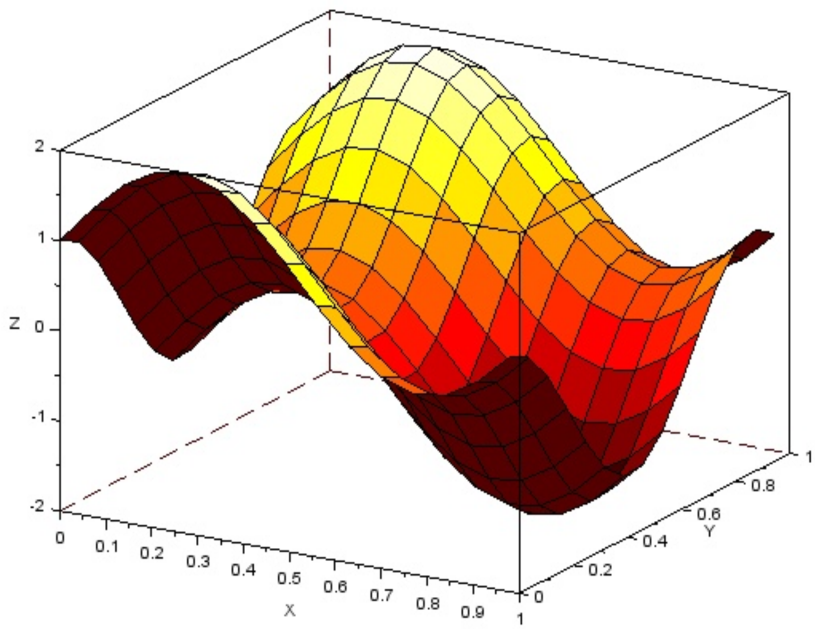


Figure 3.0 This graph represents the side view of the 3 dimensional function we used to in order to test our approximation method. *f*(x,y) = sin(6x)+cos(7y)

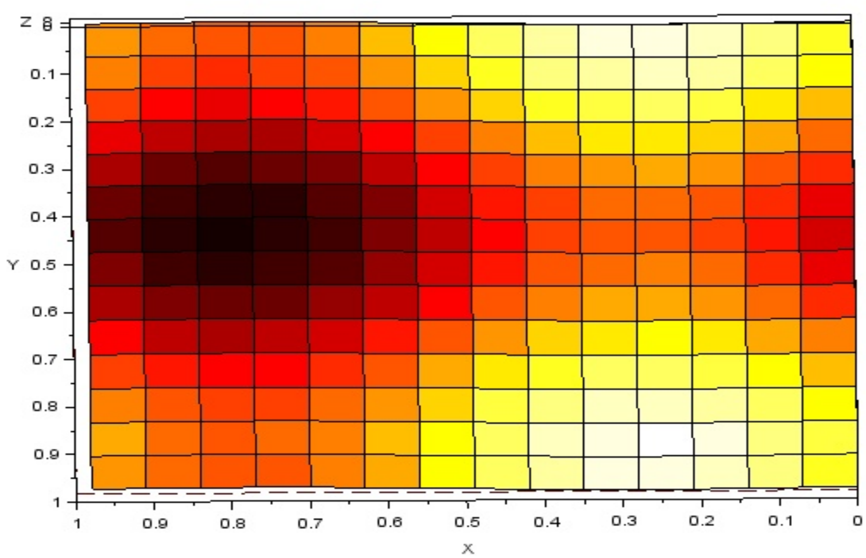


Figure 3.1 This graph represents the top view of the 3 dimensional function we used to in order to test our approximation method. *f*(x,y) = sin(6x)+cos(7y)

