# REAL-TIME FUZZY CONTROL USING LOWER-UPPER REPRESENTATION OF FUZZY NUMBERS

BY

Matthew Paul Peterson

*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

March
2016

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

Real-Time Fuzzy Control Using Lower-Upper Representation of Fuzzy Numbers

BY

Matthew Paul Peterson

HAS BEEN SUCCESSFULLY COMPLETED ON March 25, 2016.

MAJOR FIELD OF STUDY: COMPUTER SCIENCE.

APPROVED:

| | |
|---|---|
| Dmitri Volper date<br>Graduate Program Director | Xin Li date<br>Dean of Faculty |
| Pushpak Karnick date<br>Department Chair, Computer Science | Claude Comair date<br>President |

DIGIPEN INSTITUTE OF TECHNOLOGY

GRADUATE STUDIES PROGRAM

*THESIS APPROVAL*

*DATE:* March 25, 2016

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

Matthew Paul Peterson

ENTITLED

Real-Time Fuzzy Control Using Lower-Upper Representation of Fuzzy Numbers

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<br>Thesis Committee Chair | Matthew Klassen date<br>Thesis Committee Member | Pushpak Karnick date<br>Thesis Committee Member |
| Eva Maria Iwer date<br>Thesis Committee Member | Dmitri Volper date<br>Thesis Committee Member | |

Dedicated to Sylvia.

## ACKNOWLEDGMENTS

My thesis advisor, Barnabas Bede, has been incredibly supportive throughout the entire creation of this thesis. I would like to thank him for his longsuffering over thousands of miles and many years of labor. I would also like to thank the extremely understanding managers and executives at Nintendo Technology Development and Retro Studios who encouraged and pushed me to complete this thesis.

## ABSTRACT

Fuzzy systems are used in many real-world applications and are known for their ability to produce consistently desireable output while in unconstrained environements. Takagi-Sugeno fuzzy controllers are often chosen over Mamdani fuzzy inference systems when implementing a real-time solution due to their reduced computational complexity, but Mamdani systems are advantageous in their interpretability. Recent developments in fuzzy systems include the lower-upper (LU) parametric representaion, in which fuzzy numbers are defined using rational splines, with the membership grades in the domain as opposed to tradional left-right (LR) representations where the universe of discourse is in the domain.

In this thesis, an LU fuzzy inference system is proposed which can greatly reduce the computational complexity while still maintaining the exibility and intuitive interpretability that Mamdani fuzzy controllers have over Takagi-Sugeno fuzzy controllers. This is accomplished by dening non-parametric LU fuzzy numbers based on corresponding LR fuzzy numbers and then integrating along the consistenly-bounded memberhsip grade domain rather than the variably-bounded universe of discourse that LR-based Mamdani fuzzy inference systems require.

TABLE OF CONTENTS

Page

# LIST OF TABLES

## LIST OF FIGURES

Figure | Page

Figure Page

| Figure | | Page |
| --- | --- | --- |

CHAPTER 1

# Introduction

Fuzzy controllers have the ability to adapt to potentially unconstrained scenarios with multiple variables and unclear boundaries. Many real-world applications have taken advantage of fuzzy logic and fuzzy controllers. Fuzzy logic has been used in washing machines to more efficiently manage mechanical operations such as water usage and spin control based on the dirtiness and variety of fabric that is being cleaned [6]. Fuzzy controllers can be used in air conditioners by taking user temperature settings, actual room temperature, and the dew point temperature as inputs and manipulate the compressors, fans, fins, and operation modes as outputs [7, 8, 9]. A fuzzy traffic controller based on the arrival and queue of vehicles can even control the time delay on traffic lights [10]. This versatility in solving real-time problems makes fuzzy logic and fuzzy controllers obvious candidates for addressing the complexities that arise in games.

Fuzzy logic can be used in games to control agents, assess threats, and classify characters. For example, in Quake III Arena Bot [3], fuzzy logic is used to express utility. How much something wants to do, have, or use something is often classified under utility theory, which is inherent in fuzzy logic. This classification under utility theory has resulted in games such as Kohan 2: Kings of War and Axis & Allies, Prototype, Iron Man, Red Dead Redemption, and All Heroes Die utilizing fuzzy logic without the deliberate application or knowledge of its definitions [2]. Research has been done on utilizing fuzzy logic in agent-based game design [11], for artificial intelligence in games [5], and even the real-time game design of Pac-Man [4].

The primary goal of this thesis is to define and study a new class of fuzzy controllers using the lower-upper (LU) representation of a fuzzy number and to compare them to those of the more traditional (LR) representations, giving special consideration and evaluation of the benefits that arise in their applications toward game development. A secondary goal is to investigate the merits and practicality of using fuzzy controllers in a real-time environment. The basics of fuzzy sets and fuzzy logic are covered, followed by how they are used in fuzzy controllers. Then, more specifically, LU representations are discussed. Finally, the use of LU-fuzzy control is proposed and evaluated. Due to the versatility that LU-fuzzy controllers offer, different areas of real-time applications will be driven using fuzzy controllers in order to examine their potential applications. Potential applications include the sampling of an environment that includes multiple agents and then modifying parameters which affect the system being driven via a single fuzzy controller. Another potential application is explored by driving modular behaviors of individual agents and singular systems using multiple fuzzy controllers. By comparing the advantages and disadvantages that LU-fuzzy controllers provide in scenarios at these different scales, a more functional assessment is hoped to be made over when and where they can best be utilized when developing a game of any given scope.

CHAPTER 2

# Fuzzy Sets and Logic

Lotfi A. Zadeh defined a fuzzy set as a class with a continuum of membership grades [17]. That simply means that if there is a referential universe of discourse, $X$ (for example, how old someone is), then each element in $X$, $x$ (for example, 42 years old), can be mapped to a real number in the closed interval from zero to one (for example, the real number 0.4). The final mapping is the membership grade for that element; a mapping of zero implies no membership, while a mapping of one implies full membership. Without the continuum of membership grades between zero and one, a fuzzy set behaves exactly like a classical set (see Fig. 1).

Let $F(X)$ represent the class of all fuzzy subsets of $X$. A fuzzy set, $A$, can be defined as a simple mapping:

$$A : X \to [0, 1].$$

This mapping can be used to define any fuzzy set. For example, the fuzzy relationship between hours of sleep and sleep depth throughout a given sleep cycle can be represented using such a mapping (see Fig. 2).

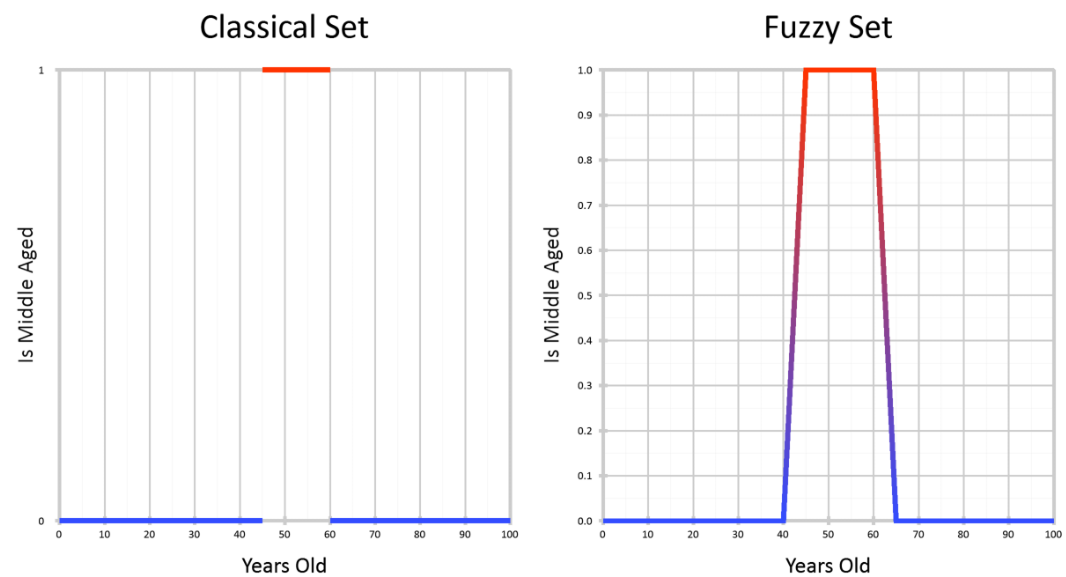


Figure 1. Classical Set vs. Fuzzy Set

## 1. Operations

Classical sets have logical operators such as union (represented using binary operator $\vee$, as in $A \vee B$), intersection (represented using the binary operator $\wedge$, as in $A \wedge B$), inclusion (represented using the binary operator $\leq$, as in $A \leq B$, and complementation (represented using the unary operator $\bar{}$, as in $\bar{A}$. Given that fuzzy sets are an extension of classical sets, these basic connectives still exist but have been more generalized and extended to better address the continuum of membership grades.

**1.1. Triangular Norms and Conorms.** Triangular Norms and Conorms generalize union and intersection, respectively. T-norms (triangular norms) and t-conorms (triangular conorms) fulfill identity, commutativity, associativity, and

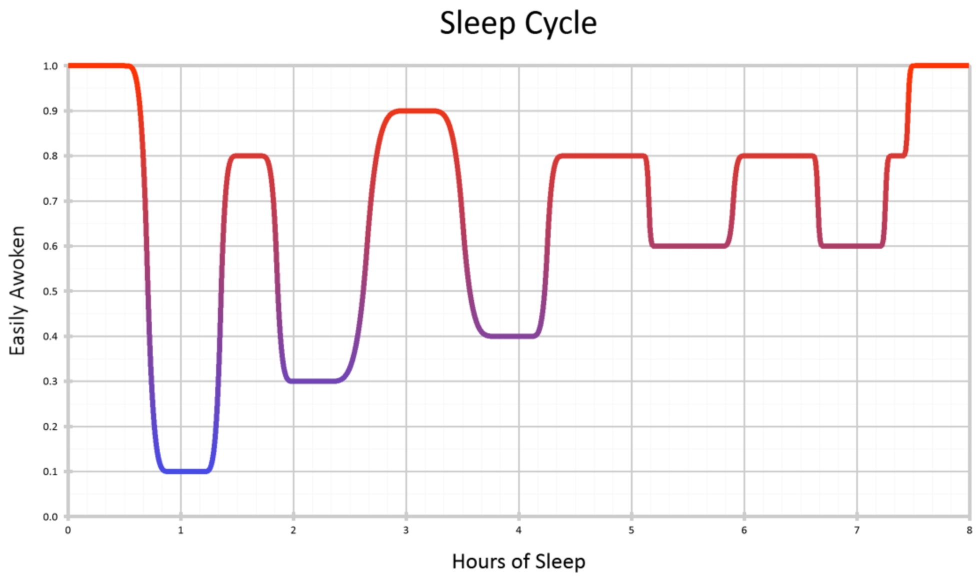


Figure 2. Sleep Cycle as a Non-Trivial Fuzzy Set

monotonicity properties. This means that the following would hold true for any t-norm $T$, and any t-conorm $S$: $xT1 = a$, $xTy = xTy$, $xT(yTz) = (xTy)Tz$, if $x \leq a$ and $y \leq b$ then $xTy \leq aTb$, $xS0 = x$, $xSy = ySx$, $xS(ySz) = (xSy)Sz$, and if $x \leq a$ and $y \leq b$ then $xSy \leq aSb$ (where $x$, $y$, $a$, and $b$ are membership grades). T-norms and t-conorms act as point-wise operations on fuzzy sets; that is, for all $x \in X$, $A\,T\,B = A(x)\,T\,B(x)$ (where $A, B \in F(X)$, and $T$ is any t-norm or t-conorm).

Common t-norms include the Gödel (minimum), the Goguen (product), and the Lukasiewicz t-norms: $min(x, y)$ (also denoted $x \wedge y$), $x \cdot y$, and $max(x + y - 1, 0)$ (also denoted $x \wedge_L y$), respectively (see Fig. 3, Fig. 4, and Fig. 6). Common t-conorms include the maximum, bounded sum, and probabilistic sum t-conorms: $max(x, y)$ (also denoted $x \vee y$), $min(x + y, 1)$, and $a + b - a \cdot b$, respectively (see Fig. 3, Fig. 5, and Fig. 6).

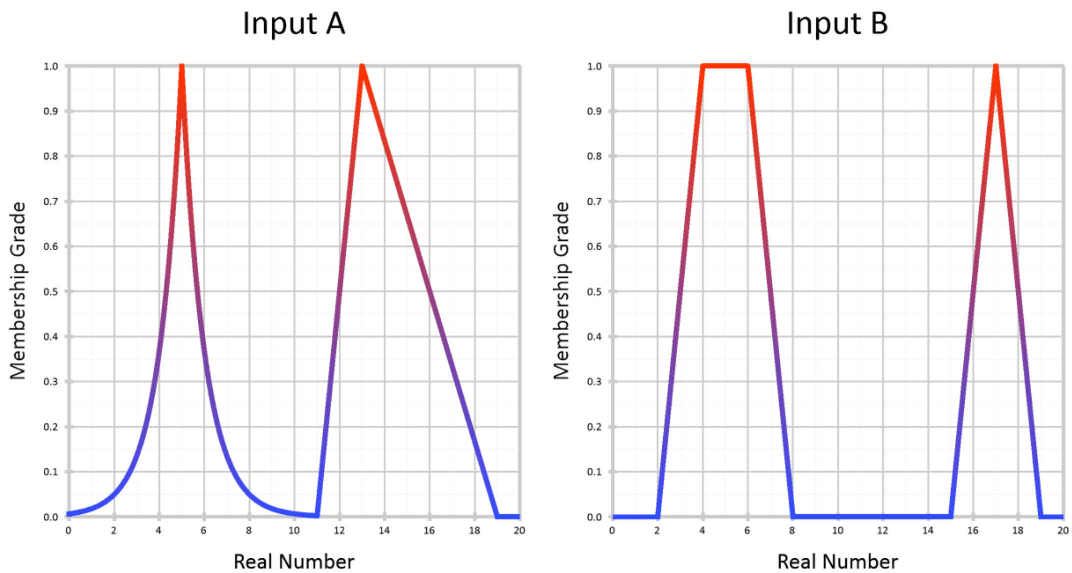


Figure 3. Fuzzy Sets Before Operations

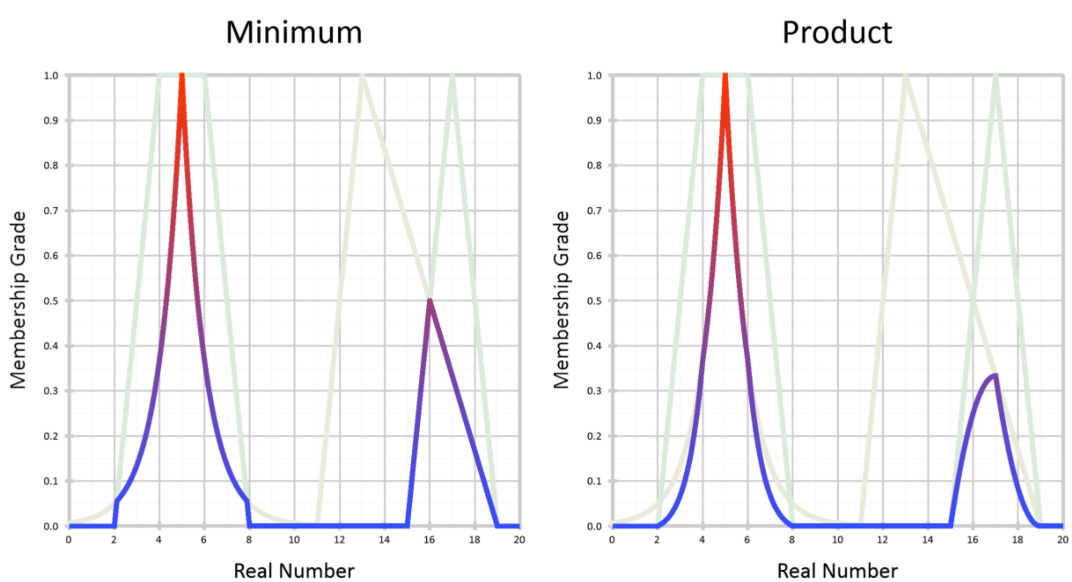


Figure 4. Minimum T-Norm and Product T-Norm

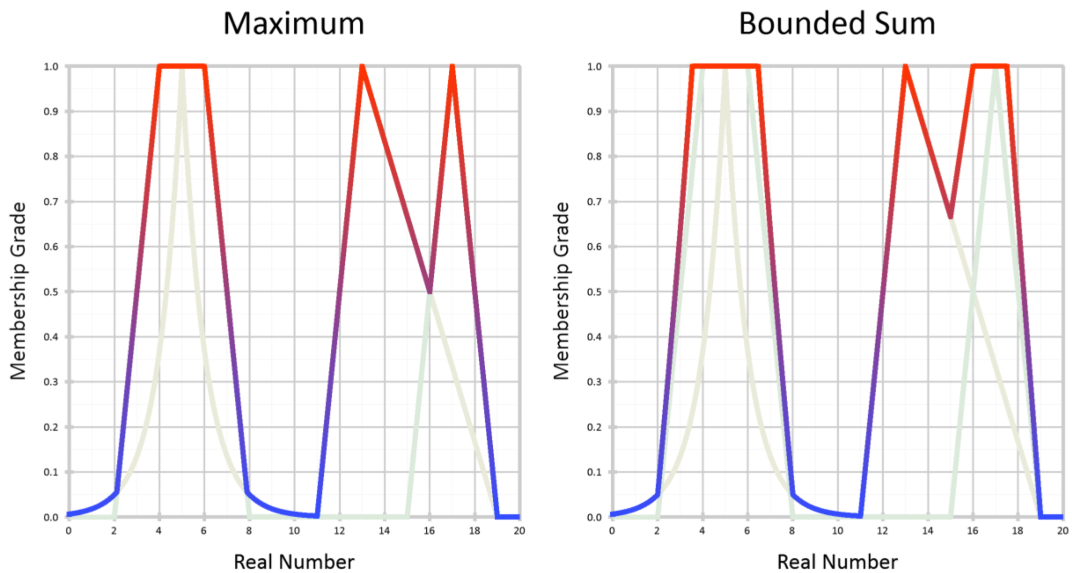


Figure 5. Maximum T-Conorm and Bounded Sum T-Conorm

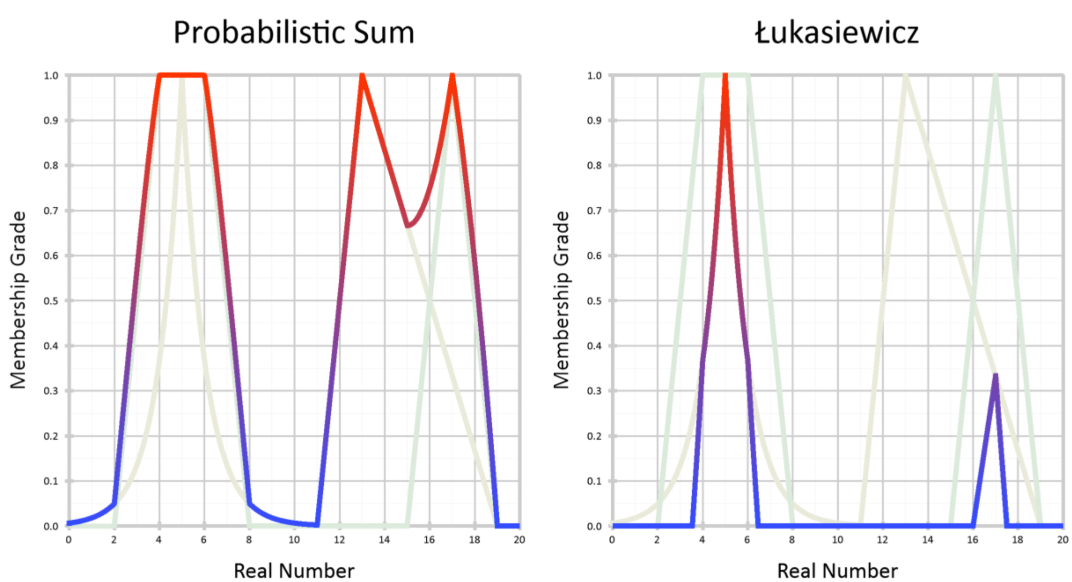


Figure 6. Probabilistic Sum T-Conorm and Lukasiewicz T-Norm

**1.2. Negation.** Complementation is generalized for fuzzy sets using a negation function, $N$. Negation requires that $N$ maps any membership of one to zero, maps any membership of zero to one, and is non-increasing. Negation is considered strict so long as $N$ is continuous and strictly decreases. Negation is considered strong if $N$ is involutive. That is, $N(0) = 1$, $N(1) = 0$, and $x \leq y \Rightarrow N(x) \geq N(y)$ for negation. If $x < y \Rightarrow N(x) > N(y)$ and $N$ is continous, $N$ is a strict negation. If for a strict negation, $N(N(x)) = x$, then $N$ is a strong negation. The most common strong negation is the standard negation, where $N(x) = 1 - x$; another example of strong negation is the $\lambda$-complement, where $\lambda > -1$: $N_\lambda(x) = \frac{1-x}{1+\lambda x}$. The $N$-complement of $A \in F(X)$ is created point-wise using $N(A) = N(A(x))$, for all $x \in X$ (see Fig. 7).

