

**Circumventing the Fuzzy Type Reduction for
Autonomous Vehicle Controller**

WAJDI RASHEED ISMAEEL AL-RIKABI

**Submitted in Partial Fulfilment of the Requirements of the
Degree of Doctor of Philosophy**

**University of Salford, Salford, UK
Autonomous Systems and Robotics Centre
School of Computing, Science & Engineering**

2017

Table of Content

List of Figures	V
Acronyms List	XII
Acknowledgments	XIV
Abstract	1
Chapter 1: Introduction	2
1.1. Introduction	2
1.2. Fuzzy Type-2 Fundamentals	3
1.2.1. Interval Fuzzy Type-2 Sets	5
1.2.2. Embedded Fuzzy Sets	5
1.2.3. Fuzzy Controllers	6
1.3. Research Hypothesis	7
1.4. Structure of the Thesis.....	8
Chapter 2: Fuzzy Type Reductions.....	9
2.1. Fuzzy Type-2 Sets Reduction	9
2.2. Defuzzification	9
2.3. Type Reduction Principles	11
2.4. Interval Fuzzy Sets Type Reductions.....	13
2.4.1. Iterative Karnik & Mendel (IKM) Type Reduction.....	13
2.4.2. Approximated KM Type Reduction (AKM).....	15
2.4.3. Wu-Mendel (Max-Min) Approximation	16
2.4.4. Wu-Tan Type Reduction.....	17
2.4.5. The Nie-Tan (N-T) method.....	19
2.4.6. Collapse Type Reduction	19
2.4.7. Enhanced Karnik–Mendel (EKM) Type Reduction Procedure	21
2.4.8. EKM Type Reduction Using Incremental Formula	23
2.4.9. Type Reduction Using Type-1 OWA	24

2.4.10. Weighted Enhanced Karnik–Mendel Type Reduction Procedure	26
2.4.11. Piecewise Linear Interval Fuzzy Type-2 Set Reduction	27
2.4.12. Linearly Approximated Karnik-Mendel Type reduction	30
2.5. Special Cases Type Reduction	31
2.6. Reduction of The General Fuzzy Type-2 Sets	35
2.6.1. General Fuzzy Sets Reduction Using Vertical Slice	37
2.6.2. The α -Planes Type Reduction Method	38
2.6.3. Random Sampling Method	39
2.6.4. The Enhanced α -Plane Type Reduction ($E\alpha$ -Plane).....	41
2.6.5. Quasi-Fuzzy Type-2 Sets Reduction.....	41
2.6.6. Geometric Type-2 Reduction-Defuzzification.....	42
2.7. Interval Set TRPs Comparison	43
2.8. General Set Reductions Comparison.....	48
2.9. Conclusions	50
Chapter 3: Autonomous Vehicle Fuzzy.....	52
Type-2 Controllers	52
3.1. The Fuzzy Type-2 Mamdani Controller.....	52
3.2. The Takagi-Sugeno Fuzzy Controller	55
3.3. The Fuzzy Type-1 Controller	56
3.4. Autonomous Vehicles	57
3.5. Autonomous Vehicle Fuzzy Type-2 Controllers Survey	58
3.6. Survey Summary of Autonomous Fuzzy Type-2 Controllers.....	64
3.7. Autonomous Fuzzy Type-2 Controller Survey Statistics.....	66
Chapter 4: Type Reduction Approximation.....	69
4.1. Research Goals and Methodology.....	69
4.2. Type Reduction Development.....	70
4.3. Basic Numerical Quadrature Rules	72

4.4. Adaptive Quadrature Rules	73
4.5. Suggested Error Evaluation.....	75
4.6. Error in Fuzzy Type Reduction	80
4.6.1. The COG's Error Dependency	84
4.7. Correct Interval Length Search	86
4.8. COG Using Adaptive Trapezoidal	88
Chapter Five: Performance Evaluation.....	94
5.1. Evaluating the Adaptive Type Reduction	94
5.1.1. Using Variable Approximation Allowances	99
5.2. Variable Approximation Allowance Error	101
5.3. Performance at Optimal Parameters.....	107
5.4. Fuzzy Type-2 Autonomous Controller using Adaptive COG.....	114
Chapter 6: Redundancy Elimination.....	118
6.1. Accelerating the EKM.....	118
6.2. A Global Type Reducer.....	121
6.3. The Proposed One-Go Type Reduction Procedure	124
6.3.1. Benefits of the Proposed One-Go Procedure	125
Chapter 7: Implementation Aspects.....	126
7.1. The Autonomous Vehicle.....	126
7.2. Using Schmitt-Trigger.....	127
7.3. Using Schmitt-Trigger.....	130
Chapter 8: Conclusions and Future Work.....	133
8.1. Conclusions	133
8.2. Contributions.....	135
8.3. Future Work	136
Appendix-A.....	138
Appendix-B	145