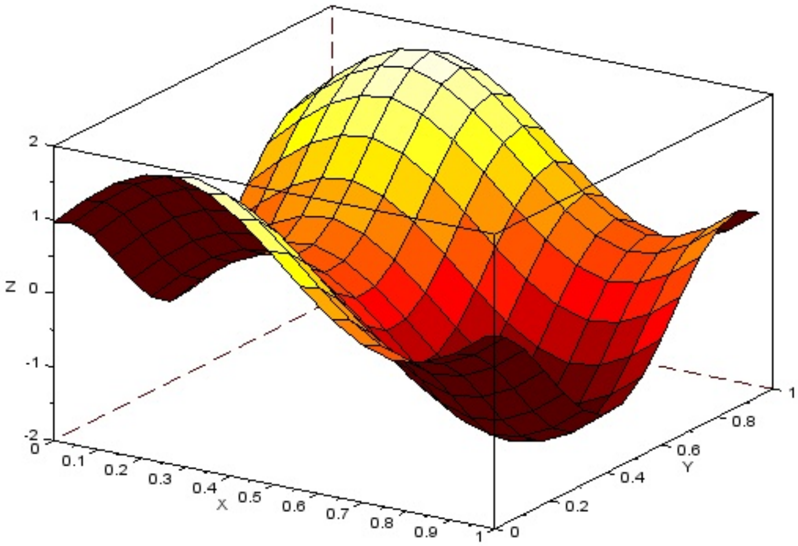


Figure 3.2 This graph represents the side view of the approximation using Mamdani systems with 14 fuzzy rules.

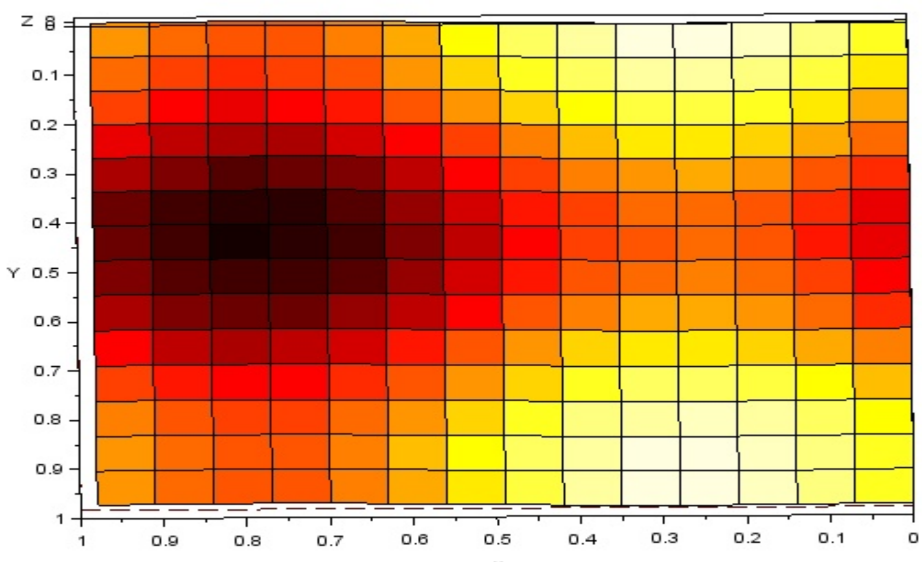


Figure 3.3 This graph represents the top view of the approximation using Mamdani systems with 14 fuzzy rules.

The time complexity to approximate the function using a Mamdani fuzzy system is $\boldsymbol{O(m.n.l)}$ & $\boldsymbol{l \geq m.n}$, $\boldsymbol{l}$ being the integration step.
As we can see the more complex the system is, the more complex the approximation will be and the efficiency will decrease with the increase of that complexity factor, which is natural. To reduce the complexity we user our proposed approach to construct a TS fuzzy system to approximate the given Mamdani System.

To test the theory of the TS approximation of this Mamdani system, we considered the following linear output for each fuzzy rule.

The premise is:

if x is A_i and y is B_j then
$Z_{ij} = a_{ij}x + b_{ij}y + c_{ij}$; i= 1, …, n, j= 1, …, m.

and the output of the TS fuzzy system is considered to be:

$$T(x,y) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}(A_i(x).\ B_j(y)\ .\ Z_{ij})}{\sum_{i=1}^{n}\sum_{j=1}^{n}(A_i(x).\ B_j(y))} \quad \text{formula (1)}$$

In order to find the unknown coefficients a_{ij}, b_{ij}, c_{ij} we use a similar approach to the method we used for finding the support endpoints of the consequence fuzzy sets.