**1.3. Fuzzy Implications.** An extension of classical implications exists in fuzzy logic. A function is considered a fuzzy implication, $\rightarrow$, when it fulfills the following: if $x \leq y$ then $x \rightarrow z \geq y \rightarrow z$, if $y \leq z$ then $x \rightarrow y \leq x \rightarrow z$, $1 \rightarrow 0 = 0$, and $0 \rightarrow 0 = 1 \rightarrow 1 = 1$.

For any given t-norm, $T$, a residual implication can be defined using $x \rightarrow_T y = sup(z|xTz \leq y)$. Some common residual implications include Gödel implication, where $T = x \wedge y$: $x \rightarrow_T y = \begin{cases} 1 & \text{if } x \leq y \\ y & \text{if } x > y \end{cases}$ and Lukasiewicz implication, where $T = max(x + y - 1, 0)$: $x \rightarrow_T y = min(1 - x + y, 1)$ (see Fig. 8 and Fig. 9).

## 2. Fuzzy Numbers

Fuzzy numbers are normal, fuzzy convex, upper semicontinuous, compactly supported fuzzy sets (for examples of sets that are not fuzzy numbers, see Fig. 10 and Fig. 11). They can be especially useful in the context of fuzzy control. Fuzzy numbers in which membership grades are defined by a left function and a right function are referred to as LR Fuzzy Numbers. Singleton fuzzy numbers and Closed Interval fuzzy numbers are the simplest types, containing only membership degrees of 0 and 1 (see Fig. 13). Trapezoidal fuzzy numbers and Triangular fuzzy numbers (triangular fuzzy numbers are just trapezoidal fuzzy numbers where the core consists of only a single point) are the simplest types of fuzzy numbers that actually consist of a continuum of membership grades (see Fig. 12). More complex fuzzy numbers include Gaussian, Exponential, and Sinusoidal functions (see Fig. 14 and Fig. 15). It should be noted that in order for these fuzzy numbers to be compactly supported, their differentiability must sometimes be sacrificed; for this reason, the requirement of compact support is sometimes dropped to produce more appealing results.

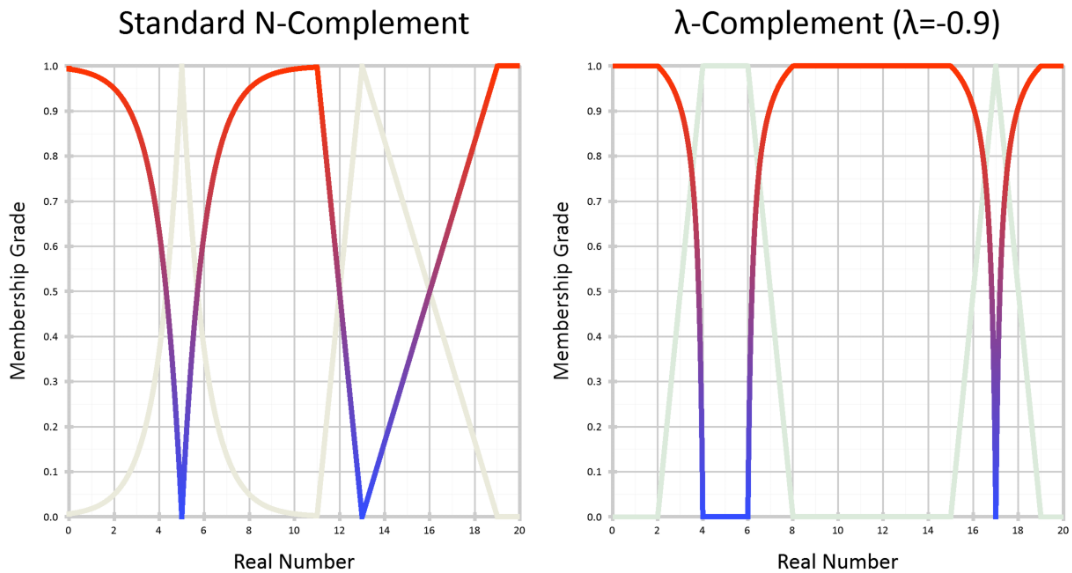


Figure 7. Standard N-Complement and λ-Complement

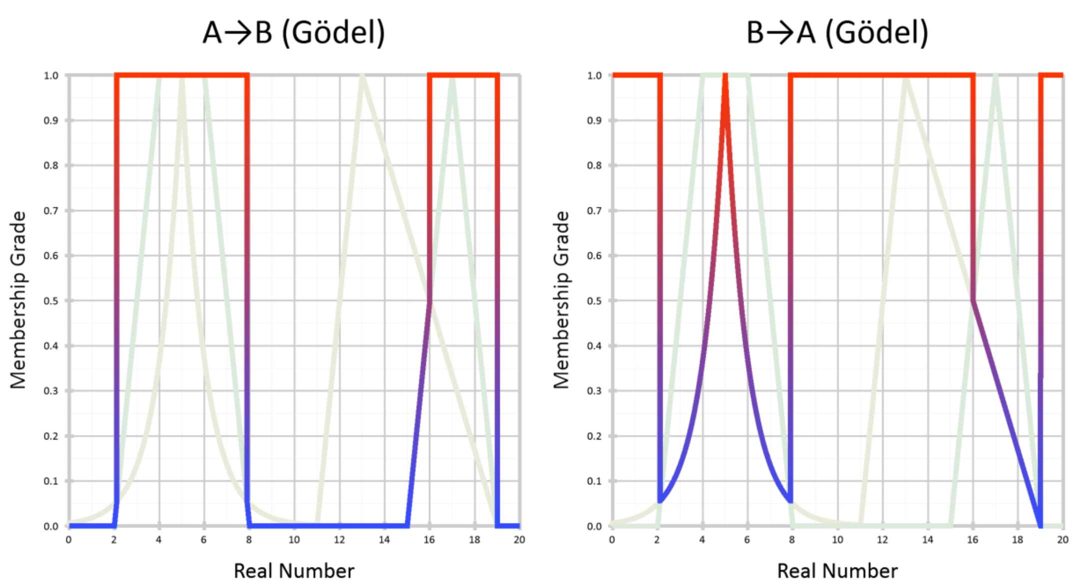


Figure 8. Gödel Implication

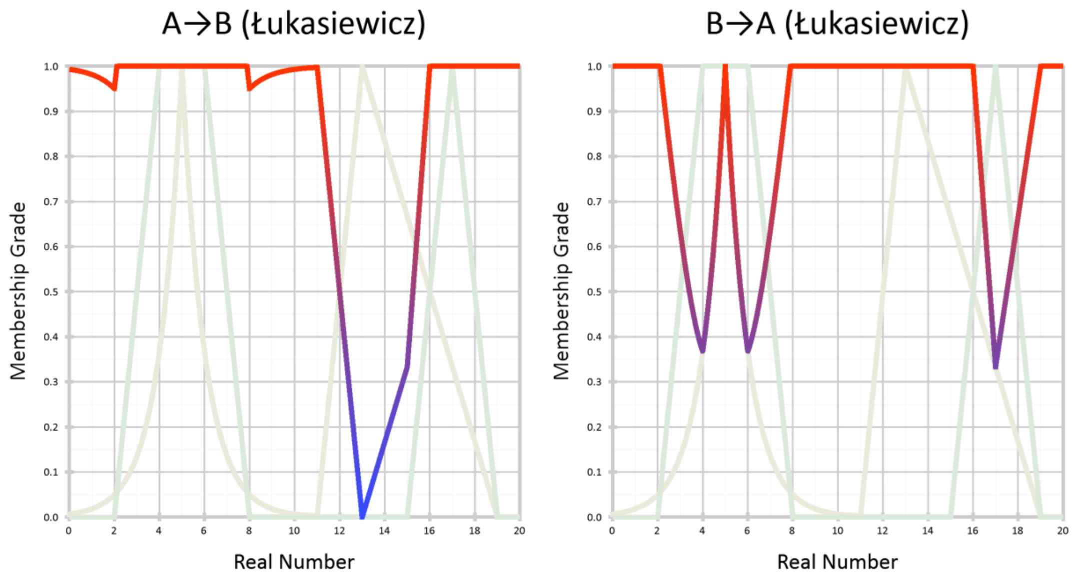


Figure 9. Lukasiewicz Implication

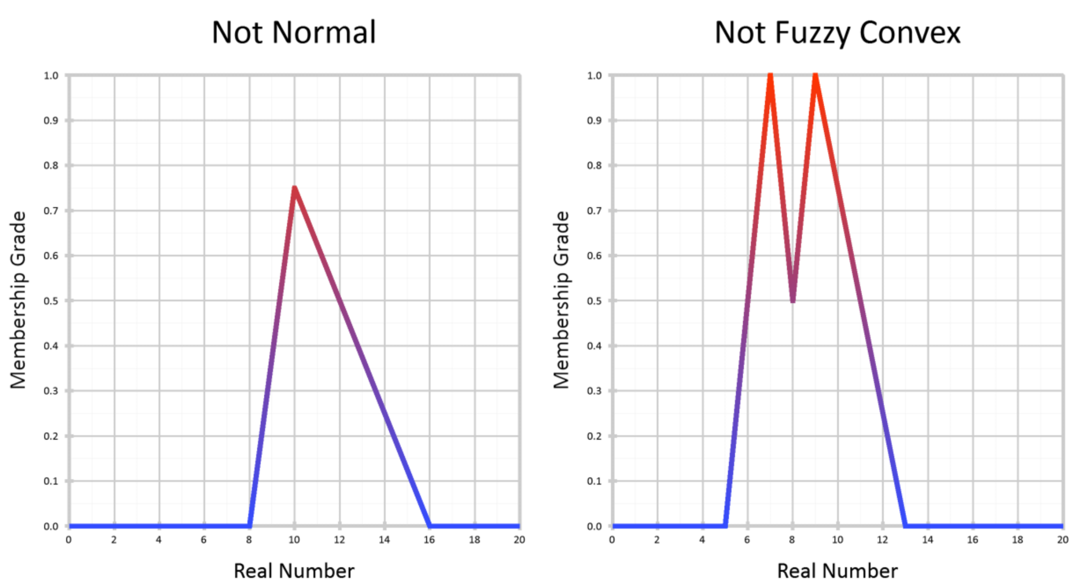


Figure 10. Not Normal and Not Fuzzy Complex Fuzzy Sets

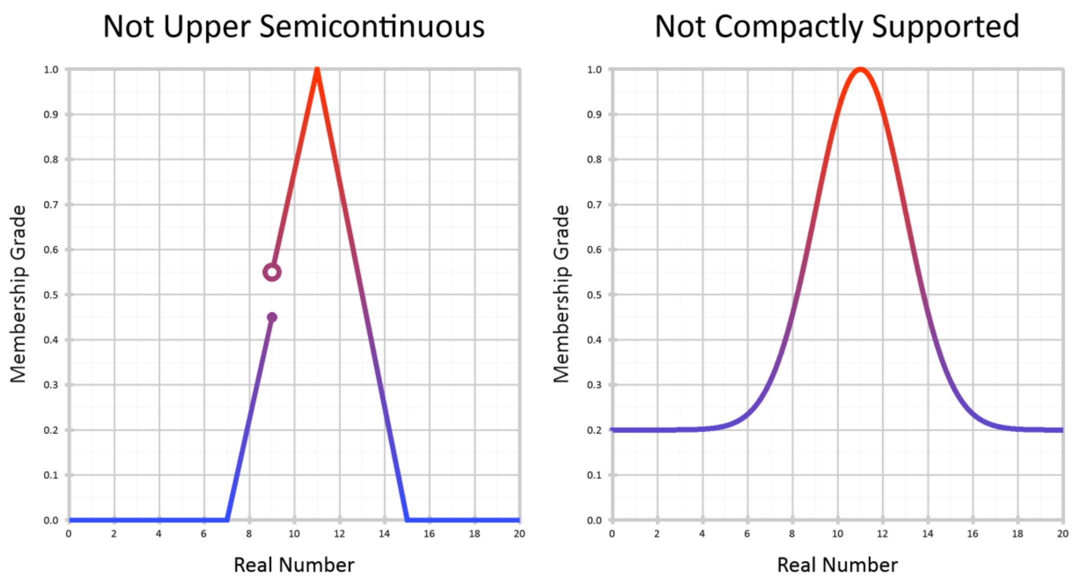


Figure 11. Not Upper Semicontinuous and Not Compactly Supported

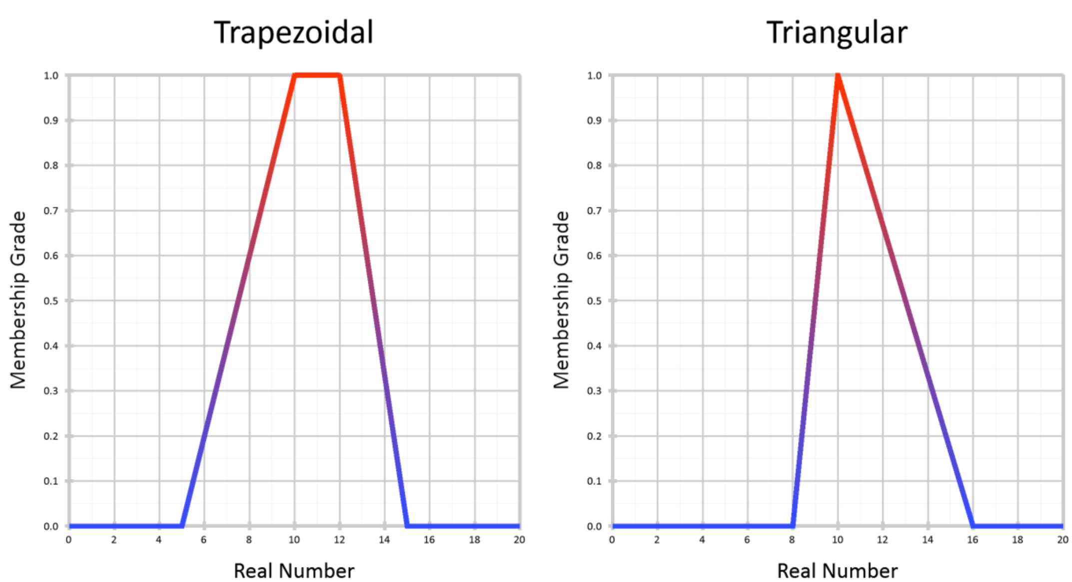


Figure 12. Trapezoidal and Triangular Fuzzy Numbers

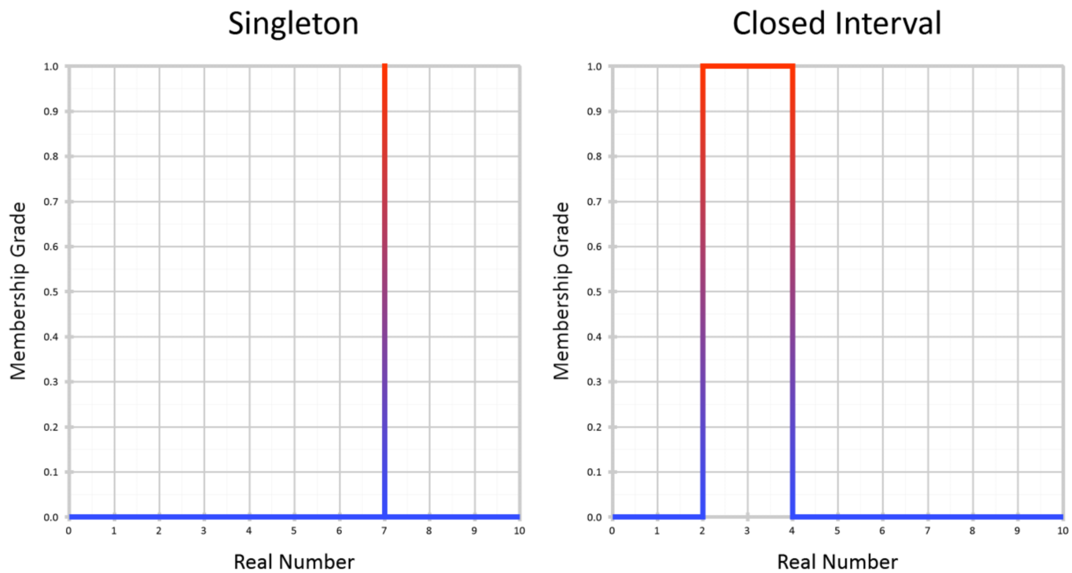


Figure 13. Singleton and Closed Interval Fuzzy Numbers

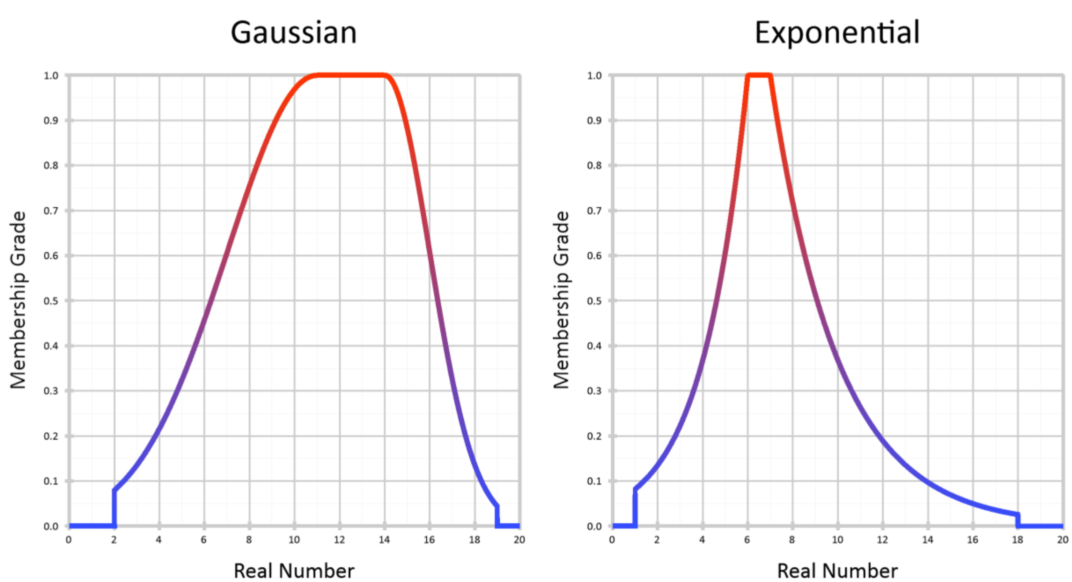


Figure 14. Gaussian and Exponential Fuzzy Numbers

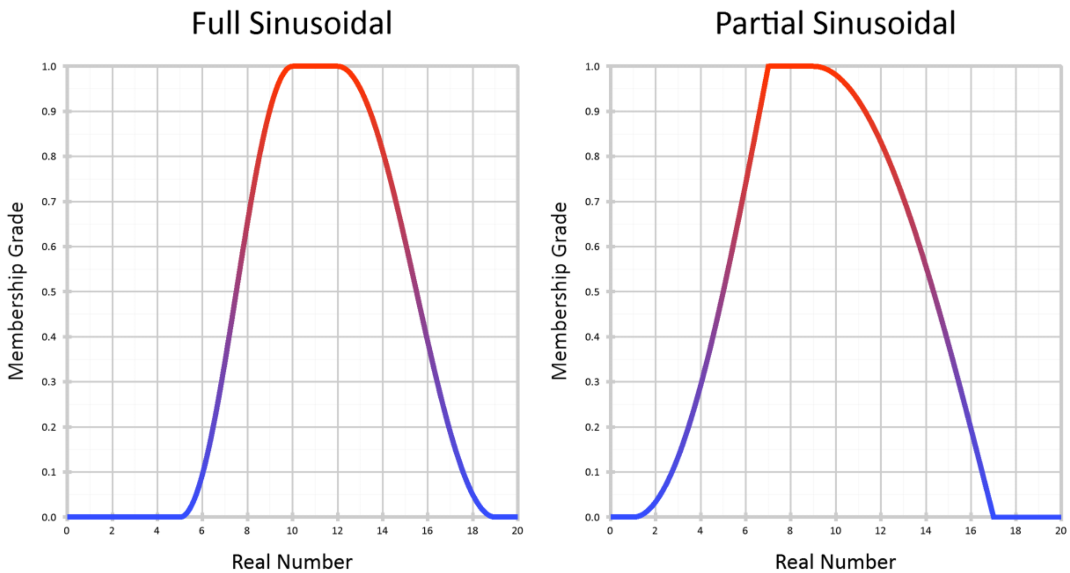


Figure 15. Sinusoidal Fuzzy Numbers

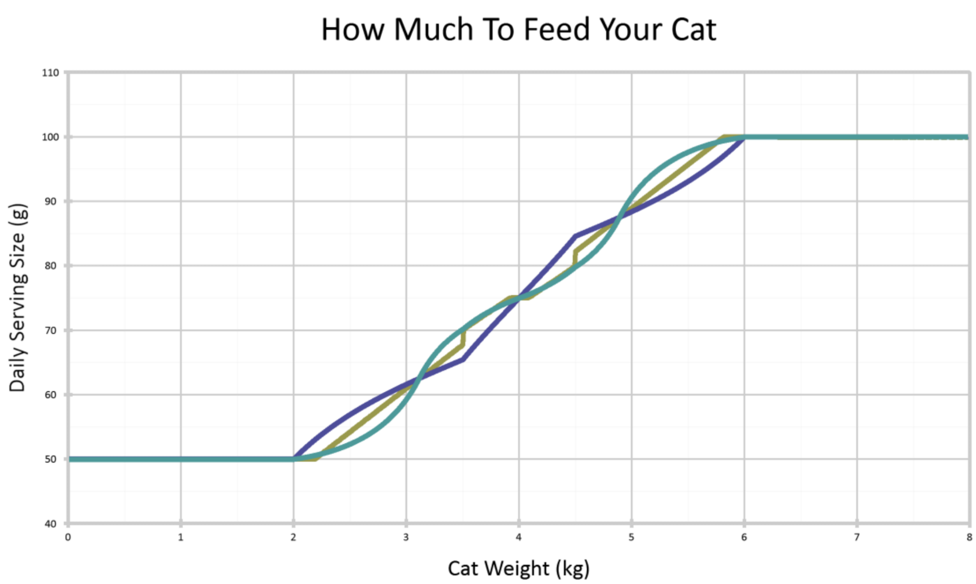


Figure 16. Example SISO Fuzzy Systems

CHAPTER 3

# Fuzzy Controllers

In order to solve problems that lie outside of the domain of fuzzy sets, a single-input, single-output fuzzy system can be employed. A SISO fuzzy system consists of a fuzzifier, a fuzzy inference system using a fuzzy rule base with a fuzzy relation, and a defuzzifier[1]. Fuzzy controllers may be considered the most well-known applications of the theory of Fuzzy Sets and Systems, being used in many practical applications. There are several commercial and non-commercial implementations of fuzzy control systems of Mamdani type or Takagi-Sugeno type [16]. SISO fuzzy systems are called fuzzy controllers when used in control problems. Essentially, a fuzzy controller allows us to take a non-fuzzy problem, convert it into a fuzzy problem, find a fuzzy solution, and then turn that into a non-fuzzy solution.