Appendix-C	158
Appendix-D	160

List of Figures

Figure 1-1 : Typical Fuzzy Sets. (a) Type-1. (b) General Type-2. (c) Interval Type-2.	4
Figure 1-2: Structure of fuzzy type-1 controllers.	7
Figure 2-1 : Iterative KM type reduction over IFT2 set.	14
Figure 2-2: Wu-Mendel Approximation Method and N-T set.	17
Figure 2-3: Example of interval FT2 sets reduction using Wu-Tan method. (a) Is the first input set: $X1$. (b) Is the second input set, $X1$, with its equivalent sets $feq(x2)$ being evaluated at three different input levels: $X1 = \{+1, 0, -1\}$	18
Figure 2-4 : (a) Collapse type reduction principles. (b) Different progress patterns effect...	20
Figure 2-5 : EKM iterations principles.	21
Figure 2-6: EKM Type Reduction in Incremental Formula. (a) Left switching point search. (b) Right switching point search.	24
Figure 2-7 : Piecewise Linear IFT2 Set reduction.	27
Figure 2-8 : Enhanced PWL type reduction cases. (a) Left and right centroid enhancement for $(UMF' - LMF') < 0$. (b) Left and right centroid enhancement when $(UMF' - LMF') > 0$	30
Figure 2-9: LR Interval FT2 sets possible cases and the possible switching point locations (1, 2, 3, 4, 5).	33
Figure 2-10: Type reduction principles of an interval FT2 set with spikes.	34
Figure 2-11: Vertical Slice Type Reduction of a General Type-2 Set.	37
Figure 2-12: General fuzzy type-2 Set decomposed by alpha planes at the values 0, 0.16, 0.33, 0.5, 0.66, 0.83 and 1.	38

Figure 2-13 : Random sampling, type reduction-defuzzification, accuracy.....	40
Figure 2-14: Importance Random Sampling type reduction principles. (a) Using importance control factor $\beta = 0$. (b) Using importance control factor $\beta = 8$	41
Figure 2-15: Geometric type reduction-defuzzification. (a) Two triangles shape the secondary membership function in a general FT2 set. (b-c) The possible sides to be described using triangles are a, b, c, d and e.....	42
Figure 2-16: Interval FT2 sets type reduction procedures cost comparison.....	47
Figure 2-17: Computational cost comparison of the general FT2 set reductions.	50
Figure 3-1: Main units of the fuzzy type-2 controllers.....	53
Figure 3-2 : Structure of the Takagi-Sugeno Fuzzy Control System.	56
Figure 3-3: Interval FT2 sets usage rate in autonomous vehicles control researches, during 1994-2014	67
Figure 3-4: Autonomous FT2 Controllers researches during the past ten years.	68
Figure 3-5: Type reduction suitability for autonomous FT1 control applications.....	68
Figure 4-1: Riemann sum maximum error limits for evenly distributed points.	71
Figure 4-2: Trapezoidal error over one long interval.	76
Figure 4-3: Estimating the possible error limit of the trapezoidal rule by monitoring the difference between a straight line and the curved section.	84
Figure 4-4: Integration result fluctuation because of error dependency.....	85
Figure 4-5: Traditional successive binary search.	87
Figure 4-6: Output fuzzy sets sub-sectioning example.....	88
Figure 4-7 : The different progressing cases of the FP iterative procedure showing all the cases in a, b, c, and d.....	90

Figure 4-8 : Some possible effects on the progressing pattern of an iterative FP routine are shows in the sub figures a, b, and c, that are caused by adding one bit noise.93

Figure 5-1: Two fuzzy type-2 sets used to evaluate the output error.96

Figure 5-2: Average time gain for the approximated type reduction using fuzzy transition step size of 20% at different approximation allowances and initial integration multipliers for: (a) - IFT2 set width of 10%, (b) - IFT2 set width of 20%, (c) - IFT2 set width of 40%.....97

Figure 5-3: Average time gain for the approximated type reduction using fuzzy transition step size of 10% at different approximation allowances and initial integration multipliers for: (a) - IFT2 set width of 10%, (b) - IFT2 set width of 20%, (c) - IFT2 set width of 40%.....98