First we find the minimum of the support denoted by l_{ij} and the maximum of the support denoted by , r_{ij}. $l_{ij}$and r_{ij} are calculated by looping through the entire matrix and finding the max and min values in 3x3 sub matrices.

After finding $l_{ij}$and r_{ij} of sample points z_{ij}, we consider the core to be $m_{ij} = z_{ij}$ for the sample points neighboring z_{ij}.

Pseudo code for finding the coefficients:

```
for i =2 : size(x) -1
   for j=2:size(y) -1

      left = min(z(i-1:i+1,j-1:j+1));
      right = max(z(i-1:i+1,j-1:j+1));
      middle = z(i,j);

      a(i,j) = (right - left ) /  ( x(j-1) - x (j +1 ) );
      b(i,j) = (right - left ) /  ( y(j-1) - y (j +1 ) );
      c(i,j) = middle - a(i,j)* x(j) - b(i,j) * y(j);


   end
end
```

All the unknowns have been found, we can now compute the output of the TS fuzzy system using the formula (1).

Pseudo code for computing the TK approximation using Mamdani system is as follows:

```
for k1 = 1: size(t)
   for k2 =1:size(s)
      top=0;
      bottom=0;
         for I = 2: size(x)-1
            for j = 2: size(y) -1
              top    = top + ((A(i,k1)*B(j,k2) ) * (  a(i,j)*t(k1)+b(i,j)*s(k2)+c(i,j)))
              bottom = bottom + (  A(i,k1)*B(j,k2) )
            end
         end
     out(k1,k2) =  (top / bottom )
   end
end
```

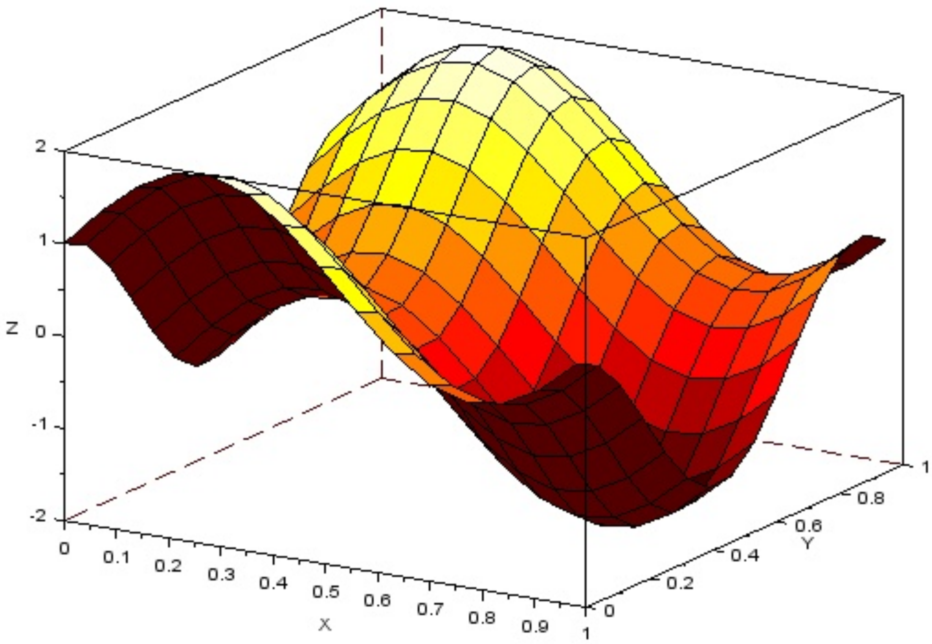


Figure 3.3 This graph represents the side view of the approximation of Mamdani fuzzy sets using Takagi-Sugeno system.