## 1. Fuzzifiers

A fuzzifier takes crisp (non-fuzzy) input and returns a fuzzy output. The simplest fuzzifier is an inclusion map: where the crisp input, $x_0 \in X$, is mapped to a singleton fuzzy set $x_0$ using the characteristic function:

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

The characteristic function does not need to result in a singleton fuzzy set, however; the mapping can result in any valid membership grade. Using a characteristic function that results in a singleton fuzzy set can greatly reduce the computational complexity of the SISO fuzzy system, so it was chosen for the purposes of applications covered here.

## 2. Fuzzy Inference Systems

After being mapped to a fuzzy set, a fuzzy inference system can then be used. Fuzzy inference systems consist of a antecedents, consequents, and a fuzzy rule base using a fuzzy relation to infer between antecedents and consequents.

**2.1. Fuzzy Rules.** The fuzzy rule "If $x$ is $A$ then $y$ is $B$" is defined as a fuzzy relation using the following:

Mamdani Rule:

$$r_M(x, y) = A(x) \wedge B(y)$$

Larsen Rule:

$$r_L(x, y) = A(x) \cdot B(y)$$

T-Norm Rule (where $T$ is a t-norm):

$$r_T(x, y) = A(x) T B(y)$$

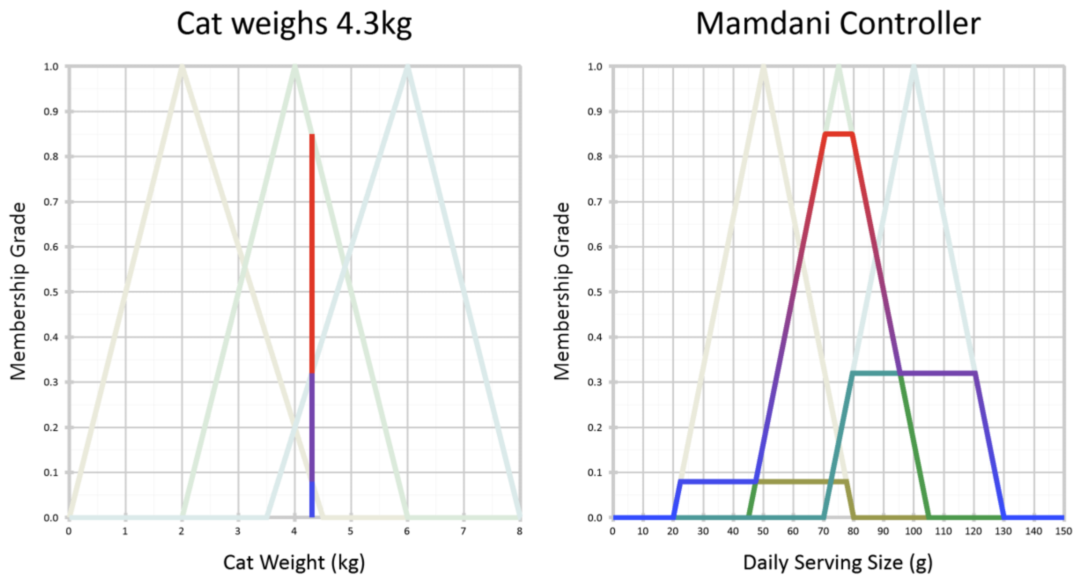


Figure 17. Fuzzy Controller using Mamdani Rule Base

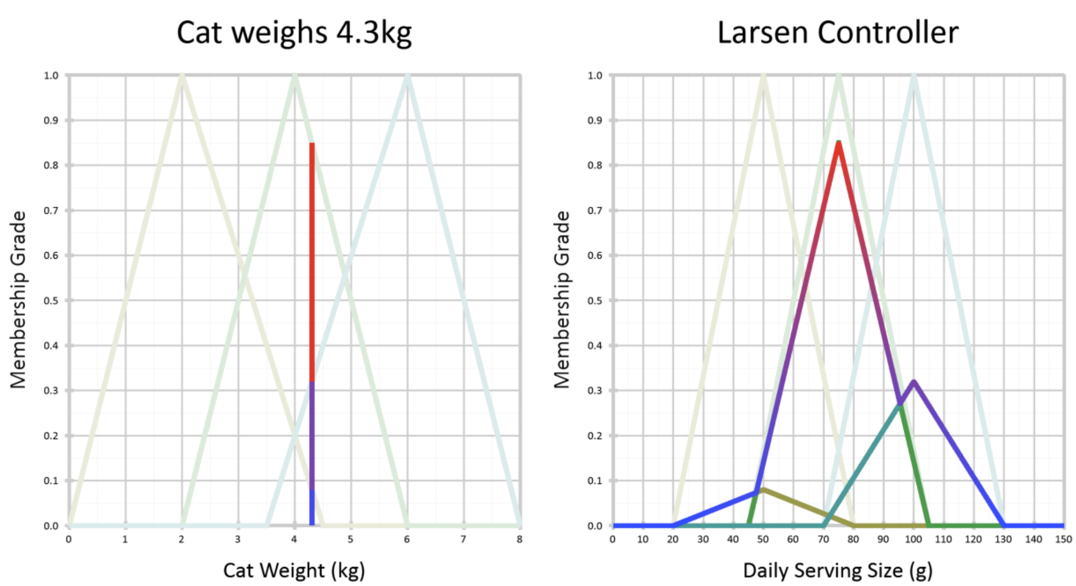


Figure 18. Fuzzy Controller using Larsen Rule Base


Figure 19. Fuzzy Controller using T-Norm Rule Base

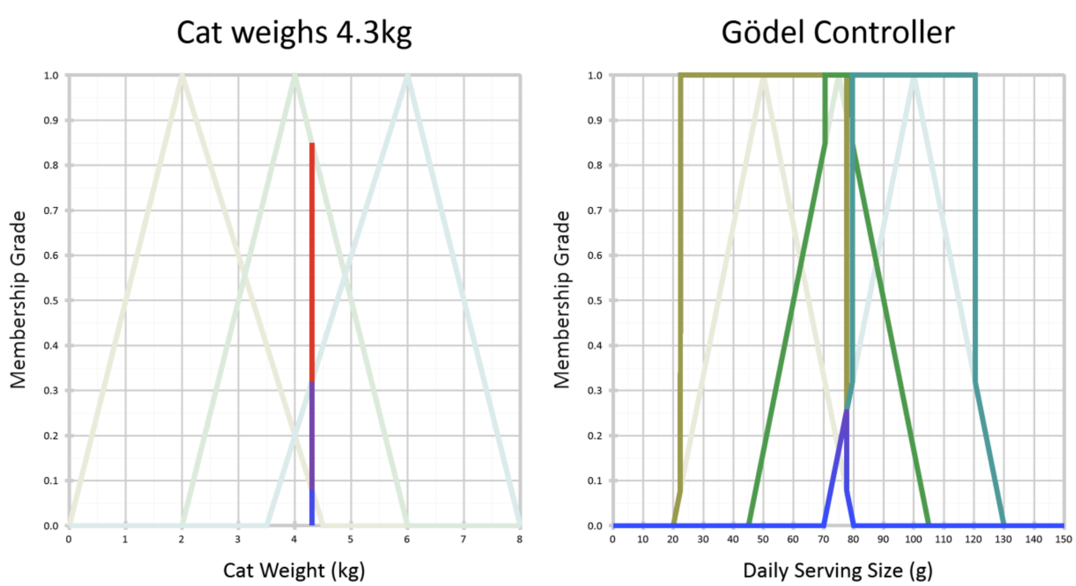


Figure 20. Fuzzy Controller using Gödel Rule Base

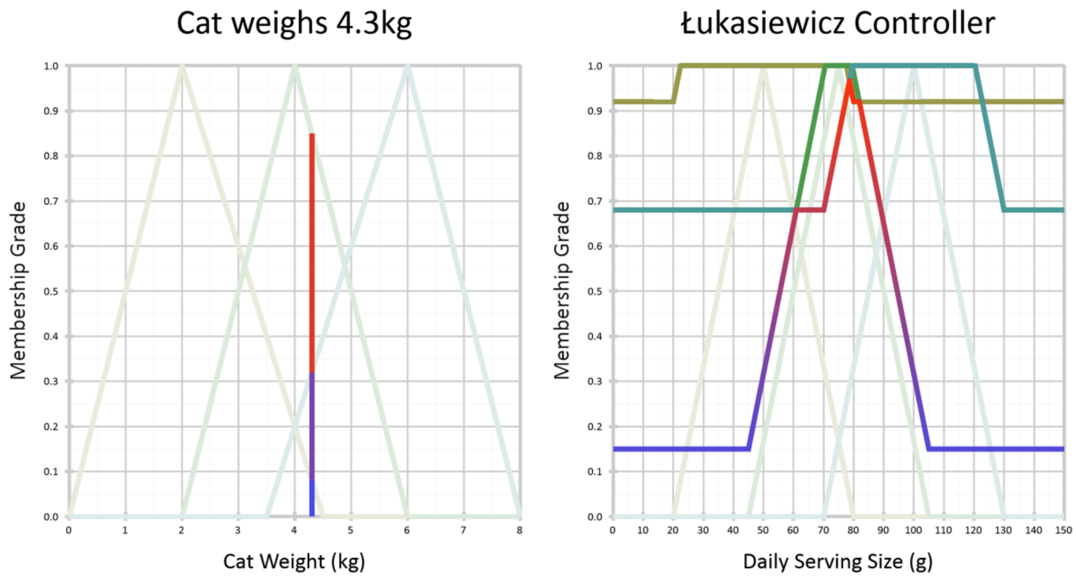


Figure 21. Fuzzy Controller using Gödel Residual Rule Base

Gödel Rule (where $\rightarrow$ is Gödel implication):

$$r_G(x, y) = A(x) \rightarrow B(y)$$

Gödel Residual Rule (where $\rightarrow_T$ is a residual implication with a given t-norm):

$$r_R(x, y) = A(x) \rightarrow_T B(y)$$

**2.2. Fuzzy Rule Bases.** The fuzzy rule base "If $x$ is $A_i$ then $y$ is $B_i$, $i = 1, \ldots, n$" can then be defined as a fuzzy relation using the following:

Mamdani Rule Base:

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

Larsen Rule Base:

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

Max-T-Norm Rule Base (where $T$ is a t-norm):

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

Gödel Rule Base (where $\rightarrow$ is Gödel implication):

$$R_G(x,y) = \bigwedge_{i=1}^{n} \left( A_i(x) \rightarrow B_i(y) \right.$$

Gödel Residual Rule Base (where $\rightarrow_T$ is a residual implication with):

$$R_R(x,y) = \bigwedge_{i=1}^{n} \left( A_i(x) \rightarrow_T B_i(y) \right.$$

**2.3. Fuzzy Inference Systems.** The fuzzy inference system will interpret a fuzzy rule base using a fuzzy relation, $R(x,y)$. There are many types of fuzzy inference systems:

Mamdani Inference:

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

Larsen Inference:

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

T-Norm-Based Inference (where $T$ is a t-norm), also known as Generalized Modus Ponens Inference:

$$B'(y) = A' \circ_T R(x,y) = \bigvee_{x \epsilon X} A'(x) T R(x,y)$$

Gödel Inference (where $\rightarrow$ is Gödel implication):

$$B'(y) = A' \triangleleft R(x,y) = \bigwedge_{x \epsilon X} \left( A'(x) \rightarrow R(x,y) \right.$$

Gödel Residual Inference (where $\to_T$ is a residual implication):

$$B'(y) = A' \triangleleft_T R(x,y) = \bigwedge_{x \epsilon X} \Big( A'(x) \to_T R(x,y)$$

When the inclusion map fuzzifier is used, all Fuzzy Inference Systems result in the same output (see [1]):

$$B'(y) = R(x_0, y)$$

For example, using the Mamdani Rule Base:

$$B'(y) = \bigvee_{i=1}^{n} A_i(x_0) \wedge B_i(y)$$

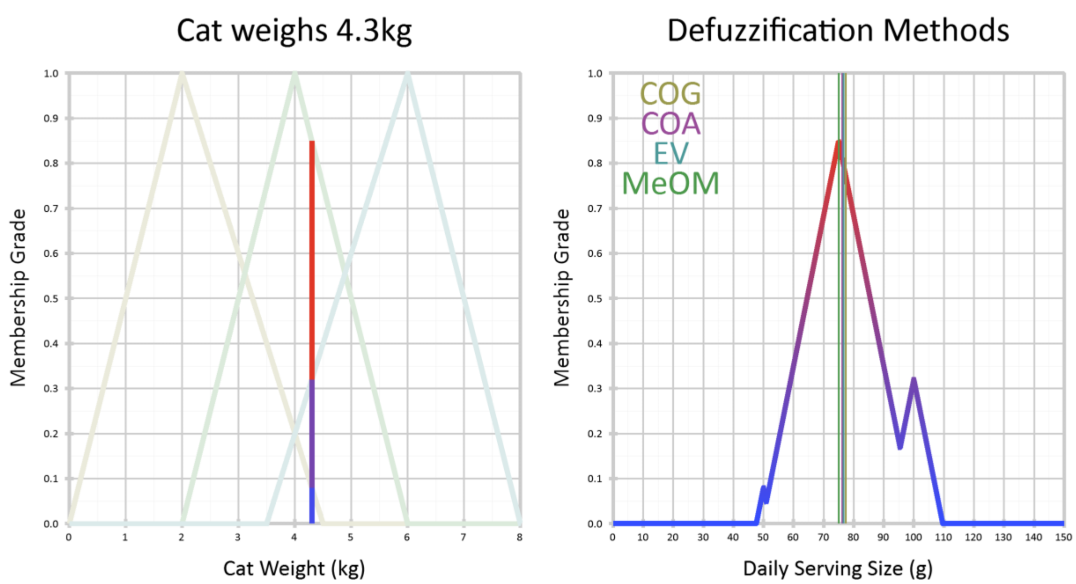


Figure 22. Defuzzification Methods

## 3. Defuzzifiers

The defuzzifier takes the fuzzy set, $u$, obtained from the fuzzy inference system as input and returns a crisp value as output. There are many defuzzifiers, some examples include:

Center of Gravity (where $W = supp(u)$):

$$COG(u) = \frac{\int_W x \cdot u(x)dx}{\int_W u(x)dx}$$

Center of Area (if $u \in F(\mathbb{R})$):

$$COA(u) = a \text{ where } \int_{-\infty}^{a} u(x)dx = \int_{a}^{\infty} u(x)dx$$

Expected Value, if $u$ is a continuous fuzzy number:

$$EV(u) = \frac{1}{2}\int_0^1 u_r^+ + u_r^- dr$$

Mean of Maxima (where $U = x \in X|u(x) = max_{t \in X} u(t)$):

$$MeOM(u) = \frac{\int_{x \in U} xdx}{\int_{x \in U} dx}$$

CHAPTER 4

# LU Representation

As described in [20]: fuzzy numbers appear as typical antecedents and consequents in the fuzzy inference systems that are at the core of a fuzzy control system. Fuzzy numbers are normal, fuzzy convex, upper semicontinuous, compactly supported fuzzy sets. The Lower-Upper (LU) representation of a fuzzy number is based on the well known Negoita-Ralescu and Goetschel-Voxman representation theorems (see [1]), stating essentially that the $\alpha$-cut form of a fuzzy number $u$ is equivalent to the description of a fuzzy set via its membership function; $\alpha$-cuts ($[u]_\alpha$) are calculated using:

$$[u]_\alpha = \begin{cases} x|u(x) \geq \alpha & \text{if } \alpha > 0 \\ cl(x)|u(x) > 0 & \text{if } \alpha = 0 \end{cases}.$$

In particular, $\alpha$-cuts can be used to uniquely represent a fuzzy number $[u]_\alpha = [u_\alpha^-, u_\alpha^+]$ , given that $\alpha \to u_\alpha^-$ and $\alpha \to u_\alpha^+$ are left-continuous for all $\alpha \in (0, 1]$, right-continuous when $\alpha = 0$, monotonically increasing for $u_\alpha^-$, monotonically decreasing for $u_\alpha^+$, and $u_\alpha^- \leq u_\alpha^+$ when $\alpha = 1$. For this section, $u$ is assumed to be a fuzzy number with $\alpha$-cuts $[u]_\alpha = [u_\alpha^-, u_\alpha^+]$ and $\alpha \longrightarrow u_\alpha^-$, $\alpha \longrightarrow u_\alpha^+$ monotonic, continuous and differentiable with respect to $\alpha$ (see Figure 23).

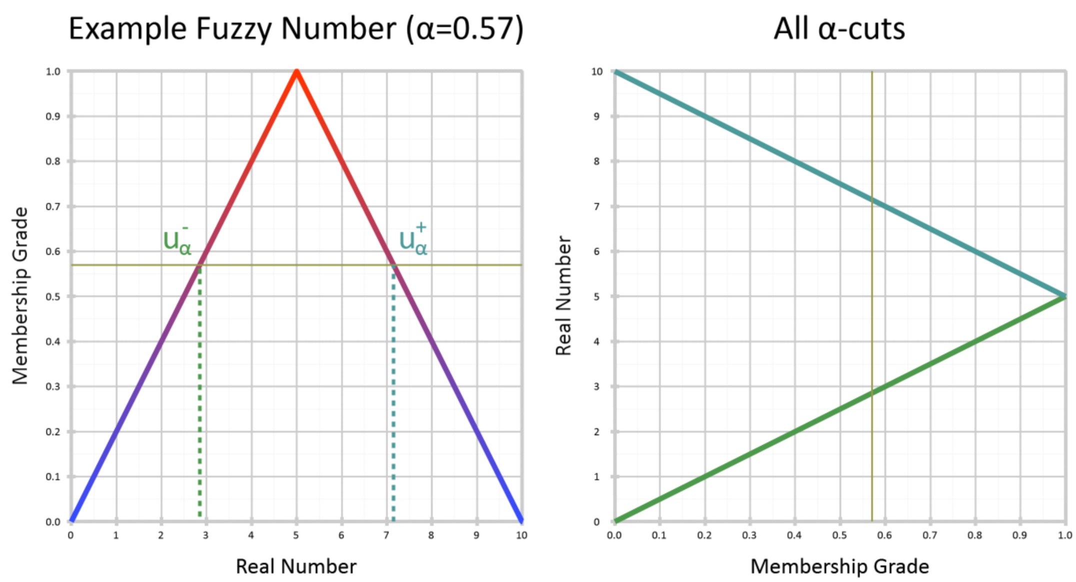


Figure 23. Example of $\alpha$-cuts

### 1. LU-Parametric Representation

For $\alpha \in [0, 1]$, let $\delta u_\alpha^-$ and $\delta u_\alpha^+$ denote the first derivatives of $u_\alpha^-$ and $u_\alpha^+$ with respect to $\alpha$ (using right-derivatives for $\alpha = 0$ and left-derivatives for $\alpha = 1$). The LU-parametric representation of fuzzy numbers proposed in [14, 15] is shown to have a great application potential in the area of fuzzy arithmetic and fuzzy calculus. Some results shown in the above cited papers are as follows:

Consider a family of standardized differentiable and increasing shape functions $p : [0, 1] \longrightarrow [0, 1]$, depending on two parameters $\beta_0, \beta_1 \geq 0$ such that

1. $p(0) = 0$, $p(1) = 1$,

2. $p'(0) = \beta_0, p'(1) = \beta_1$ and

3. $p(t)$ is increasing on $[0, 1]$ if and only if $\beta_0, \beta_1 \geq 0$.