Figure 5-4: Average time-gain for the approximated type reduction using Gaussian IFT2 set having width of 10% and fuzzy transition step size of 10%, at different approximation allowances and various integration multipliers of the second iteration..... 100

Figure 5-5: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 10% and incremental fuzzy transition step size of 10%, at different approximation allowance multipliers during the second iteration calculations. 103

Figure 5-6: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 10% and incremental fuzzy transition step size of 20%, at different approximation allowance multipliers during the second iteration calculations. 104

Figure 5-7: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 10% and decrementing fuzzy transition step size of 10%, at different approximation allowance multipliers during the second iteration calculations. 105

Figure 5-8: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 10% and decrementing fuzzy transition step size of 20%, at

different approximation allowance multipliers during the second iteration calculations.	106
Figure 5-9: Time-Gain for all the possible combinations of two fuzzy firing levels, U1 and U2, transitioning from one to zero in step size of 10%, and acting on two Gaussian's IFT2 sets having 10% interval width.	108
Figure 5-10: Time-Gain for all the possible combinations of two fuzzy firing levels, U1 and U2, transitioning from one to zero in step size of 10%, and acting on two Gaussian's IFT2 sets having 20% interval width.	109
Figure 5-11: Time-Gain for all the possible combinations of two fuzzy firing levels, U1 and U2, transitioning from one to zero in step size of 10%, and acting on two Gaussian's IFT2 sets having 40% interval width.	110
Figure 5-12: Time-Gain for all the possible combinations of two fuzzy firing levels, U1 and U2, transitioning from one to zero in step size of 20%, and acting on two Gaussian's IFT2 sets having 10% interval width.	111
Figure 5-13: Time-Gain for all the possible combinations of two fuzzy firing levels, U1 and U2, transitioning from one to zero in step size of 20%, and acting on two Gaussian's IFT2 sets having 20% interval width.	112
Figure 5-14: Time-Gain for all the possible combinations of two fuzzy firing levels, U1 and U2, transitioning from one to zero in step size of 20%, and acting on two Gaussian's IFT2 sets having 40% interval width.	113
Figure 5-15: Wall following test field arena.....	115
Figure 5-16 : Comparing two fuzzy autonomous wall following controllers.	115
Figure 5-17: Execution time-gain at calculation allowances of one and two using control loop times of 10ms, 20ms, and 30ms for 10%, 20%, and 40% interval widths.	116
Figure 5-18: Mean absolute error at calculation allowances of one and two using control loop times of 10ms, 20ms, and 30ms for 10%, 20%, and 40% interval widths.	116

Figure 5-19: The traditional KM and adaptive type reduction time-gain for output discretisation levels of 250, 500, and 1000 points using interval set width of 10%....	117
Figure 6-1: Modified incremental EKM.....	120
Figure 6-2: Execution time gain of the modified incremental EKM execution time in comparison to the initial incremental formula using different interval set widths.	121
Figure 6-3: Possible computation cut due to using a one-go type reduction.....	123
Figure 6-4: Time gain due to reducing multiple sets in one go	123
Figure 7-1: The Pioneer P3-DX mobile robot.	127
Figure 7-2: Areas where the left or right wall following state can be captured and where they can act.	128
Figure 7-3: Left and right wall-following discrimination using Schmitt-trigger	130
Figure 7-4: Scaling the distance sensors for the wall-following behaviour	132
Figure 8-1: Average time-gain for the approximated type reduction using Gaussian IFT2 set having width of 10% and fuzzy transition step size of 10%, at different approximation allowances and various integration multipliers of the second iteration.....	139
Figure 8-2: Average time-gain for the approximated type reduction using Gaussian IFT2 set having width of 20% and fuzzy transition step size of 10%, at different approximation allowances and various integration multipliers of the second iteration.....	140
Figure 8-3: Average time-gain for the approximated type reduction using Gaussian IFT2 set having width of 40% and fuzzy transition step size of 10%, at different approximation allowances and various integration multipliers of the second iteration.....	141
Figure 8-4: Average time-gain for the approximated type reduction using Gaussian IFT2 set having width of 10% and fuzzy transition step size of 20%, at different approximation allowances and various integration multipliers of the second iteration.....	142

Figure 8-5: Average time-gain for the approximated type reduction using Gaussian IFT2 set having width of 20% and fuzzy transition step size of 20%, at different approximation allowances and various integration multipliers of the second iteration..... 143

Figure 8-6: Average time-gain for the approximated type reduction using Gaussian IFT2 set having width of 40% and fuzzy transition step size of 20%, at different approximation allowances and various integration multipliers of the second iteration..... 144