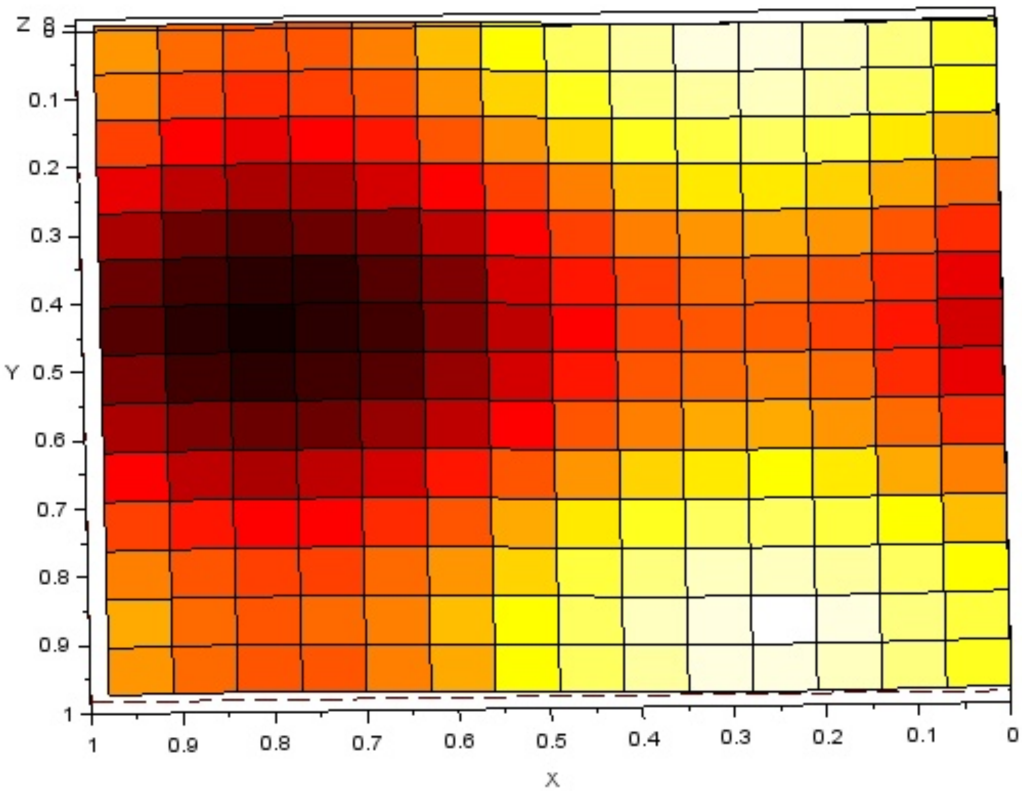


Figure 3.4 This graph represents the top view of the approximation of Mamdani fuzzy sets using Takagi-Sugeno system.

The complexity of the TS fuzzy system with respect to Mamdani fuzzy system is simpler and much more efficient with unnoticeable error margin. The big O notation for this approximation is $\boldsymbol{O}(\boldsymbol{m}.\boldsymbol{n})$ where $\boldsymbol{m}.\boldsymbol{n}$ is the number of rules.

How to use the Approximation in a Fuzzy Control System?

Why do We use Fuzzy Controller System?

Fuzzy logic is a technique to embed human like thinking into a control system. A fuzzy control is designed to emulate human deductive thinking, that is, the process people use to infer conclusions from what they know. Although genetic algorithms and neural networks can perform just as well as fuzzy logic in many cases, fuzzy logic has the advantage that the solution to the problem can be casted in terms that human operators can understand so that the experience can be used in design of the control.

A fuzzy control system represents a mathematical system that analyzes inputs and outputs and links them through the fuzzy rules. Fuzzy controllers are very simple conceptually. They consist of an input stage, a processing stage, and an output stage. The input stage maps sensor or other inputs, such as switches, thumbwheels and so on, to the appropriate membership functions and membership values. The processing stage invokes each appropriate rule and generates a result for each, then combines the results of rules. The output stage converts the combined result back into a specific control output value.

A game developer uses fuzzy controllers in games to create an AI that follows fuzzy rules. For example, most of the games have a character movement control that can be used to move and navigate game units, vehicle aircrafts, foot units, etc. through waypoints and around obstacles. Also fuzzy controllers are used for threat assessments where the enemy assesses the threat of the other characters, AI will determine if your threat level is low, medium or high. So we can see, fuzzy controllers are implemented in multiple areas in games. As a consequence, optimizing these controllers will benefit the games performance without taking a hit in the quality of the game's artificial intelligence.