Consider the following rational splines as examples of valid shape functions (see Figure 24):

$$p(t; \beta_0, \beta_1) = \frac{t^2 + \beta_0 t(1 - t)}{1 + (\beta_0 + \beta_1 - 2)t(1 - t)}$$

or

$$p(t; \beta_0, \beta_1) = \frac{t^3 + \beta_0 t(1 - t)^2 + \beta_0 t^2(1 - t)}{1 + (\beta_0 + \beta_1 - 3)t(1 - t)}.$$

These rational splines can be adopted to represent the functions $u_\alpha^-$ and $u_\alpha^+$ as

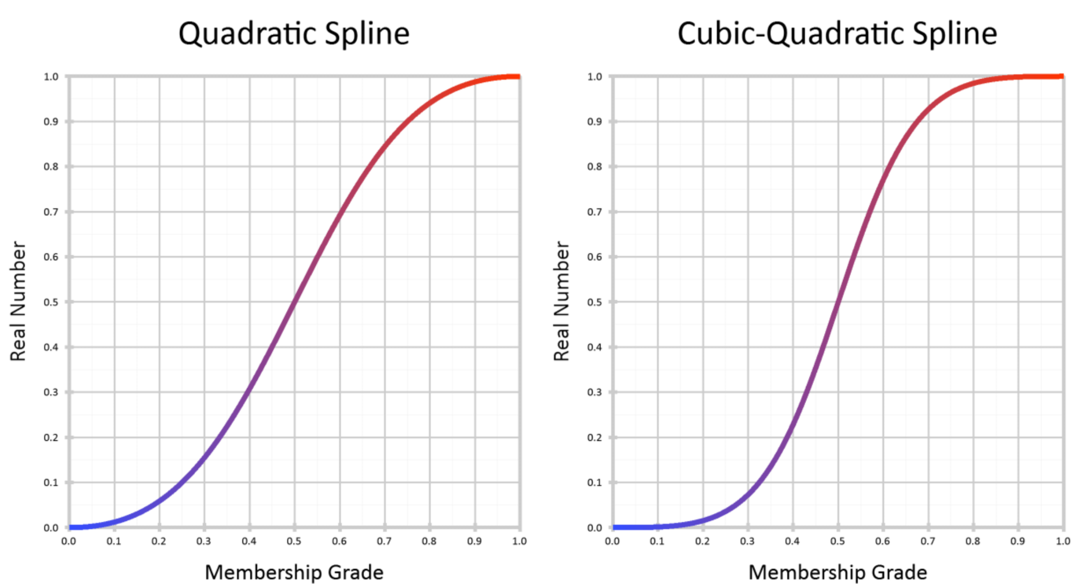


Figure 24. Rational Splines with Parameters of 0

piecewise differentiable, on a decomposition of the interval $[0, 1]$ into $N$ subintervals $0 = \alpha_0 < \alpha_1 < \ldots < \alpha_{i-1} < \alpha_i < \ldots < \alpha_N = 1$. At the extremal points of each subinterval $I_i = [\alpha_{i-1}, \alpha_i]$, the values and the first derivatives (slopes) of the two functions are given

$$u^{-}_{(\alpha_{i-1})} = u^{-}_{0,i}, u^{+}_{(\alpha_{i-1})} = u^{+}_{0,i}, u^{-}_{(\alpha_i)} = u^{-}_{1,i}, u^{+}_{(\alpha_i)} = u^{+}_{1,i} \tag{4.1}$$

$$u'^{-}_{(\alpha_{i-1})} = d^{-}_{0,i}, u = d^{+}_{0,i}, u'^{-}_{(\alpha_i)} = d^{-}_{1,i}, u'^{+}_{(\alpha_i)} = d^{+}_{1,i} \tag{4.2}$$

and by the transformation $t_\alpha = \frac{\alpha-\alpha_{i-1}}{\alpha_i-\alpha_{i-1}}, \alpha \in I_i,$ each subinterval $I_i$ is mapped into $[0,1]$ to determine each piece independently. Using this, the fuzzy number and its shape can be controlled. Let $p_i^{\pm}(t)$ denote the model functions on $I_i$; for example, $p_i^-(t) = p(t;\beta^-_{0,i}, \beta^-_{1,i})$, $p_i^+(t) = p(t;\beta^+_{0,i}, \beta^+_{1,i})$ is obtained with $\beta^-_{j,i} = \frac{\alpha_i-\alpha_{i-1}}{u^-_{1,i}-u^-_{0,i}} d^-_{j,i}$ and $\beta^+_{j,i} = -\frac{\alpha_i-\alpha_{i-1}}{u^+_{1,i}-u^+_{0,i}} d^+_{j,i}$ for $j = 0, 1$ so that, for $\alpha \in [\alpha_{i-1}, \alpha_i]$ and $i = 1, 2, ..., N$:

$$\left.\begin{aligned} u^-_\alpha &= u^-_{0,i} + (u^-_{1,i} - u^-_{0,i}) p^-_i\left(t_\alpha;\beta^-_0, \beta^-_1\right) & (4.3)\\ u^+_\alpha &= u^+_0 + (u^+_{1,i} - u^+_{0,i}) p^+_i\left(t_\alpha;\beta^+_{0,i}, \beta^+_{1,i}\right) & (4.4)\end{aligned}\right\}$$

A fuzzy number with differentiable lower and upper functions is obtained by taking the values and the slopes appropriately, i.e. $u^-_{1,i} = u^-_{0,i+1} =: u^-_i$, $u^+_{1,i} = u^+_{0,i+1} =: u^+_i$ and, for the slopes, $d^-_{1,i} = d^-_{0,i+1} =: \delta u^-_i$, $d^+_{1,i} = d^+_{0,i+1} =: \delta u^+_i$. This requires $4(N+1)$ parameters, with $N \geq 1$,

$$u = (\alpha_i; u^-_i, \delta u^-_i, u^+_i, \delta u^+_i)_{i=0,1,\ldots,N} \text{ with} \tag{4.5}$$

$$u^-_0 \leq u^-_1 \leq ... \leq u^-_N \leq u^+_N \leq u^+_{N-1} \leq ... \leq u^+_0 \tag{4.6}$$

$$\delta u^-_i \geq 0, \delta u^+_i \leq 0, i = 0, 1, ..., N \tag{4.7}$$

and the branches are computed according to (4.3)-(4.4).

The parameters $\delta u^-_i, \delta u^+_i$ are used to control the shape of the fuzzy numbers under consideration. These can be defined by the user, allowing, together with the values $u_i^-, u_i^+$, for a flexible specification of the shapes of the fuzzy numbers being considered as antecedents or consequents for a given fuzzy system.

An important particular case is obtained for $N = 1$ and it can be used for illustration (see Fig. 25). The fuzzy number can be described by 8 parameters as $u = (u_0^-, \delta u_0^-, u_0^+, \delta u_0^+, u_1^-, \delta u_1^-, u_1^+, \delta u_1^+,)$ with

$$u_\alpha^- = u_0^- + (u_1^- - u_0^-)p^-\left(\alpha; \beta_0^-, \beta_1^-\right) \tag{4.8}$$

$$u_\alpha^+ = u_0^+ + (u_1^+ - u_0^+)p^+\left(\alpha; \beta_0^+, \beta_1^+\right) \tag{4.9}$$

and $\beta_0^- = \frac{1}{u_1^- - u_0^-}\delta u_0^-$, $\beta_1^- = \frac{1}{u_1^- - u_0^-}\delta u_1^-$, $\beta_0^+ = -\frac{1}{u_1^+ - u_0^+}\delta u_0^+$, $\beta_1^+ = -\frac{1}{u_1^+ - u_0^+}\delta u_1^+$.

The parameters $u_0^-, \delta u_0^-, u_0^+, \delta u_0^+, u_1^-, \delta u_1^-, u_1^+, \delta u_1^+$ determine the shape of the fuzzy number $u$. The values $u_0^-, u_0^+, u_1^-, u_1^+$ determine the endpoints of the 0 and 1 level sets, while $\delta u_0^-, \delta u_0^+, \delta u_1^-, \delta u_1^+$ determine the shape of the constructed fuzzy numbers. For example $\delta u_0^- = 0, \delta u_0^+ = 0, \delta u_1^- = 0, \delta u_1^+ = 0$ gives us a fuzzy set that has horizontal tangent at the endpoints of the 0 and 1 level sets, while e.g., $\delta u_0^- = \delta u_1^- = \frac{1}{u_1^- - u_0^-}$ and $\delta u_0^+ = \delta u_1^+ = \frac{-1}{u_0^+ - u_1^+}$ will give triangular numbers.

In this setting, a very general, consistent fuzzy arithmetic was developed in [14]. Using $\widetilde{\mathbb{F}}_N$ to denote the set of all the LU-fuzzy numbers of the form (4.5) over the same uniform decomposition with $N$ subintervals. Structuring $\widetilde{\mathbb{F}}_N$ can be accomplished using addition, +, and a scalar multiplication, ·. Let $u, v \in \widetilde{\mathbb{F}}_N$ be two LU-fuzzy numbers

$$u = (\alpha_i; u_i^-, \delta u_i^-, u_i^+, \delta u_i^+)_{i=0,1,\ldots,N}$$

$$v = (\alpha_i; v_i^-, \delta v_i^-, v_i^+, \delta v_i^+)_{i=0,1,\ldots,N}.$$

Then, there is

$$u + v = (\alpha_i; u_i^- + v_i^-, \delta u_i^- + \delta v_i^-; u_i^+ + v_i^+, \delta u_i^+ + \delta v_i^+)$$

$$k \cdot u = (\alpha_i; ku_i^-, k\delta u_i^-, ku_i^+, k\delta u_i^+)_{i=0,1,\ldots,N} \text{ if } k \geq 0$$

$$k \cdot u = (\alpha_i; ku_i^+, k\delta u_i^+, ku_i^-, k\delta u_i^-)_{i=0,1,\ldots,N} \text{ if } k < 0,$$

where $i = 0, 1, ..., N$. Addition $u + v$ in LU-parametric form is exact at the points $\alpha_i, i = 0, ..., N$ of the decomposition, up to the first derivative of the shape functions. It is an approximation in any other point $\alpha \in [0, 1]$ (see [14, 15]).

## 2. Non-Parametric LU Representation

While the parametric representation allows for any shape to be approximated, cases where $N > 1$ can be computationally expensive and potentially give undesirable results concerning smoothness (compared to gaussian and exponential LR fuzzy numbers). For this reason, it may be more adventageous to use a generalized LU representation in which the lower and upper functions are not strictly defined using rational splines, but instead use the inverse functions of those defined for LR fuzzy numbers (see Fig. 26, Fig. 27, Fig. 28, and Fig. 29).

In the same way that exponential and gaussian LR fuzzy numbers can give smooth results, the same rational splines used for LU parametric representation can potentially be used to create LR fuzzy numbers where $N = 1$ (see Fig. 30). If the inverses of these rational splines are used, identically smooth results could then be achieved with non-parametric LU representation (see Fig. 31).