Figure 8-7: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 10% and incremental fuzzy transition step size of 10%, at different approximation allowance multipliers during the second iteration calculations. 146

Figure 8-8: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 20% and incremental fuzzy transition step size of 10%, at different approximation allowance multipliers during the second iteration calculations. 147

Figure 8-9: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 40% and incremental fuzzy transition step size of 10%, at different approximation allowance multipliers during the second iteration calculations. 148

Figure 8-10: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 10% and incremental fuzzy transition step size of 20%, at different approximation allowance multipliers during the second iteration calculations. 149

Figure 8-11: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 20% and incremental fuzzy transition step size of 20%, at different approximation allowance multipliers during the second iteration calculations. 150

Figure 8-12: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 40% and incremental fuzzy transition step size of 20%, at

different approximation allowance multipliers during the second iteration calculations.	151
Figure 8-13: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 10% and decrementing fuzzy transition step size of 10%, at different approximation allowance multipliers during the second iteration calculations.	152
Figure 8-14: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 20% and decrementing fuzzy transition step size of 10%, at different approximation allowance multipliers during the second iteration calculations.	153
Figure 8-15: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 40% and decrementing fuzzy transition step size of 10%, at different approximation allowance multipliers during the second iteration calculations.	154
Figure 8-16: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 10% and decrementing fuzzy transition step size of 20%, at different approximation allowance multipliers during the second iteration calculations.	155
Figure 8-17: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 20% and decrementing fuzzy transition step size of 20%, at different approximation allowance multipliers during the second iteration calculations.	156
Figure 8-18: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 40% and decrementing fuzzy transition step size of 20%, at different approximation allowance multipliers during the second iteration calculations.	157

Acronyms List

AGV	: Autonomous Ground Vehicle.
APF	: Artificial Potential Field path planning.
COG	: Centre of Gravity
COS	: Center of Sets type reduction.
CPU	: Central Processing Unit.
DV	: Difference in a Vertical displacement, which estimated to evaluate error in a COG calculation.
EKM	: Enhanced Karnik–Mendel type reduction procedure.
EKM-IF	: Enhanced Karnik–Mendel, type reduction procedure, using Incremental Form.
FCM	: Fuzzy C-Mean Clustering algorithm.
FCS	: Fuzzy Control System.
FOU	: Footprint of Uncertainty.
FP	: The Fixed Point.
FPGA	: Field Programmable Gate Arrays.
FST1	: Fuzzy Set Type-1.
FST2	: Fuzzy Set Type-2.
FT1	: Fuzzy Type-1.
FT2	: Fuzzy Type-2.
GS	: Global Stop point.
IKM	: Iterative Karnik–Mendel, type reduction procedure.
ISE	: Integrated Square Error.
KM	: Karnik–Mendel, Type Reduction Procedure.
LMF	: Lower Membership Function.
MOM	: Mean of Maximum type reduction.

- OWA** : Ordered Weighted Average.
- PDF** : Probability Density Function
- PID** : Proportional Integral Derferential.
- PS** : Progressing Steps size of Iterative Routine
- PWL** : Piecewise Linear.
- RG** : Reduced Gradient method.
- SIF** : Sampling Importance Factor.
- T1FLS** : Type-1 Fuzzy Logic System.
- T2FLS** : Type-2 Fuzzy Logic System.
- TES** : Total Embedded Sets.
- TRDF** : Type Reduction-Defuzzification process
- TRP** : Type Reduction Process.
- TSK** : Takagi Sugeno Kang fuzzy system model.
- Type-1 OWA** : Type-1 Ordered Weighted Average.
- UMF** : Upper Membership Function.
- VS** : Vertical Slice.
- VSTR** : Vertical Slice Type Reduction.
- WEKM** : Weighted Enhanced Karnik–Mendel, type reduction procedure.

Acknowledgments

I would like to thank my supervisors, Professor Samia Nefti-Meziani and Dr. Theo Theodoridis for their helpful and insightful advice to transmitting my ideas into reality and getting this work done.

I would also like to thank all the workers at the Autonomous Systems and Robotics Centre, and my colleagues, who were supportive, and helpful. Many of them have demonstrated that an intelligent thoughtful question is as valuable for true understanding of a problem as is the solution itself.

I dedicate my success to the three angels in my life, Raghad Alrubai, Arwaa, and Areej, who have brought me up of every frustrating day that I have witnessed through my research. They have supported me with kindness and love when they were suffering through tough times themselves living away from home and needing my support.