Usually the fuzzy controller's fuzzy sets and output are precomputed and saved, and then we use these fuzzy sets to determine the output of the fuzzy inputs given inside the game. However, if we want to add a new rule in the middle of the game play run time we have to compute again the weight of each fuzzy set in order to have a complete working fuzzy system. Mamdani fuzzy systems are usually the best systems to design fuzzy controllers since they are precise, interpretable and all the variables are known so our output will not be a guess of the fuzzy the controller.

How to Construct Fuzzy Controllers using Takagi-Sugeno as Approximation to a Mamdani System

As we stated earlier, fuzzy rules are of the form of if … then… and each rule is made up of linguistic variables. The linguistic variables are fuzzy sets with values that were calculated from a membership function. From these values we build a relationship to Takagi-Sugeno system that approximates the given Mamdani system.

So as a result the fuzzy rule set changed from:

if x is A_i and $y\ is\ B_j$ then $z\ is\ C_{i,j}$
to
if x is A_i and $y\ is\ B_j$ then $z = a_{ij}X + b_{ij}Y + c_{ij} \quad \forall\ i = 0, \dots n, j = 0, \dots, m$.

In the Takagi-Sugeno approach we have unknown variables such a_{ij}, b_{ij}, and c_{ij} that are used in the above general rule. These variables are usually given or obtained using adaptive techniques such as machine learning and neural networks. In the proposed method we construct the TS system with simple linear consequences, directly avoiding, avoiding the computation complexity of the learning algorithms.

How to compute the unknown coefficients?

Consider a fuzzy rule:

If A_i (x) and B_j (y) then C_{ij}

Where A_i (x) = (t, $x_{i-1}, x_i, x_{i+1})$, B_j (y) = (t, $y_{j-1}, y_j, y_{j+1})$ and

C_{ij} = (t, $l_{ij}, m_{ij}, r_{ij})$ represent the membership function used, in this case it was a triangular membership function.

$$\boldsymbol{Z_{ij} = a_{ij}X_i + b_{ij}Y_j + c_{ij}} \ .$$

$$\frac{\partial z_{ij}}{\partial x_i} = \mathbf{a_{ij}} = \frac{r_{ij} - l_{ij}}{\mathbf{X_{i+1} - X_{i-1}}}$$

$$\frac{\partial z_{ij}}{\partial y_j} = \mathbf{b_{ij}} = \frac{r_{ij} - l_{ij}}{\mathbf{y_{i+1} - y_{i-1}}}$$

$$m_{ij} = \mathbf{a_{ij}} X_i + \mathbf{b_{ij}} Y_j + \mathbf{c_{ij}}$$

then

$$\mathbf{c_{ij}} = m_{ij} - \mathbf{a_{ij}} X_i - \mathbf{b_{ij}} Y_j$$

r_{ij} and l_{ij} represent the endpoints support of the membership function and m_{ij} represents the core of the membership function.

Using these coefficients, we are able to approximate the Mamdani fuzzy system using Takagi-Sugeno with significantly less computation complexity. The reduced complexity makes it much more suitable for applications and for learning.

In order to test our work, we used a fuzzy control example proposed in **[1]** which talks about a hypothetical crime scene. A friend claimed to have found a body and called the police. In order for the detective on the scene to believe the story that the friend was not a murderer, he had to touch the engine of the car to estimate the temperature of the engine. The detective also asked about his trip from where the man left home till he arrived to his friend's home. This is a very interesting problem since we have a variety of fuzzy variable inputs that includes the duration of the trip from the origin point to the friend's house and the other input is how much time did he have to arrange the crime scene. These two variables cannot be described as real numbers but can be described as fuzzy variables. And the output of these two inputs is the approximate temperature of the engine should be. The result will state if the man's alibi is true.