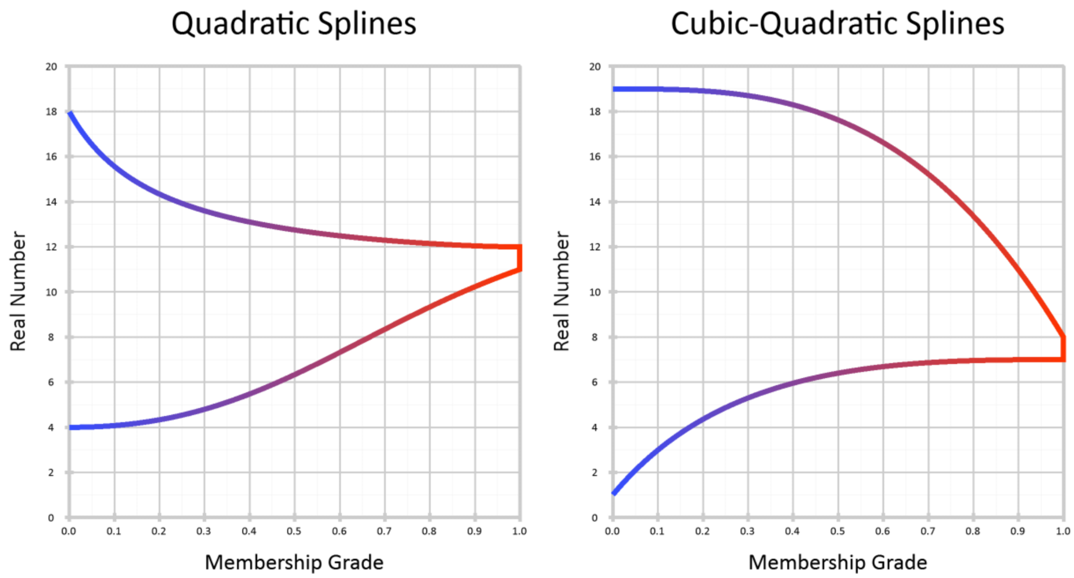


Figure 25. LU Parametric Fuzzy Numbers where $N = 1$

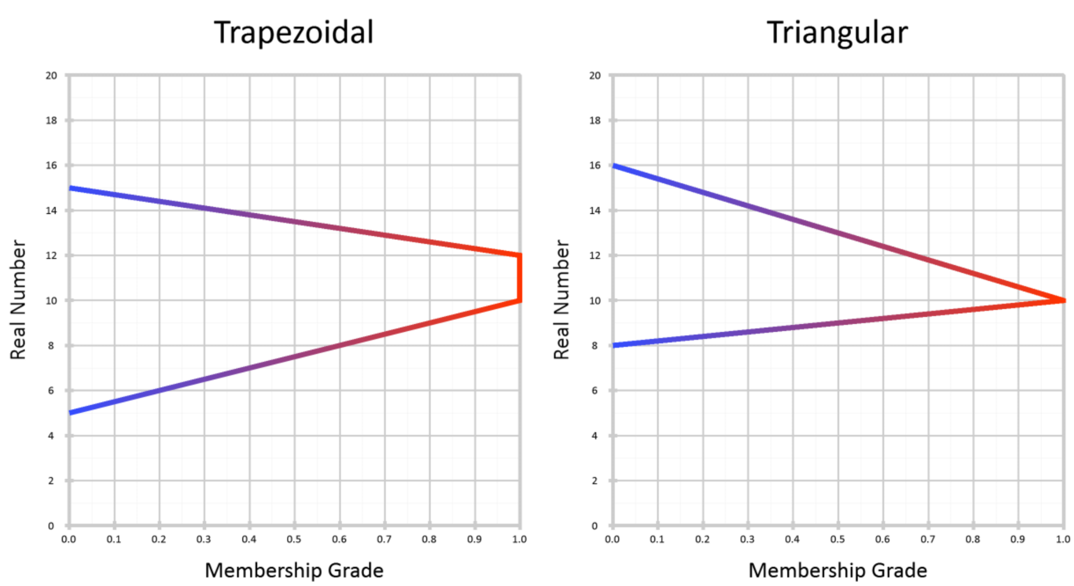


Figure 26. Trapezoidal and Triangular LU Fuzzy Numbers

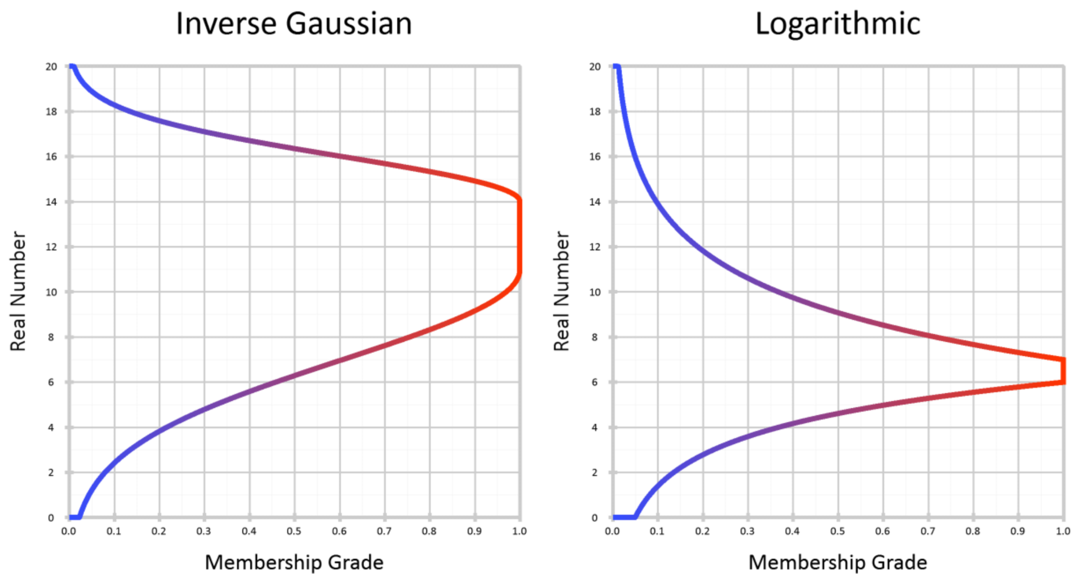


Figure 27. Inverse Gaussian and Logarithmic LU Fuzzy Numbers

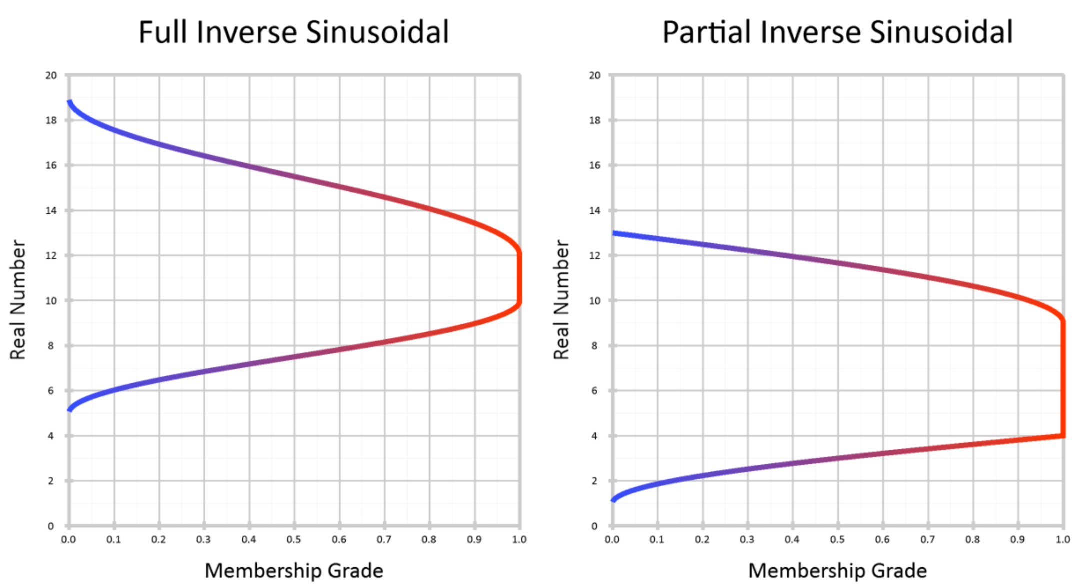


Figure 28. Inverse Sinusoidal LU Fuzzy Numbers

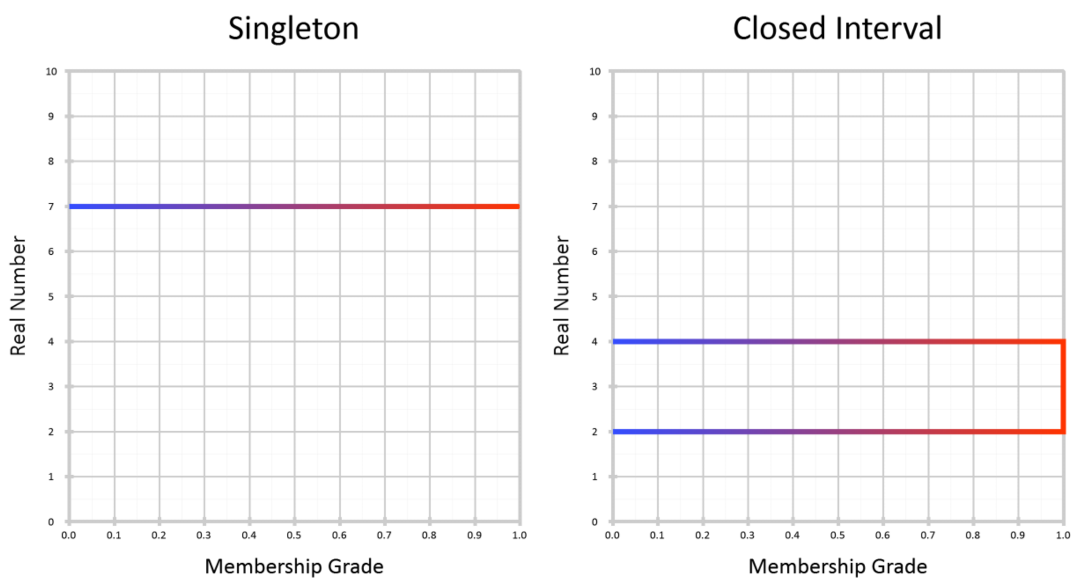


Figure 29. Singleton and Closed Interval LU Fuzzy Numbers

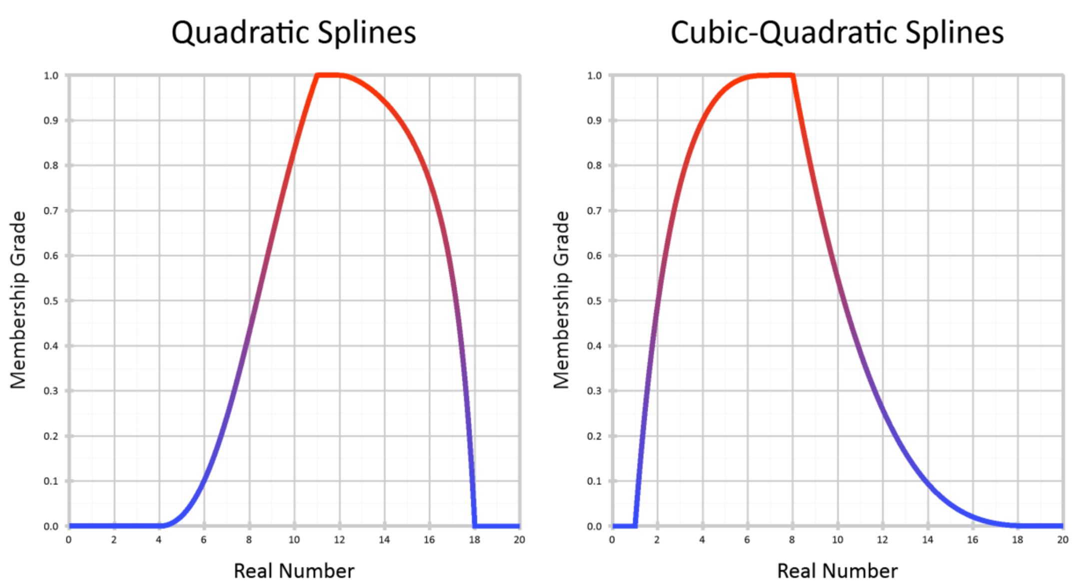


Figure 30. LR Fuzzy Numbers Using Rational Splines

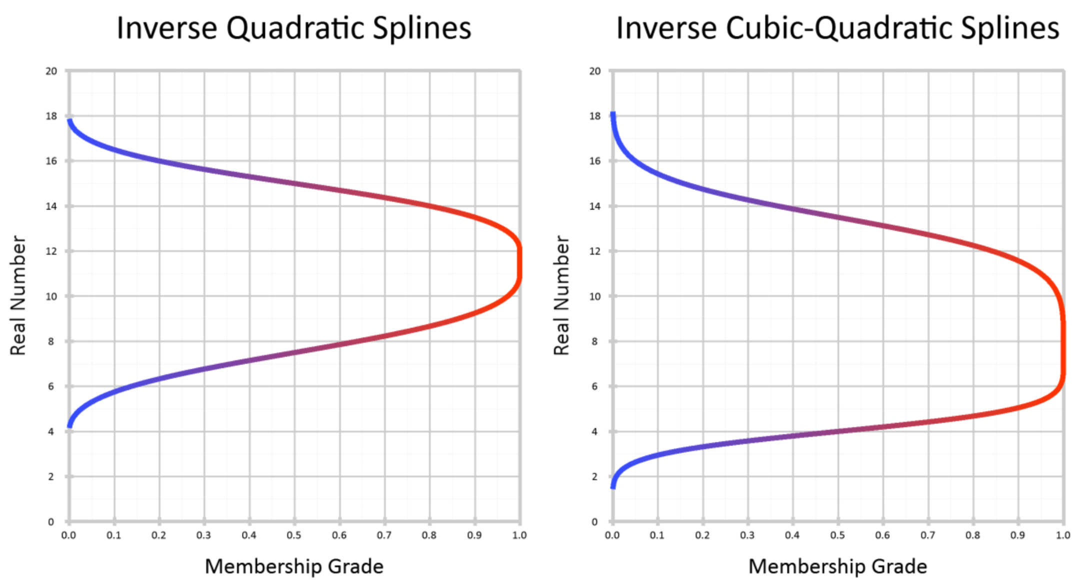


Figure 31. LU Fuzzy Numbers Using Inverse Rational Splines

CHAPTER 5

# LU-Fuzzy Control

Mamdani and Larsen Fuzzy Controllers have been used to manipulate complex data and produce solutions in realtime but are known to be more computationally expensive than Takagi-Sugeno Fuzzy Controllers. The bounded domain of LU-Fuzzy Numbers allows for potentially computationally less-expensive operations, as numerical integration can be performed over a set number of iterations that can be pre-defined. An LU-Fuzzy Controller can maintain the more intuitive and interpretable nature of Mamdani Controllers while potentially being less computationally expensive.

## 1. Definition of LU-Fuzzy Controller

As described previously, a Single Input Single Output Fuzzy System consists of a fuzzifier, fuzzy rule base, fuzzy inference system and defuzzifier. Most systems use the most basic fuzzifier: inclusion. Fuzzy systems of Mamdani type are built based on the minimum t-norm (denoted as $\wedge$) and the maximum t-conorm (denoted as $\vee$).

If $x \in X$ is a crisp input of the SISO fuzzy system with fuzzy rule base

$$\text{if } x \text{ is } A_i \text{ then } y \text{ is } B_i,\ i = 1, ..., n$$

then the fuzzy output of the system is given by

$$B'(y) = \bigvee_{i=1}^{n} A_i(x) \wedge B_i(y).$$

Supposing that both the antecedents, $A_i$, and consequents, $B_i$, are fuzzy numbers, given in the LU representation, then having given functions $(A_i)^-_\alpha$, $(A_i)^+_\alpha$ and $(B_i)^-_\alpha$, $(B_i)^+_\alpha$, $\alpha \in [0, 1]$. If $B'$ is fuzzy convex, then the following can be calculated:

$$B'(y)^\pm_\alpha = \left. \bigvee_{i=1}^{n} A_i(x) \wedge B_i(y) \right)^\pm_\alpha .$$

Then, for simplicity, if there are at any value $x$ only two fuzzy rules that are active at a time, i.e., $x \in (A_j)_0 \cap (A_k)_0$, it can be concluded that

$$B'(y)^-_\alpha = \begin{cases} B_j(y)^-_\alpha & \text{if } \alpha \le A_j(x)^-_\alpha \\ B_k(y)^-_\alpha & \text{if } A_j(x)^-_\alpha < \alpha \le A_k(x)^-_\alpha \\ 0 & \text{if } A_k(x)^-_\alpha < \alpha \end{cases}$$

$$B'(y)^+_\alpha = \begin{cases} B_k(y)^+_\alpha & \text{if } \alpha \le A_k(x)^+_\alpha \\ B_j(y)^+_\alpha & \text{if } A_k(x)^+_\alpha < \alpha \le A_j(x)^+_\alpha \\ 0 & \text{if } A_j(x)^+_\alpha < \alpha \end{cases} .$$

The output of this fuzzy system is the same as that of a traditional Mamdani fuzzy system, just with a new representation. This means that all the output and the properties for Mamdani fuzzy systems are kept intact by changing into LU representation.

In cases where more than two rules are active at a time, a slightly more computationally expensive solution can be obtained using the following:

$$B'(y)_\alpha^- = \begin{cases} B_j(y)_\alpha^- & \text{if } \exists i | A_i(x)_\alpha^- \leq \alpha, A_i(x)_\alpha^+ \geq \alpha \\ 0 & \text{if } \nexists i | A_i(x)_\alpha^- \leq \alpha, A_i(x)_\alpha^+ \geq \alpha \end{cases},$$

$$\text{where } A_j(x)_\alpha^- = MAX(A_i(x)_\alpha^-) \forall i | A_i(x)_\alpha^- \leq \alpha, A_i(x)_\alpha^+ \geq \alpha,$$

$$B'(y)_\alpha^+ = \begin{cases} B_j(y)_\alpha^+ & \text{if } \exists i | A_i(x)_\alpha^+ \geq \alpha, A_i(x)_\alpha^- \leq \alpha \\ 0 & \text{if } \nexists i | A_i(x)_\alpha^+ \geq \alpha, A_i(x)_\alpha^- \leq \alpha \end{cases},$$

$$\text{where } A_j(x)_\alpha^+ = MIN(A_i(x)_\alpha^+) \forall i | A_i(x)_\alpha^+ \geq \alpha, A_i(x)_\alpha^- \leq \alpha.$$

Examples of lower antecedents, $A_i(x)_\alpha^-$, and upper antecedents, $A_i(x)_\alpha^+$, are shown in Fig. 32; lower consequents, $B_i(x)_\alpha^-$, and upper consequents, $B_i(x)_\alpha^+$, are shown in Fig. 33. Examples of final LU antecedents and LU consequents are shown in Fig. 34, and Fig. 35 shows an example output using the LU Mamdani controller (with output identical to the LR Mamdani controller output shown in Fig. 17).

This still produces a single LU Fuzzy Number and, as a result, departs from traditional Mamdani fuzzy systems that would not normally produce a set that is fuzzy convex.

Defuzzification is the final step in a fuzzy system. Based on the fuzzy output of a fuzzy controller, a crisp quantity must be produced for the output value of the controller. As described previously, there are several defuzzification methods. Based on a given application, a convenient defuzzification method can be selected.

A popular choice for defuzzification is Center of Gravity (COG). The traditional center of gravity of $u \in \mathcal{F}(X)$, weighted by the membership grade is

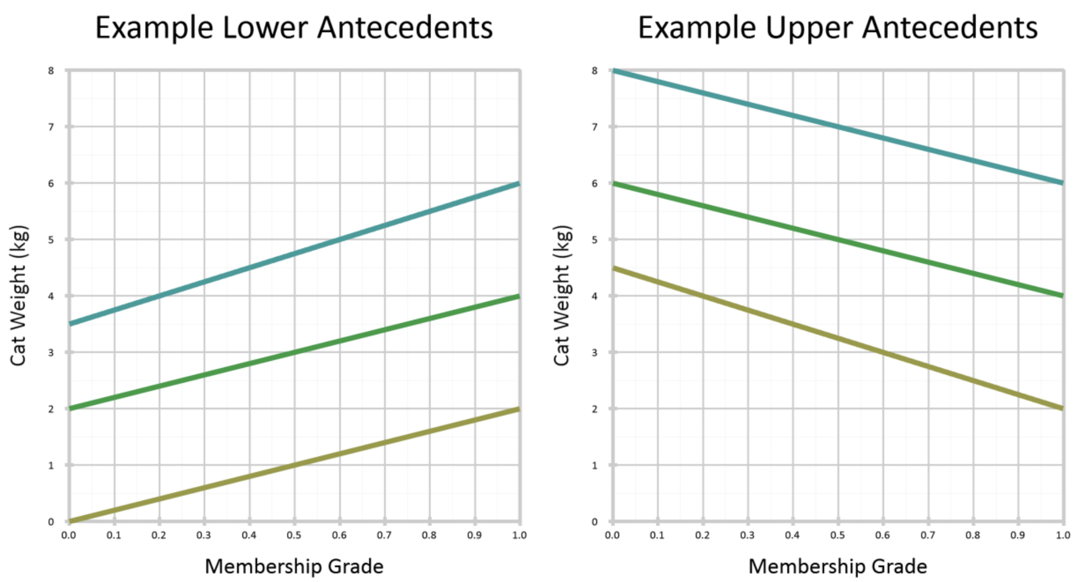


Figure 32. Lower Antecedents, $A_i(x)_\alpha^-$, and Upper Antecedents, $A_i(x)_\alpha^+$

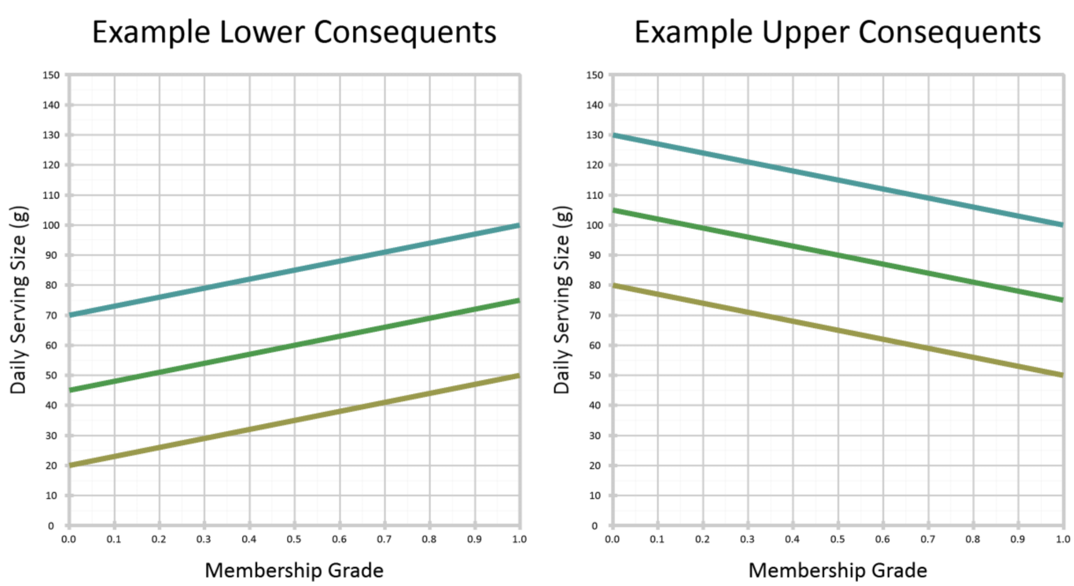


Figure 33. Lower Consequents, $B_i(x)_\alpha^-$, and Upper Consequents, $B_i(x)_\alpha^+$

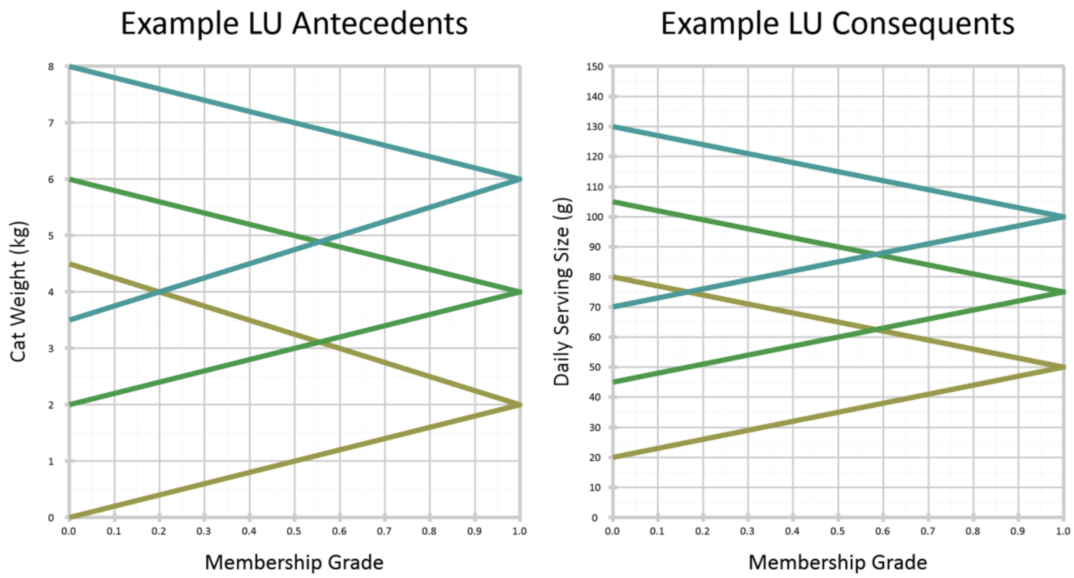


Figure 34. Example LU Antecedents and LU Consequents

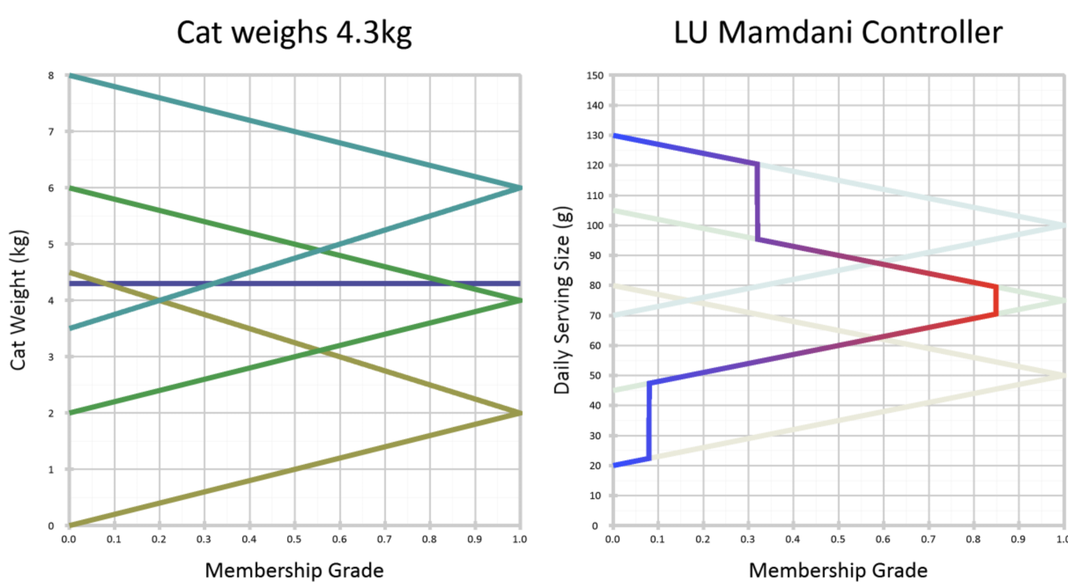


Figure 35. Example LU Fuzzy Controller; Outputs Identical to Fig. 17

calculated using

$$COG(u) = \frac{\int_X x \cdot u(x)dx}{\int_X u(x)dx}, \tag{5.1}$$

where $X$ is the universe of discourse for a given problem. This integral can be restricted to the support of the fuzzy set, $u$, but given that the endpoints of the support can be anywhere on the universe of discource, this can be computationally expensive.

The Center of Gravity defuzzification can be calculated very efficiently using the LU representation. This can be easily obtained using:

$$COG(u) = \frac{\frac{1}{2}\int_0^1 [(u_r^+)^2 - (u_r^-)^2]dr}{\int_0^1 (u_r^+ - u_r^-)dr}. \tag{5.2}$$

First, observe that the denominator calculates the area of $u$. The numerator is obtained from

$$\int_0^1 (u_r^+ - u_r^-)\frac{u_r^+ + u_r^-}{2}dr,$$

which gives the above expression.

The Expected Value defuzzification:

$$EV(u) = \int_0^1 \frac{u_r^+ + u_r^-}{2}dr,$$

could also be used in the proposed fuzzy system and it would be computationally less expensive, but it is not necessarily better than COG.

### 2. Comparison between LU and LR timing

The comparison of the COG expressions in the two approaches allows for the computational advantage of the LU approach in Mamdani systems to be immediately observed. The primary advantage in using the LU model is the ability to numerically integrate over the inverval $[0, 1]$ rather than the entire universe of discourse. Even further, numerically, it is possible to simply integrate over $[0, \max(A_j, A_k)]$, taking even fewer cycles to calculate.

A complexity estimation for the COG defuzzification discussed above can be shown. Let $X = [a, b]$ be the universe of discourse in a fuzzy control application. The COG calculated by (5.1) relies on the calculation of two integrals on the $[a, b]$ interval. Suppose that the same quadrature rule is used for the calculation of the integrals, e.g., trapezoid rule. It is well known that the error in the trapezoid rule (supposing that the integrand is twice differentiable, see [12]) is given by

$$Error_1 \leq \frac{M_1}{12} \frac{(b-a)^3}{n_1^2},$$

where $M_1 = sup_{x \in [a,b]} |u''(x)|$, $u$ is the membership function of the fuzzy set being considered, and $n_1$ represents the number of subintervals used for the quadrature rule. Consider the COG calulated by 5.2, then calculate two integrals on the $[0, 1]$ interval. The error for both can then be estimated

$$Error_2 \leq \frac{M_2}{12 n_2^2},$$

with

$$M_2 = \max(sup_{r \in [0,1]} |((u_r^+)^2 - (u_r^-)^2)'|, sup_{r \in [0,1]} |(u_r^+ - u_r^-)'|)$$

and $n_2^2$ the number of subintervals. Suppose that the values of the constants $M_1, M_2$ are comparable while $b - a = 10$ (the universe of discourse is e.g. $X = [0, 10]$). In this case, to have the same error estimate, the following can be stated:

$$\frac{M_1 \cdot 1000}{M_2} = \frac{n_1^2}{n_2^2}.$$

This shows that the time complexity of the two algorithms has the quotient approximately $\frac{M_1 \cdot 1000}{M_2}$.

While the time complexity of LU controller may be drastically less than the LR controller, Takagi-Sugeno controllers are still more efficient than both. However, Takagi-Sugeno controllers are considered to be less intuitive than Mamdani controllers, and so there is an immediate advantage in interpretability when using LU and LR controllers.