The proposed implementation takes a text file that represents the linguistic variables and the values to construct the fuzzy sets, and another file that has the rules using these variables. This example is in line with the computing with words paradigm of Zadeh.

For this test we used the following linguistic variables:

DriveDuration 5 ANTECEDENT
VSmall Triangle 0 10 20
Small Triangle 10 20 30
Medium Triangle 20 30 40
High Triangle 30 40 50
VHigh Triangle 40 50 60
TimeStopped 5 ANTECEDENT
VSmall Triangle 0 10 20
Small Triangle 10 20 30
Medium Triangle 20 30 40
High Triangle 30 40 50
VHigh Triangle 40 50 60
EngineTemperature 5 CONSEQUENCE
VCold Triangle 0 10 20
Cold Triangle 10 20 30
Medium Triangle 20 30 40
Hot Triangle 30 40 50
VHot Triangle 40 50 60

And the following rules:

If DriveDuration = Small and TimeStopped = Medium then EngineTemperature = Cold

If DriveDuration = Medium and TimeStopped = High then EngineTemperature = Medium

If DriveDuration = High and TimeStopped = Small then EngineTemperature = Hot

If DriveDuration = VHigh and TimeStopped = VSmall then EngineTemperature = VHot

If DriveDuration = VSmall and TimeStopped = VHigh then EngineTemperature = VCold

Results

And the results turned out to be as expected

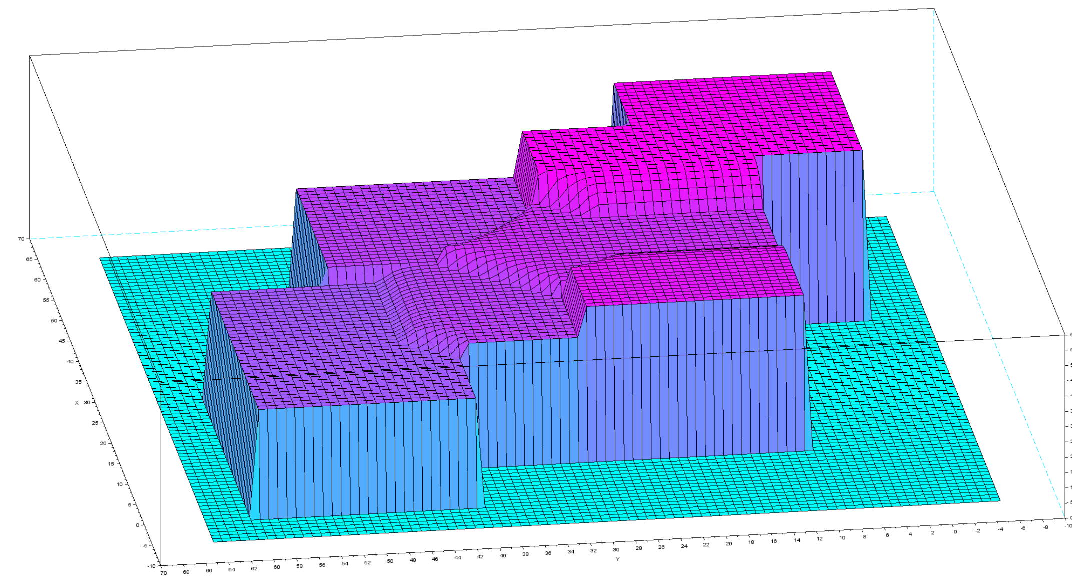


Figure 4.0 This graph represents a model that was computed through a fuzzy controller that used words as weighted values. The model was constructed using Mamdani SISO fuzzy system.

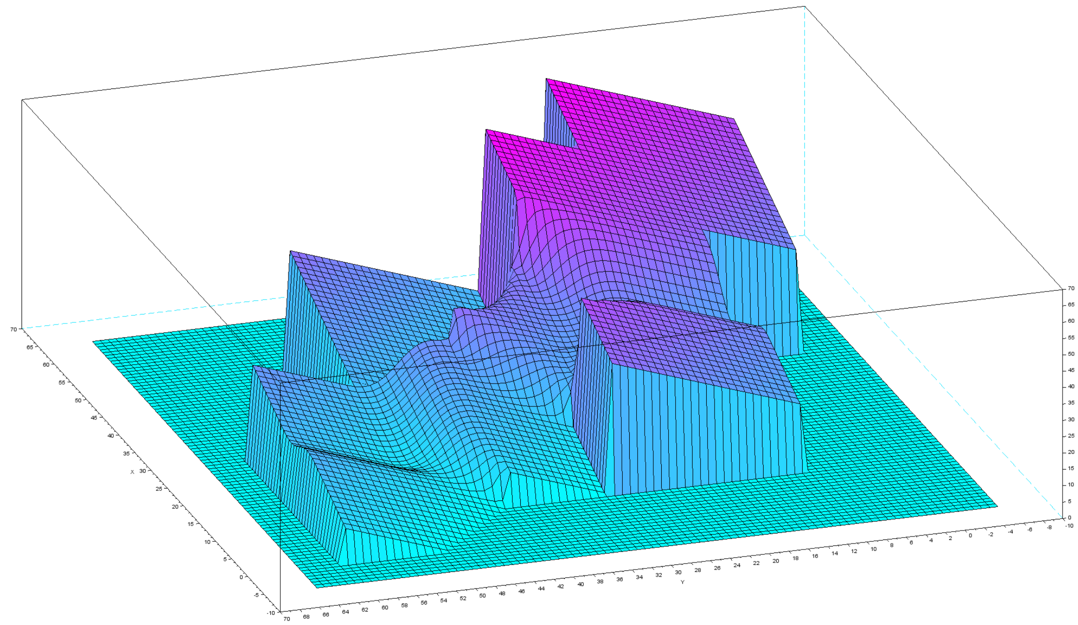
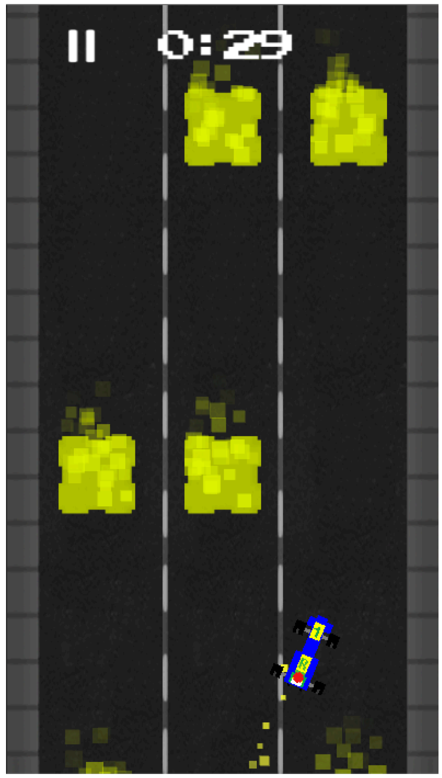


Figure 4.1 This graph represents a model that was computed through a fuzzy controller that used words as weighted values. The fuzzy sets were constructed using Mamdani inference system and then approximated using Takagi-Sugeno.

This example shows that the TS approximation of Mamdani system can be derived directly from the Mamdani system without going through a function approximation.

Another implementation of the proposed TS approximation of a Mamdani system was done on a mobile game application which is a 2D fast paced game where the player is a car maneuvering through obstacles made up of cubes.

A fuzzy control was used in order to control the difficulty of the game. The difficulty of the game is based on the distance between the row of cubes and the input of the game was based on the reaction time of the player and how fast the car is moving. Using these two input variables we were able to derive a number to determine how far the row of cubes should be from each other.


0:29

Chapter 5

Future Work

To improve on our results, as a future work, we propose implementing a neuro fuzzy system that adjusts the values of the unknown coefficients of the Takagi-Sugeno System, then using these values to reconstruct the Mamdani system from Takagi-Sugeno system **[12]**. As the starting points for the learning algorithm we plan to use the constructive approach presented in this work. This will give us a two-way connection from Takagi-Sugeno and Mamdani systems which can be helpful for more accurate representation of a fuzzy controller by using the positive aspects of both fuzzy systems and combining them together. But there might be some runtime performance complications since adaptive learning is expensive and recomputing the system by Mamdani is also expensive so there should be studies on how to optimize the learning curve of the values in order to save performance issues.