For direct computational timings, a function approximation application was run thousands of times using both LU and LR SISO Mamdani fuzzy controllers with COG defuzzifiers (with each producing near-identical output).

| | LR | LU |
|---|---|---|
| Slowest Time | 1.446305s | 0.018977s |
| Fastest Time | 1.354017s | 0.008881s |
| Average Time | 1.362096s | 0.013563s |

From this data, it can be observed that the LU fuzzy controller is at least 71 times faster than the LR fuzzy controller. The slowest LU iterations are over 76 times faster than the slowest LR iterations. The average times for LR take over 100 times longer than the average LU times. The fastest times are over 152 times faster for LU than LR. Finally, in a best-case scenario, the LU fuzzy controller can be over 162 times faster than the LR fuzzy controller. The fact that the LU fuzzy controller can be two orders of magnitude less expensive makes it far more applicaple to real-time applications.

CHAPTER 6

# Applications

## 1. Function Approximation

The study of approximation capability of fuzzy systems was first proposed by B. Kosko. Most literature either uses the Takagi-Sugeno approach or uses the sum as the aggregation method for the fuzzy rules. In [19] it was shown that the function providing the output of the Larsen type fuzzy system is capable of approximating any continuous function and that it is continuously differentiable under very relaxed conditions (when antecedents have continuous differentiability except at their core, and consequents have continuous differentiability except at the core and the support endpoints):

**Theorem 1** *Any continuous function* $f : [a, b] \to [\alpha, \beta]$ *can be approximated by the Larsen fuzzy system*

$$F(f,x) = \frac{\int_{\alpha}^{\beta} [\bigvee_{i=1}^{n} A_i(x) \cdot B_i(y)] \cdot y \cdot dy}{\int_{\alpha}^{\beta} [\bigvee_{i=1}^{n} A_i(x) \cdot B_i(y)] \cdot dy}$$

*with any membership functions for the antecedents and consequents* $A_i, B_i, i = 1, ..., n$ *such that there exist* $\varepsilon > 0$, $r \in \mathbb{N}$, $r < n$, *such that* *(i)* $A_i$ *continuous,* $A_i(x_i) = 1$

$$(A_i)_\varepsilon \subseteq [x_{i-r}, x_{i+r}], i = 1, ..., n;$$

*(ii)* $B_i$ *integrable,* $B_i(y_i) = 1$,

$$(B_i)_\varepsilon \subseteq [\min\{y_{i-r}, ..., y_{i+r}\}, \max\{y_{i-r}, ..., y_{i+r}\}],$$

$y_i = f(x_i)$, $i = 1, ..., n$.

*Moreover the following error estimate holds true*

$$\|F(f,x) - f(x)\| \leq 2r\omega\,(f,\delta) + \varepsilon^2(\beta-\alpha)^2 M$$

*with*

$$\delta = \max_{i=1,...,n} \{x_i - x_{i-1}\}$$

*and*

$$M = \quad \int_\alpha^\beta [\bigvee_{i=1}^n A_i(x) \cdot B_i(y)] \cdot dy\Bigg)^{-1}.$$

It is also observable that Mamdani, Lukasiewicz, and Gödel SISO fuzzy systems using identical antecedents and consequents can be used in function approximation.

**1.1. Smoothness.** In investigating approximation, smoothness properties of Larsen type single input single output (SISO) fuzzy systems begin to become appararent. The case of a fuzzy system of Larsen type creating output which is continuously differentiable may not be immediately obvious because of the usage of the maximum operator, which is known to destroy differentiability. Smoothness

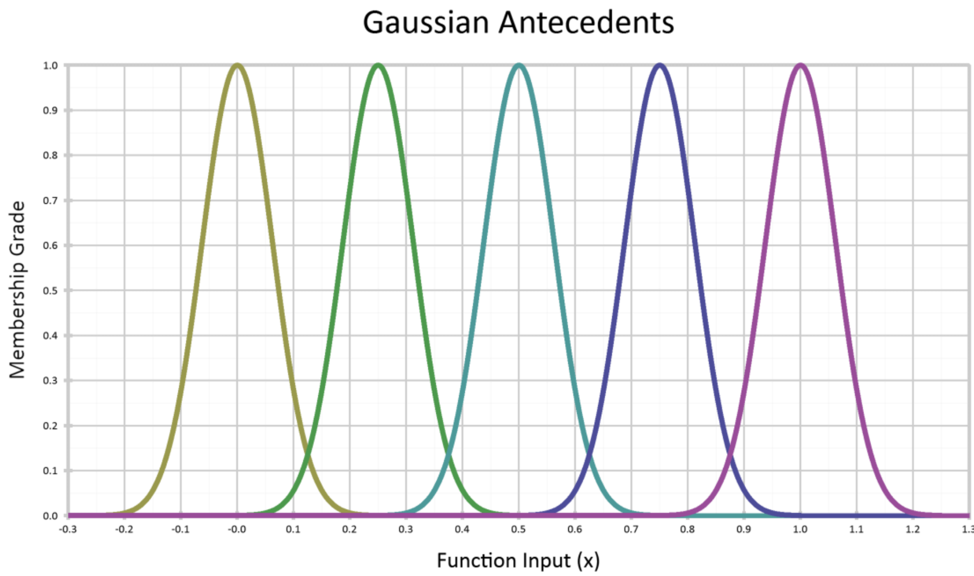


Figure 36. Gaussian Antecedents ($A_i$) Used to Approximate $x^2$

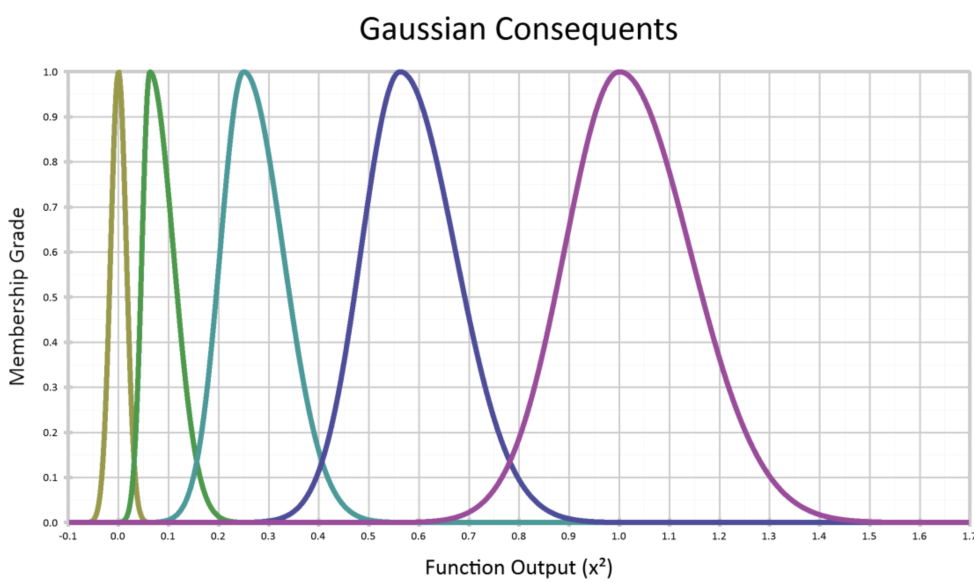


Figure 37. Gaussian Consequents ($B_i$) Used to Approximate $x^2$

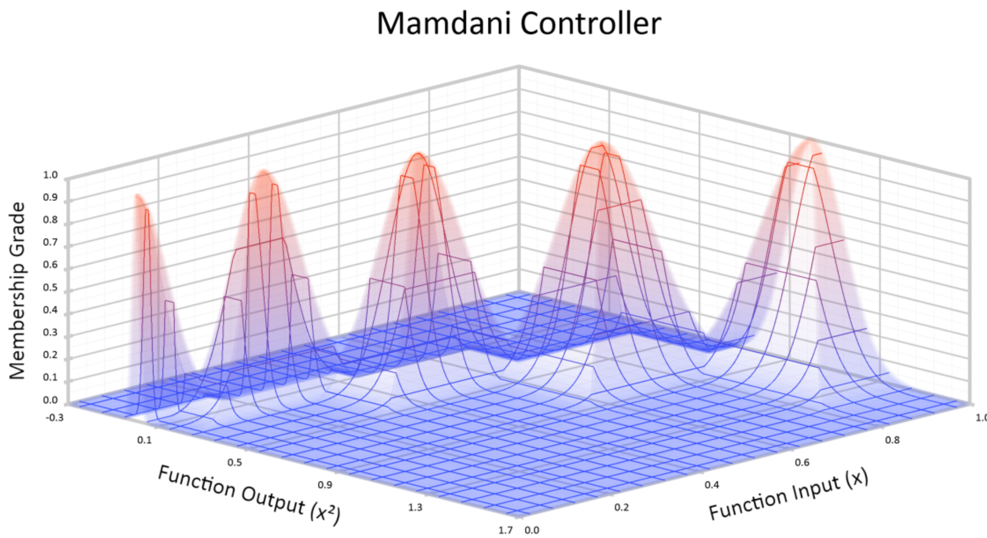


Figure 38. Mamdani Controller Using Gaussian Input to Approximate $x^2$

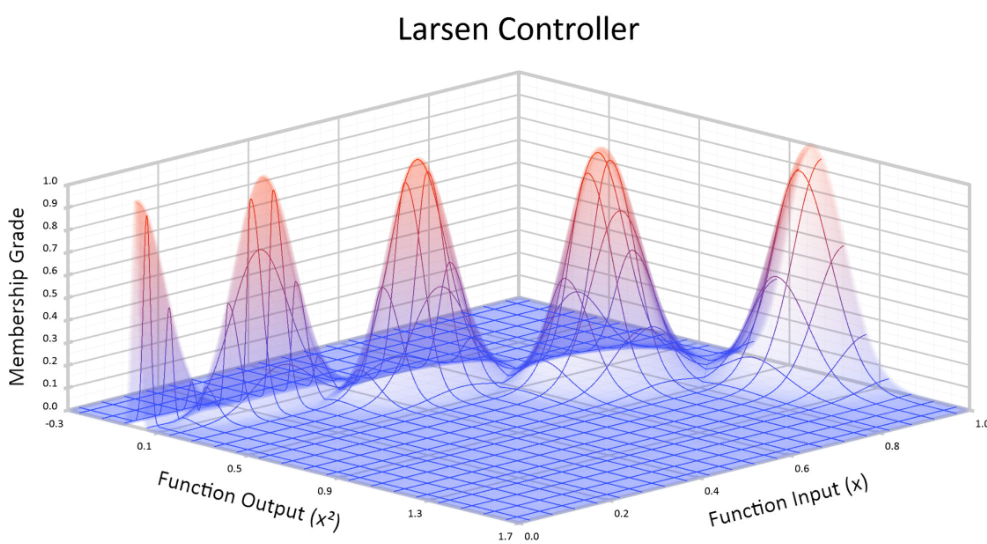


Figure 39. Larsen Controller Using Gaussian Input to Approximate $x^2$

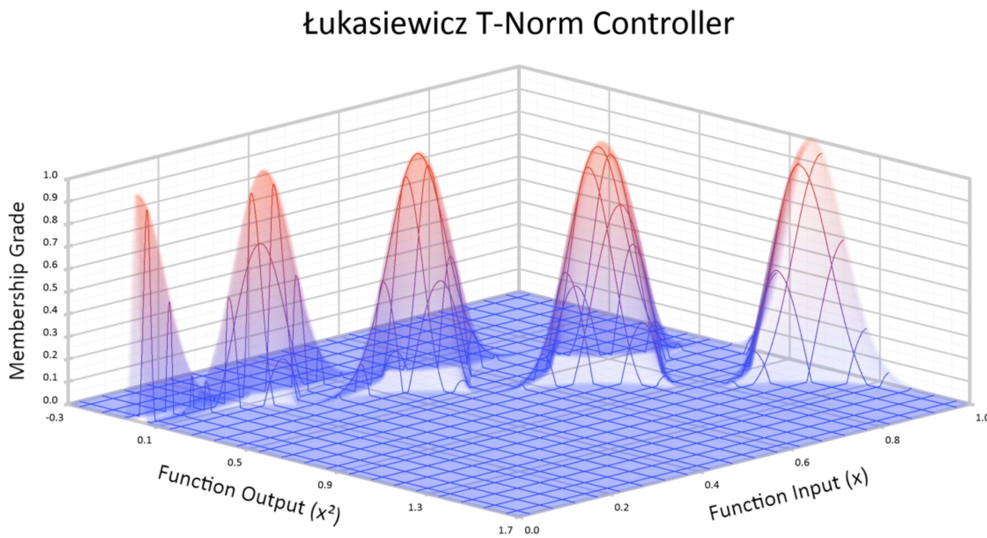


Figure 40. T-Norm Controller Using Gaussian Input to Approximate $x^2$

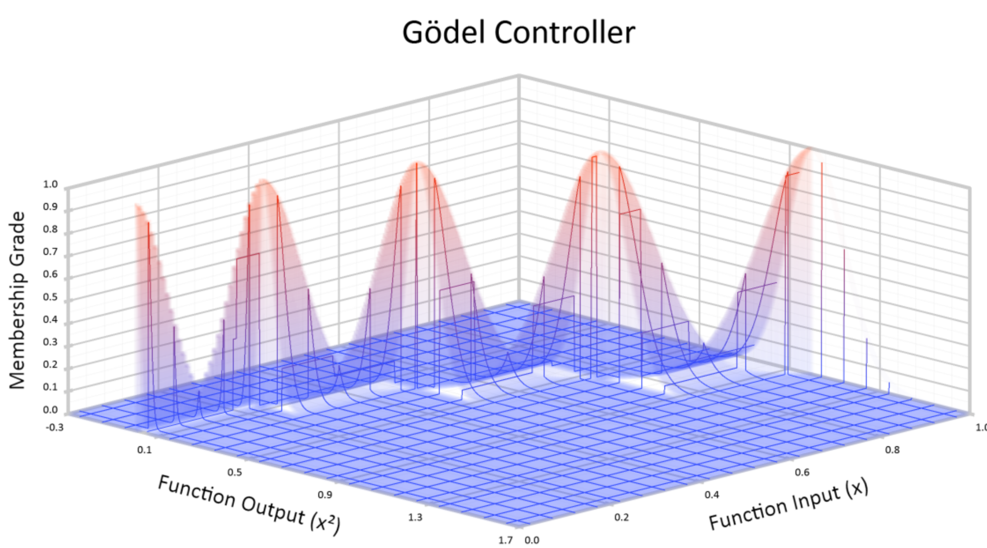


Figure 41. Gödel Controller Using Gaussian Input to Approximate $x^2$

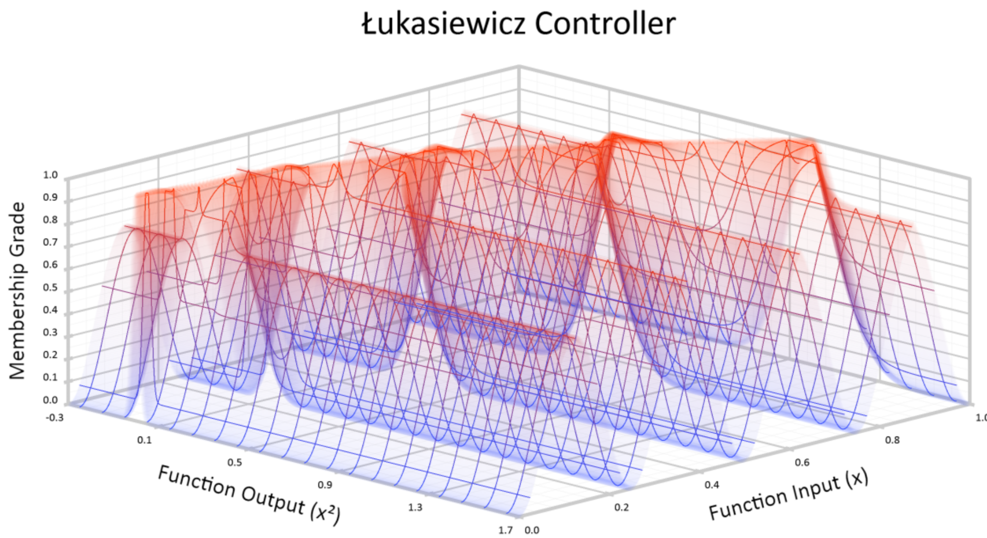


Figure 42. Gödel Risidual Controller Used to Approximate $x^2$

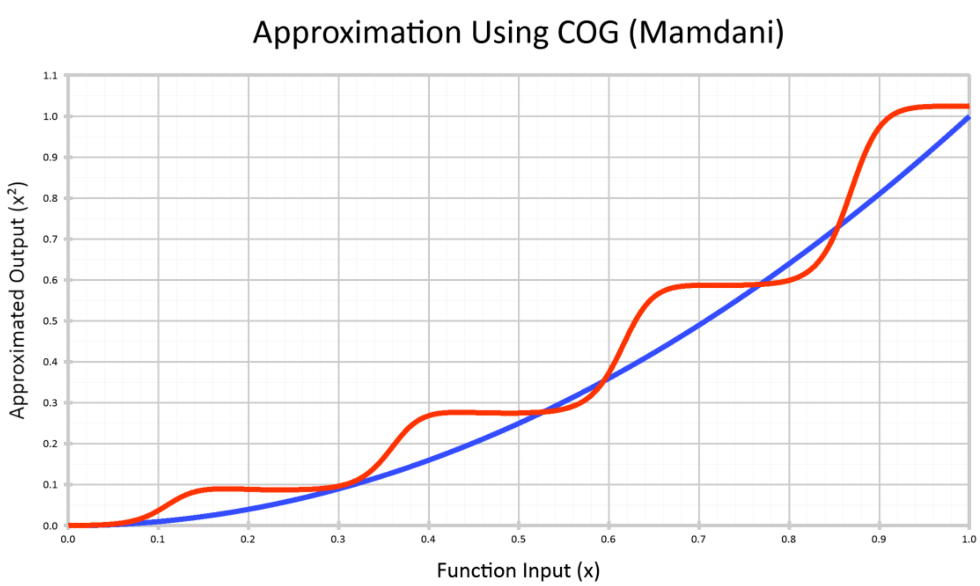


Figure 43. Center Of Gravity Output Using Fig. 38 to Approximate $x^2$

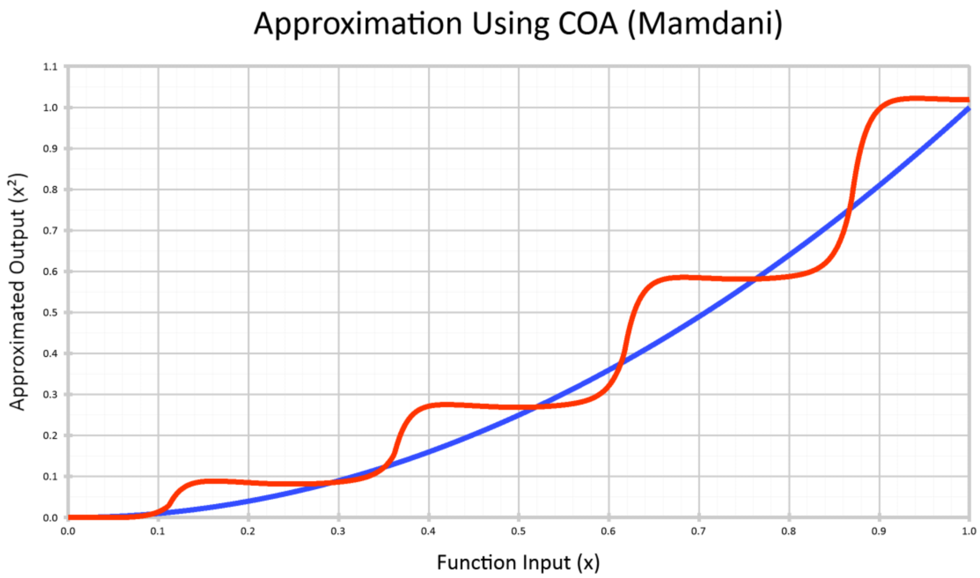


Figure 44. Center Of Area Output Using Fig. 38 to Approximate $x^2$

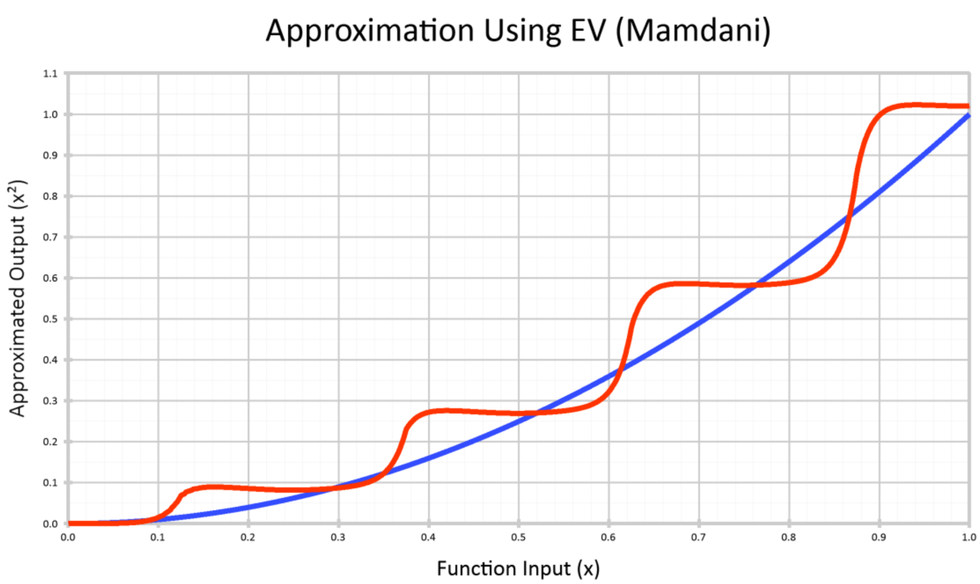


Figure 45. Expected Value Output Using Fig. 38 to Approximate $x^2$

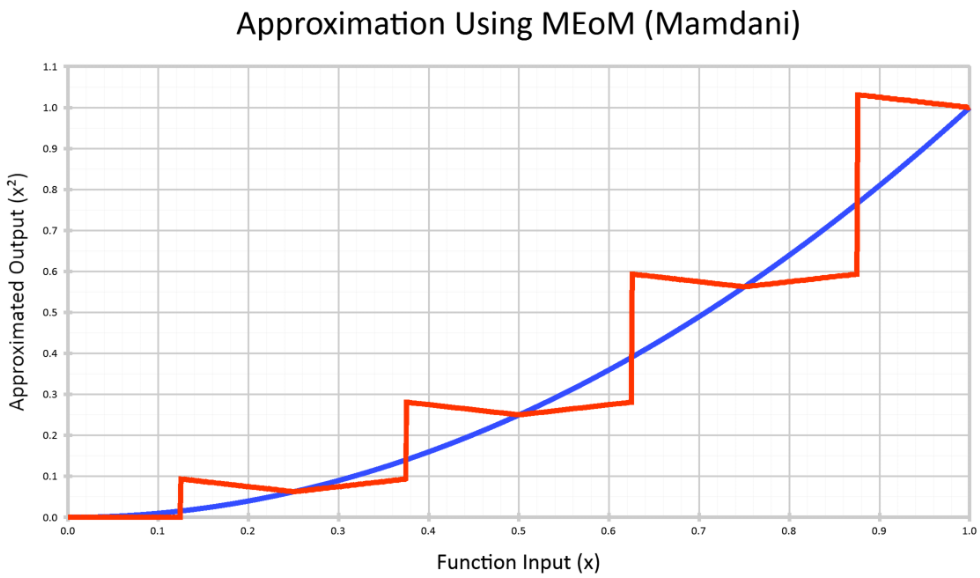


Figure 46. Mean of Maxima Output Using Fig. 38 to Approximate $x^2$

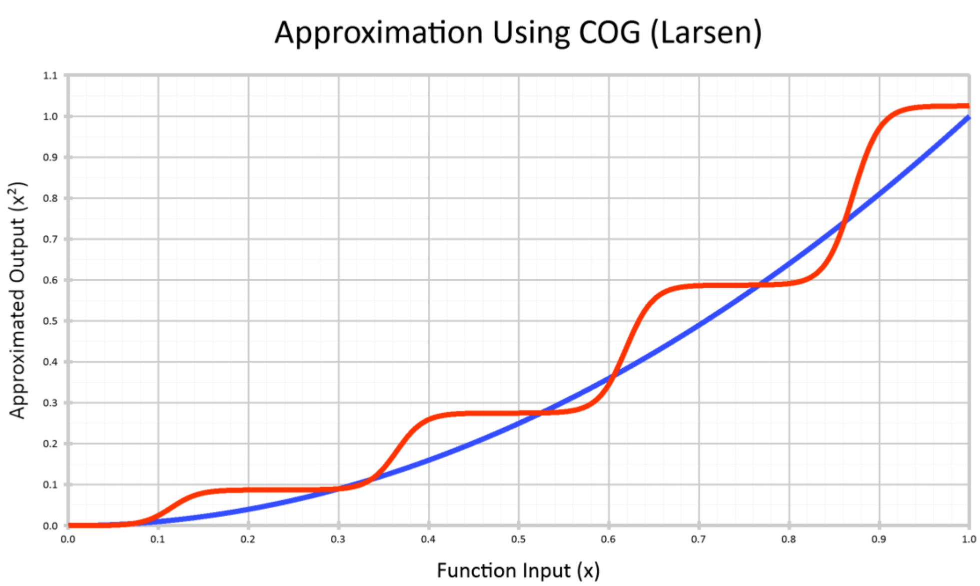


Figure 47. Center Of Gravity Output Using Fig. 39 to Approximate $x^2$

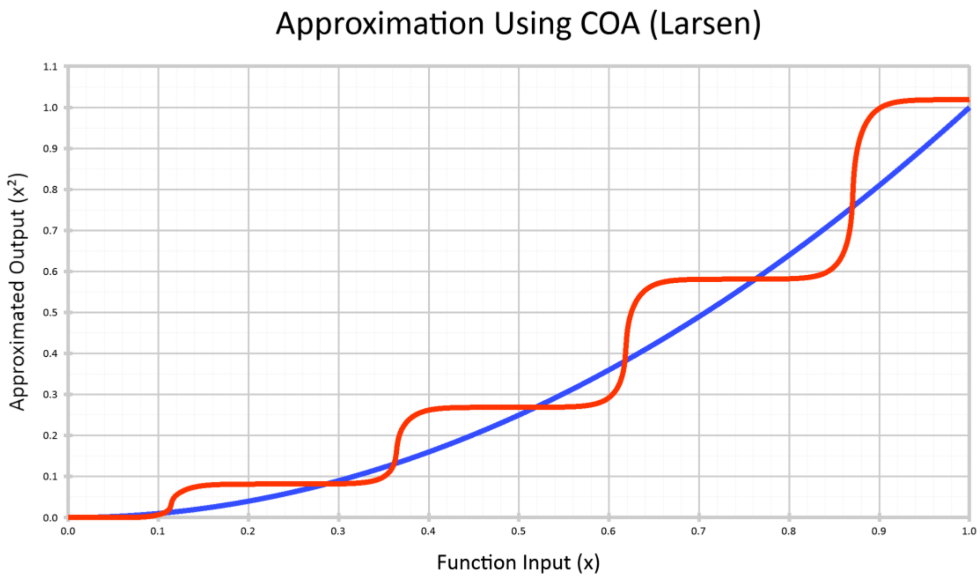


Figure 48. Center Of Area Output Using Fig. 39 to Approximate $x^2$

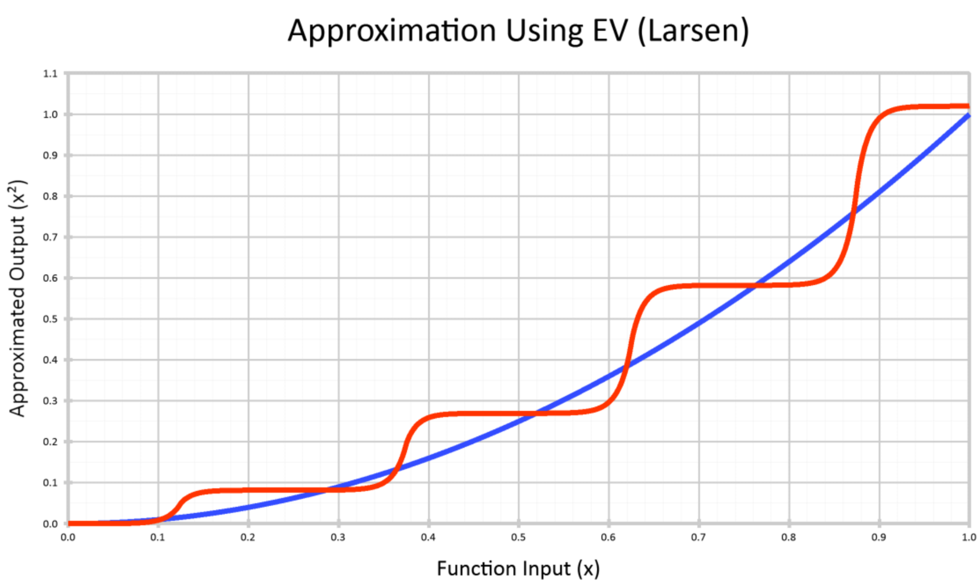


Figure 49. Expected Value Output Using Fig. 39 to Approximate $x^2$

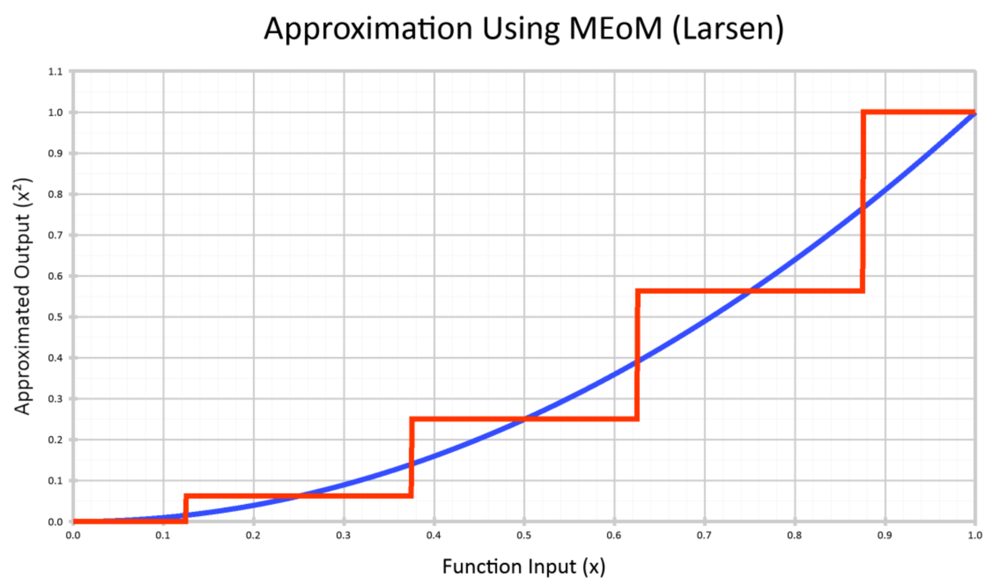


Figure 50. Mean of Maxima Output Using Fig. 39 to Approximate $x^2$

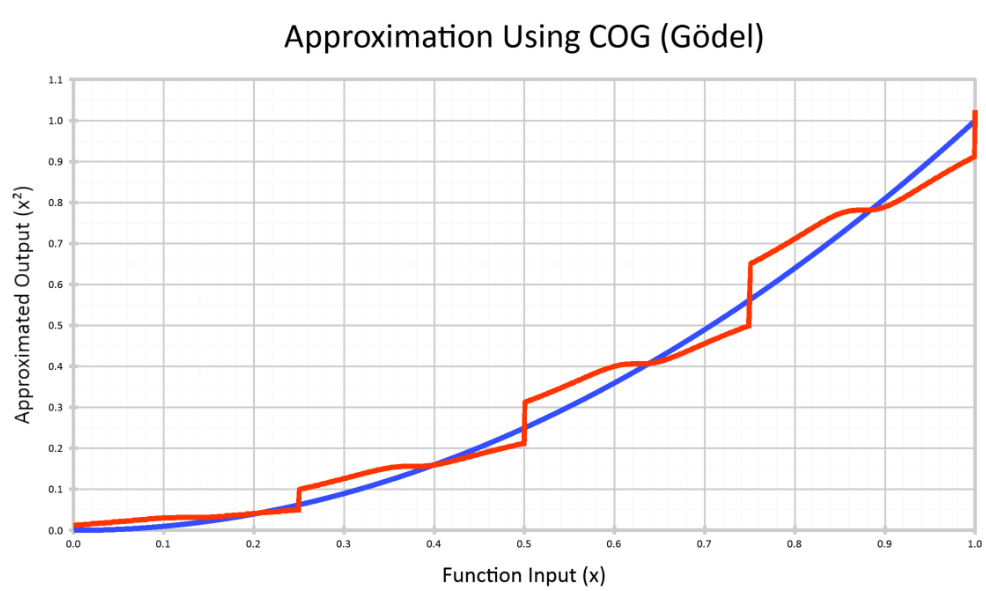


Figure 51. Center Of Gravity Output Using Fig. 41 to Approximate $x^2$

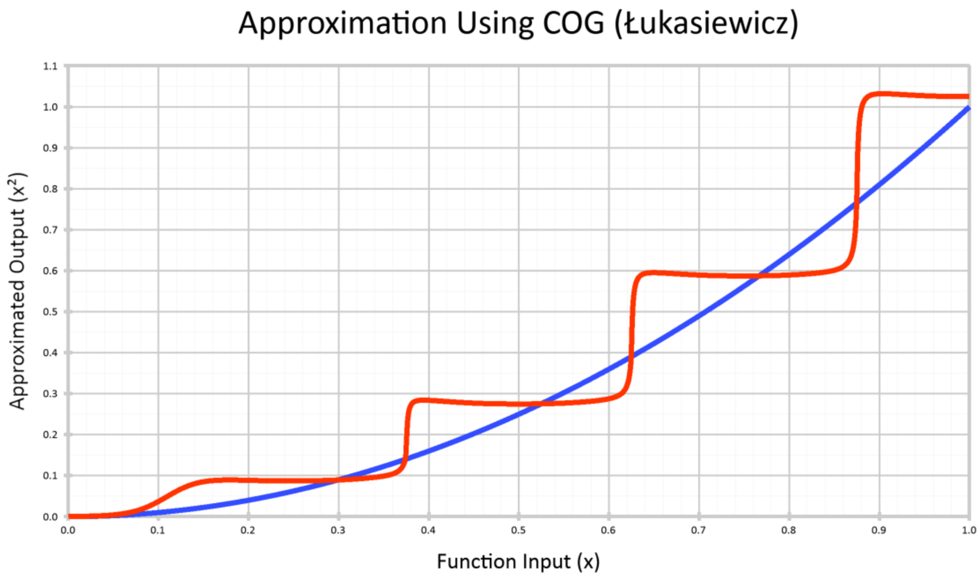


Figure 52. Center Of Gravity OutputUsing Fig. 42 to Approximate $x^2$

has been investigated in cases where additive fuzzy systems were used with Gaussian membership functions. Sggregation using fuzzy implications have also been considered, but smoothness properties were not investigated. The fact that a fuzzy system provides a smooth output is very intuitive and has been mentioned in other works. In fact, this is widely known as being one of the main advantages of fuzzy controllers of Mamdani types and Larsen types. Fuzzy logic systems using the maximum as aggregation for the individual rule outputs, product (Goguen) t-norm as the conjunctive operator and center of gravity defuzzification were investigated and shown to be, under very relaxed conditions, continuously differentiable[19]:

**Theorem 2** *Let* $f : [a, b] \to \mathbb{R}$ *be a monotone function and let* $y_i = f(x_i)$, $i =$

*1, ..., n.. Consider the Larsen type SISO fuzzy system*

$$F(f,x) = \frac{\int_\alpha^\beta [\bigvee_{i=1}^n A_i(x) \cdot B_i(y)] \cdot y \cdot dy}{\int_\alpha^\beta [\bigvee_{i=1}^n A_i(x) \cdot B_i(y)] \cdot dy}$$

*with any membership functions for the antecedents and consequents $A_i, B_i, i = 1, ..., n$ satisfying*

*(i) $A_i$ monotone increasing and differentiable on $(-\infty, x_i)$ and monotone decreasing and differentiable on $(x_i, \infty)$, with the closure of its support being*

$$(A_i)_0 = [x_{i-1}, x_{i+1}], i = 1, ..., n;$$

*(ii) $B_i$ strictly increasing and differentiable on $[\min\{y_{i-1}, y_i, y_{i+1}\}, y_i)$ and strictly decreasing and differentiable on $(y_i, \max\{y_{i-1}, y_i y_{i+1}\}]$,*

$$(B_i)_0 = [\min\{y_{i-1}, y_i, y_{i+1}\}, \max\{y_{i-1}, y_i, y_{i+1}\}], i = 1, ..., n;$$

*Then the Larsen type system given above is continuous and continuously differentiable function (class $C^1$) on $[a, b]$.*

Visually, similar smoothness properties can be observed in Mamdani and Lukasiewicz type SISO fuzzy systems when using COA and COG as well.

**1.2. Approximation Using LU Representation.** Again considering a given continuous function $f : [a, b] \to \mathbb{R}$. The function is approximated using LU fuzzy controllers, using different LU fuzzy numbers used to describe antecedents and consequents. The LU representation is then compared with the widely accepted membership function representation (LR representation).

Let $x_0 \leq x_1 \leq ... \leq x_n$ be a partition of $[a, b]$ such that $f(x_0) = y_0$, $f(x_1) = y_1, ... f(x_n) = y_n$. The conclusions are ordered triplets $(y_0 \leq ... \leq y_n)$.

If the antecedents and consequents are LU parametric fuzzy numbers:

$$A_k = (\alpha_{ki}; u_{ki}^-, \delta u_{ki}^-, u_{ki}^+, \delta u_{ki}^+)_{i=0,1,...,N},$$

$$B_k = (\alpha_{ki}; v_{ki}^-, \delta v_{ki}^-, v_{ki}^+, \delta v_{ki}^+)_{i=0,1,...,N},$$

$k = 0, ..., n$ that satisfy the conditions

$$u_{k0}^- = x_{k-1}, u_{kN}^- = u_{kN}^+ = x_k, u_{k0}^+ = x_{k+1},$$

$$v_{k0}^- = y_{k-1}, v_{kN}^- = v_{kN}^+ = y_k, v_{k0}^+ = y_{k+1},$$

assuming $x_{-1}$ and $x_{n+1}$ are auxiliary knots with equidistant data. The remaining parameter values can be used to increase the adaptivity of the system and potentially produce more accurate approximations. In the simple case of $N = 1$ described in (4.8) and (4.9), there are:

$$u_{k0}^- = x_{k-1}, u_{k1}^- = u_{k1}^+ = x_k, u_{k0}^+ = x_{k+1}$$

$$v_{k0}^- = y_{k-1}, v_{k1}^- = v_{k1}^+ = y_k, v_{k0}^+ = y_{k+1}.$$

with no restriction on the values $\delta u_{k0}^-, \delta u_{k1}^-, \delta u_{k1}^+, \delta u_{k0}^+$.

If non-parametric LU fuzzy numbers are used, smoother results can potentially be achieved but the accuracy that the parameters allow for may be sacrificed.

To start, an LU fuzzy controller that approximates $f(x) = x^2$, using 5 rules, with triangular antecedents and consequents is demonstrated as a baseline (triangular fuzzy numbers can be achieved using either parametric or non-parametric LU fuzzy numbers). The result of approximation with $f(x) = x^2$ and triangular fuzzy numbers are shown in Figure 53.

Next, considering a non-monotonic function such as

$$f(x) = x + \frac{\sin(30x)}{10},$$

Its approximation obtained using an LU fuzzy controller with triangular antecendent is shown in Figure 54.

Consider parametric LU fuzzy numbers using cubic quadratic rational splines for the antecedents and consequents. For $f(x) = x^2$, these are shown in Figure 55. The result is shown in Figure 56.

If non-parametric LU fuzzy numbers are used for the antecedents and consequents, smoother results can be achieved. For $f(x) = x^2$, these are shown in Figure 57. The produced approximation is shown in Figure 58.

Finally, in Figure 59 and Figure 60 the same non-monotonic function from Figure 54 is approximated. The LU parametric fuzzy numbers were optimized to produce a more accurate approximation; the non-parametric LU fuzzy numbers are capable of producing a smooth result.

## 2. Games

In determining the applicability of the previously defined LU fuzzy controller, an implementation consisting of sampling of an environment that includes multiple agents and then modifying parameters which affect the system being driven via a single fuzzy controller was created. The implemented LU SISO fuzzy system takes the position of whichever non-fuzzy controlled agent is in the lead as input and determines how much the fuzzy controlled agent should accelerate as output. If the player-controlled agents gain a larger lead, the fuzzy-controlled agent will rapidly catch up and will likely win. In this way, a group of players is motivated to play strategically and ensures a closer race (see Fig.61). An LR implementation was taken and timings were recorded across thousands of iterations:

| | LR | LU |
|---|---|---|
| Slowest Time | 9869046ns | 64050ns |
| Fastest Time | 4237978ns | 9032ns |
| Average Time | 4873806ns | 27728ns |

While the LR times are still beneath the 16ms required to maintain 60hz, the much-faster LU times allow for other computations to be performed and more agents to be active simultaneously.

The LU SISO fuzzy controller was also expanded to allow multiple inputs in order to drive modular behaviors of individual agents and singular systems using multiple fuzzy controllers. Each LU fuzzy controller manipulated the movement rules for a single fish, with one controller dictating acceleration towards a goals while another determined tangental acceleration used to avoid other fish swimming around them (see Fig. 62).

The rules for turning were simply:

If GoalIsFarBehind and GoalIsLeft, then TurnLeftVeryFast.

If ClosestFishIsRight and ClosestFishIsInFront, then TurnLeftFast.

If GoalIsLeft, then TurnLeft.

If GoalIsRight, then TurnRight.

If ClosestFishIsLeft and ClosestFishIsInFront, then TurnRightFast.

If GoalIsFarBehind and GoalIsRight, then TurnRightVeryFast.

The rules for speeding up were also straightforward:

If GoalIsInFront and ClosestFishIsBehind, then SpeedUp.