References

[1]. B. Bede, Mathematics of Fuzzy Sets and Fuzzy Logic, Springer, 2013.

[2]. B. Bede, I.J. Rudas, Approximation properties of higher order Takagi-Sugeno Fuzzy systems. IEEE 2013 IFSA World Congress NAFIPS Annual Meeting Ed-monton, Canada, 2013: 1-6.

[3]. Bede, Barnabas, and Imre J. Rudas. "Takagi-Sugeno Approximation of a Mamdani Fuzzy System." Advance Trends in Soft Computing. Springer International Publishing, 2014. 293-300.

[4]. E.H. Mamdani, S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller, J. Man Machine Stud., 7(1975) 1-13.

[5]. Hao Ying, General Takagi-Sugeno fuzzy systems with simplified linear rule conse- quent are universal controllers, models and filters, Information Sciences, 108(1998), 91-107.

[6]. Jang, Jyh-Shing Roger. "ANFIS: adaptive-network-based fuzzy inference system." *Systems, Man and Cybernetics, IEEE Transactions on* 23.3 (1993): 665-685.

[7]. Kosko, Bart. "Fuzzy systems as universal approximators." *Computers, IEEE Transactions on* 43.11 (1994): 1329-1333.

[8]. Kukolj, Dragan. "Design of adaptive Takagi–Sugeno–Kang fuzzy models."*Applied Soft Computing* 2.2 (2002): 89-103.

[9]. L.A. Zadeh, Fuzzy Sets, Information and Control, 8(1965) 338-353.

[10]. L.X. Wang, Adaptive Fuzzy Systems and Control, Design and Stability Analysis, Prentice-Hall, Englewood Cliffs, NJ, 1994.

[11]. L.X. Wang, J.M. Mendel, Generating fuzzy rules by learning from examples, IEEE Trans. Systems Man Cybernet. 22 (6) (1992) 1414–1427.

[12]. S. Abe, Fuzzy function approximators with ellipsoidal regions, IEEE Trans. Syst. Man Cybern, Part B 29 (4) (1999) 654–661.

[13]. Singh, Harpreet, et al. "Real-Life Applications of Fuzzy Logic." *Adv. Fuzzy Systems* 2013 (2013): 581879-1.

[14]. S.J. Kang, C.H. Woo, H.S. Hwang, K.B. Woo, Evolutionary design of fuzzy rule base for nonlinear system modeling and control, IEEE Trans. Fuzzy Syst. 8 (2000) 37–45.

[15]. S. Mitaim, B. Kosko, The shape of fuzzy sets in adaptive function approximation, IEEE Transactions on Fuzzy Systems, 9(2001) 637-656.

[16]. Takagi, Tomohiro, and Michio Sugeno. "Fuzzy identification of systems and its applications to modeling and control." *Systems, Man and Cybernetics, IEEE Transactions on* 1 (1985): 116- 132.

[17]. Takagi,M.Sugeno,Fuzzyidentificationofsystemsanditsapplicationstomodellingandcontrol,IEEETrans.Systems Man Cybernet. 15 (1985) 116–132.

[20]. Tung, Whye Loon, and Chai Quek. "A mamdani-takagi-sugeno based linguistic neural-fuzzy inference system for improved interpretability-accuracy representation." *Fuzzy Systems, 2009. FUZZ-IEEE 2009. IEEE International Conference on*. IEEE, 2009.

[19]. V. Cherkassky, D. Gehring, F. Mulier, Comparison of adaptive methods for function estimation from samples, IEEE Trans. Neural Networks 7 (4) (1996) 969–984.

[20]. Y. Lin, G.A. Cunningham Jr., S.V. Coggeshall, Using fuzzy partitions to create fuzzy systems from input–output data and set the initial weights in a fuzzy neural network, IEEE Trans. Fuzzy Syst. 5 (4) (1997) 614–621.