Is MovingForwardFast and ClosestFishIsCenter and ClosestFishIsInFront, then SlowDown.

Timings were taken between LR and LU implementations:

| | LR | LU |
|---|---|---|
| Slowest Time | 4210470ns | 92790ns |
| Fastest Time | 2142392ns | 2463ns |
| Average Time | 2661820ns | 26630ns |

The nature of the LR controller consistently taking longer than 2ms makes it prohibitive for real-time use; the LU controller's speed allows for it to be used with many more agents in a single 60hz frame.

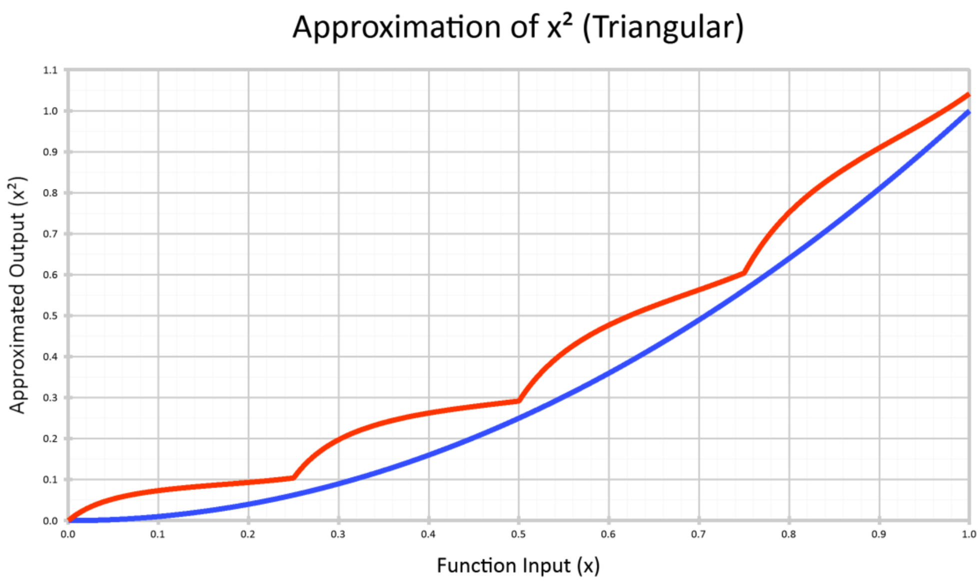


Figure 53. Approximation of $x^2$ (Triangular LU)

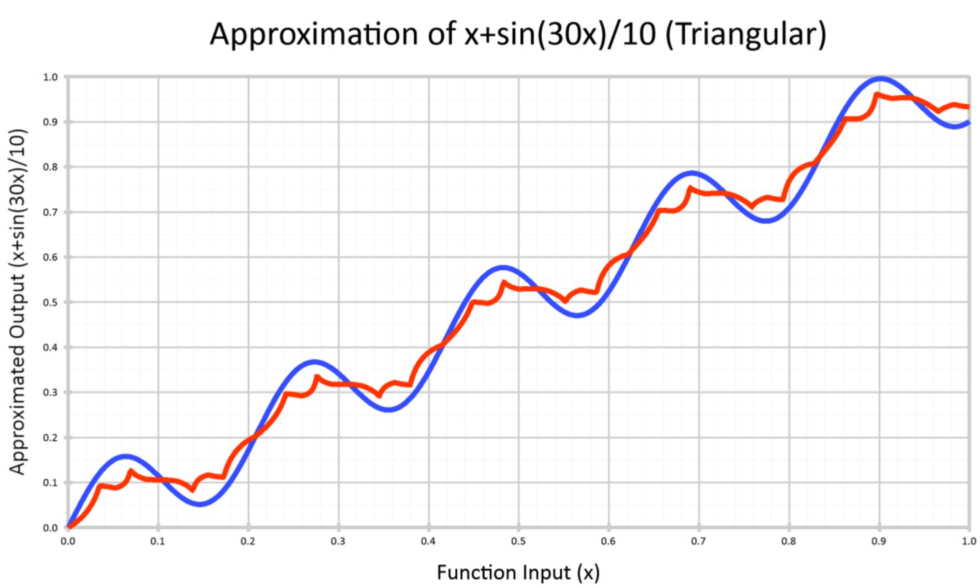


Figure 54. Approximation of $f(x) = x + \frac{\sin(30x)}{10}$ (Triangular LU)

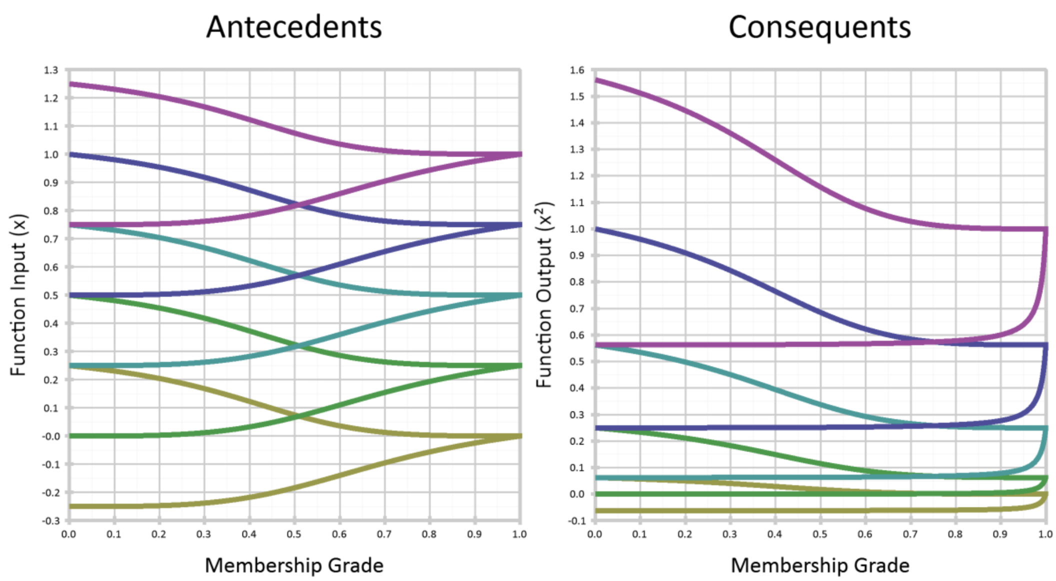


Figure 55. Input For Approximating $x^2$ (LU Parametric)

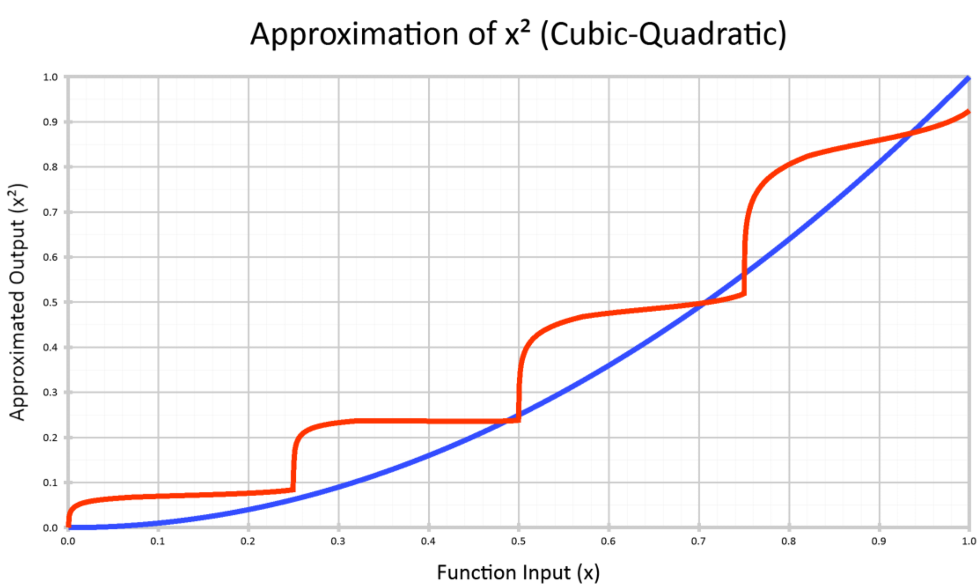


Figure 56. Approximation of $f(x) = x + \frac{\sin(30x)}{10}$ (LU Parametric)

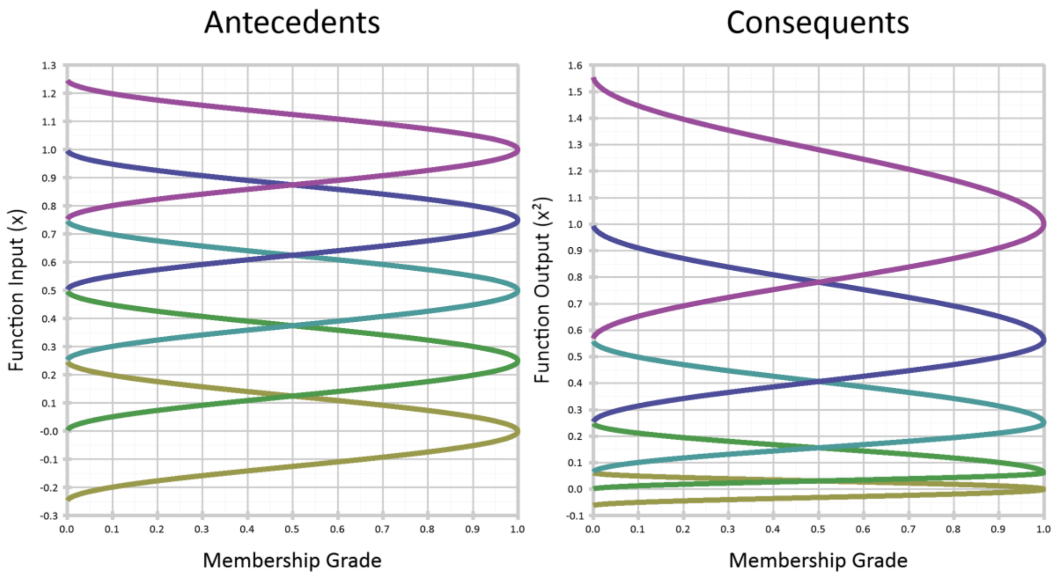


Figure 57. Input For Approximating $x^2$ (Non-Parametric LU)

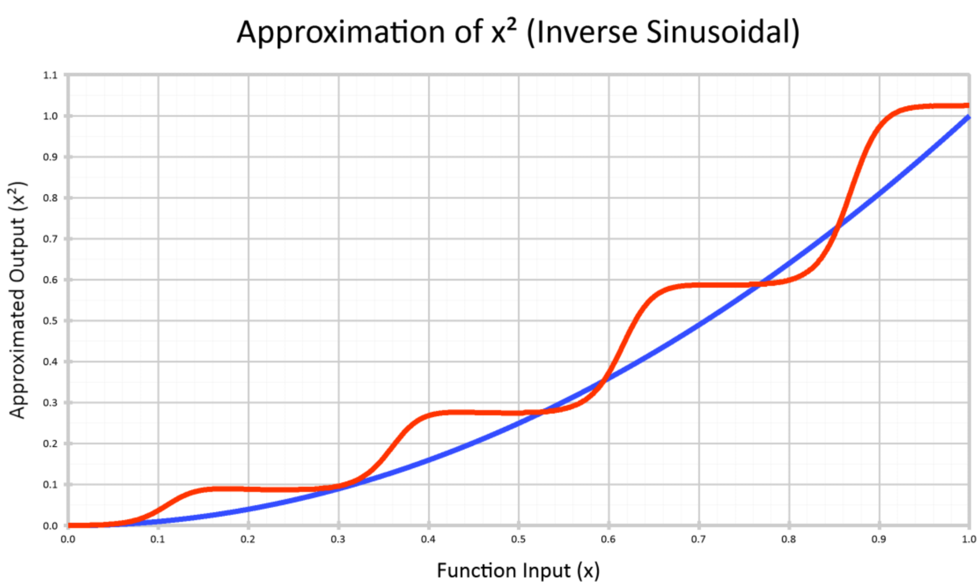


Figure 58. Approximation of $x^2$ (Non-Parametric LU)

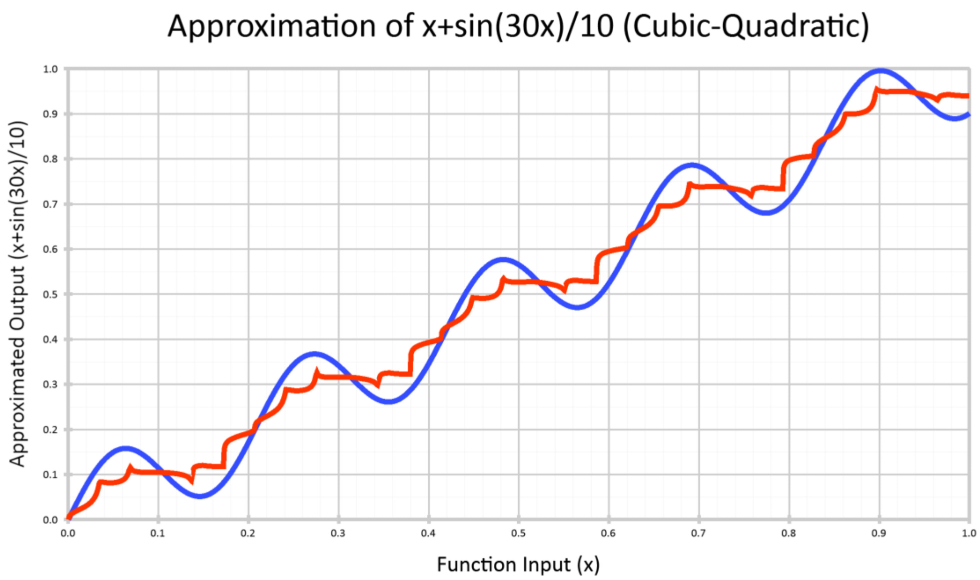


Figure 59. Approximation of $x + \frac{\sin(30x)}{10}$ (LU Parametric)

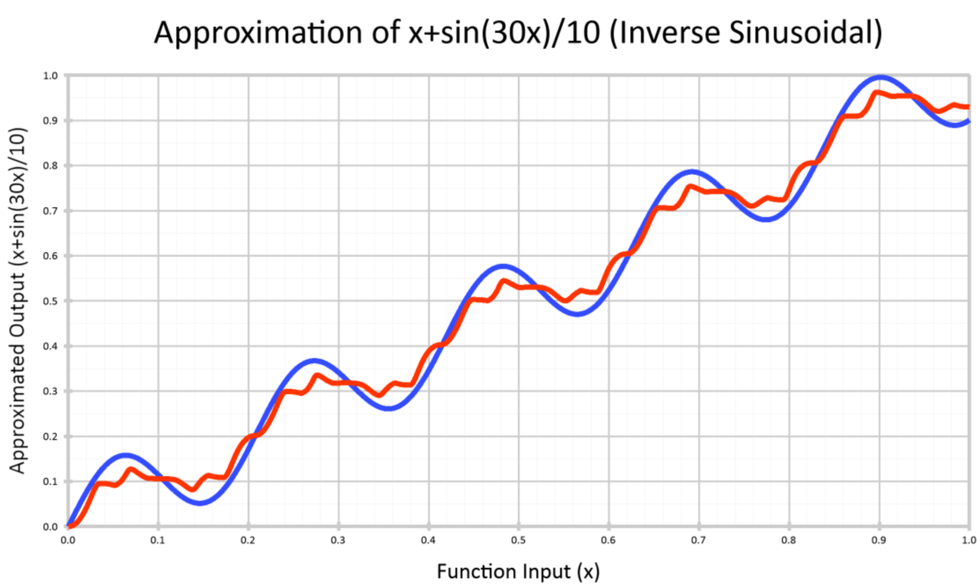


Figure 60. Approximation of $x + \frac{\sin(30x)}{10}$ (Non-Parametric LU)

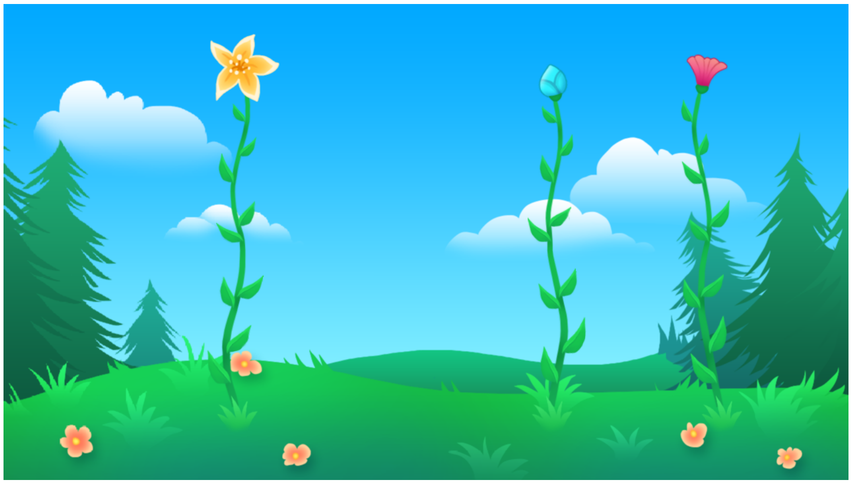

Figure 61. Flowers Race to Grow

Figure 62. Fish Swim Towards Goals and Avoid One Another

CHAPTER 7

# Conclusions and Future Work

LU fuzzy controllers allow for intuitive manipulation of data in complex environments without the steep computational penalties that traditional LR fuzzy controllers entail, allowing for real-time applications to be explored with far less impedence.

Further research could be done to create a hybrid LR-LU fuzzy controller, where LR antecedents are used with LU consequents, which could allow for similarly low computation costs while also enabling output closer to that of Larsen and Lukasiewicz fuzzy systems.

# REFERENCES

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

[2] D. Mark, K. Dill. Improving AI Decision Making Through Utility Theory, AI Summit GDC 2010, 2010.

[3] J. M. P. Van Waveren, The Quake III Arena Bot, University of Technology Delft, 2001.

[4] A. Shaout, B. King, L. Reisner. Real-Time Game Design of Pac-Man Using Fuzzy Logic, The International Arab Journal of Information Technology, Vol. 3, No. 4, 2006.

[5] N. Sanornoi, P. Sooraksa. Artificial Intelligence based on Fuzzy Behavior for Game Programming, Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology, pp. 277-279, ACM, 2004.

[6] R. Narayanan, Fuzzy Logic Washing Machines, Electronicsforu.com, 2003.

[7] J. H. Kim, K. S. Kim, M. S. Sim, K. H. Han, B. S. Ko. An application

of fuzzy logic to control the refrigerant distribution for the multi-type air conditioner, Fuzzy Systems Conference Proceedings, 1999, FUZZ-IEEE'99, 1999 IEEE International, vol. 3, pp. 1350-1354. IEEE, 1999.

[8] C. B. Chiou, C. H. Chiou, C. M. Chu, and S. L. Lin. The application of fuzzy control on energy saving for multi-unit room air-conditioners, Applied Thermal Engineering 29, no. 2 (2009): 310-316.

[9] J. N. Lygouras, P. N. Botsaris, J. Vourvoulakis, V. Kodogiannis. Fuzzy logic controller implementation for a solar air-conditioning system, Applied Energy 84, no. 12 (2007): 1305-1318.

[10] Center, A. I., & Semarak, J. (1996). Intelligent traffic lights control by fuzzy logic. Malaysian Journal of Computer Science, 9(2), 29-35.

[11] Y. Li, P. Musilek, L. Wyard-Scott. Fuzzy Logic in Agent-Based Game Design, Proceedings of the 2004 Annual Meeting of the North American Fuzzy Information Processing Society, Banff, Cananda, vol. 2, pp. 734-39, 2004.

[12] R.L. Burden, J. D. Faires. Numerical Analysis. 2001. Brooks/Cole, USA.

[13] P. Diamond, P. Kloeden, Metric Spaces of Fuzzy Sets, World Scientific, New Jersey, 1994.

[14] L. Stefanini, L. Sorini, M.L. Guerra, Parametric representation of fuzzy numbers and application to fuzzy calculus, Fuzzy Sets and Systems 157 (2006) 2423 – 2455.

[15] L. Stefanini, L. Sorini, M.L. Guerra, Fuzzy numbers and Fuzzy Arithmetic, Chapter 12 in W. Pedrycz, A. Skowron, V. Kreinovich (eds), Handbook of Granular Computing, John Wiley & Sons, Ltd, 2008.

[16] T. Takagi, M. Sugeno, Fuzzy identification of system and its applications to modelling and control, IEEE Transactions on Systems, Man and Cybernetics, 15 (1985) 116-132

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

[18] R.E. Precup, M.L. Tomescu, E.M. Petriu, S. Preitl, C.A. Dragos, Stable Design of a Class of Nonlinear Discrete-Time MIMO Fuzzy Control Systems, Acta Polytechnica Hungarica 9.2 (2012), 57-76.

[19] B. Bede, M. Peterson, Approximation Properties and Continuous Differentiability of a Class of SISO Fuzzy Systems, 2013 IEEE 8th International Symposium on Applied Computational Intelligence and Informatics (SACI), 11-16.

[20] B. Bede, M. Peterson, Fuzzy systems of Mamdani type in the LU representation, IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013 Joint, 908-913.