Great thanks to my relative family members, who have provided helpful and supportive counsel, continually praying and encouraging me. Most especially to my mother, mother in law, father in law, my brothers and sisters who were always reminding me of what matters the most.

Finally, I would like to thank the Iraqi Government, the Ministry of Higher Education, and the University of Technology-Baghdad, who have initiated, administrated and sponsored my study throughout this scholarship.

Wajdi Al-Rikabi

Abstract

Fuzzy type-2 controllers can easily deal with systems nonlinearity and utilise humans' expertise to solve many complex control problems; they are also very good at processing uncertainty, which exists in many robotic systems, such as autonomous vehicles. However, their computational cost is high, especially at the type reduction stage. In this research, it is aimed to reduce the computation cost of the type reduction stage, thus to facilitate faster performance speed and increase the number of actions able to be operated in one microprocessor. Proposed here are adaptive integration principles with a binary successive search technique to locate the straight or semi-straight segments of a fuzzy set, thus to use them in achieving faster weighted average computation. This computation is very important because it runs frequently in many type reductions. A variable adaptation rate is suggested during the type reduction iterations to reduce the computation cost further. The influence of the proposed approaches on the fuzzy type-2 controller's error has been mathematically analysed and then experimentally measured using a wall-following behaviour, which is the most important action for many autonomous vehicles. The resultant execution time-gain of the proposed technique has reached to 200%. This evaluated with respect to the execution time of the original, unmodified, type reduction procedure.

This study develops a new accelerated version of the enhanced Karnik-Mendel type reducer by using better initialisations and better indexing scheme. The resulting performance time-gain reached 170%, with respect to the original version.

A further cut in the type reduction time is achieved by proposing a One-Go type reduction procedure. This technique can reduce multiple sets altogether in one pass, thus eliminating much of the redundant calculations needed to carry out the reduction individually.

All the proposed type reduction enhancements were evaluated in terms of their execution time-gain and performance error using every possible fuzzy firing level combination. Tests were then performed using a real autonomous vehicle, navigates in a relatively complex arena field with acute, right, obtuse, and reflex angled corners, to assure evaluating wide variety of operation conditions.

A simplified state hold technique using Schmitt-trigger principles and dynamic sense pattern control was suggested and implemented to assure small rule base size and to obtain more accurate evaluation of the type reduction stages.

Chapter 1: Introduction

1.1. Introduction

The autonomous vehicles can perform complex functions, each of them is consisting of one or more of the basic operations such as obstacle avoidance; path planning; trajectory following; localization; environment mapping; objects detecting and identifying. These different operations are mostly performed in dynamic, noisy and uncertain environments. In these conditions, the humans perform very well, therefore utilising components of humans performance expertise is essential to produce an autonomous vehicle controller with a similar capacity. This can be performed in fuzzy controllers where the human expertise can be captured in their rule-bases. It is possible, using fuzzy techniques, to model and process the uncertainty, which can exist in autonomous vehicles, their surrounding environments, and the degree of the human experts. This possibility arose from the simple act of defining variables like hot, cold, and fast using fuzzy sets. However, uncertainty processing and modelling in fuzzy systems has been extended by proposing the fuzzy type-2 (FT2) sets, where the sets boundaries are now fuzzy. Using these fuzzy type-2 sets in controllers has shown superiority over their competent fuzzy type-1 (FT1) controllers regarding their settling time, system overshoot, and noise immunity. Nevertheless, the fuzzy type-2 systems have a bottleneck at the type reduction stage, where a computationally intensive process has to be performed. This bottleneck limits using FT2 controllers in autonomous vehicles, as an example, when real-time and high-speed actions are required. In this research, the fuzzy type-2 controllers of the autonomous vehicles are to be studied, regarding their type reduction computation costs, to find solutions that can offer higher

execution speed and better accuracy. Autonomous fuzzy type-2 controller types and the associated type reduction methods are surveyed in ensuing sections.

1.2. Fuzzy Type-2 Fundamentals

The fuzzy principles, which were proposed by (Zadeh 1965), comprise an extension to the traditional set theory, where the set boundaries have been extended to a more flexible form describes the gradual features transition. The boundaries of the fuzzy set define how much any set element in a specific set satisfies the set's feature. This satisfaction level is defined using a range of [Zero, One], where zero means the feature does not exist, while one means it does exist. This technique managed to simply describe lingual quantities like freezing, cold, cool, warm, and hot by adjacent sets each has a soft transitional section overlapping its preceding set. The fuzzy description to lingual quantities can have different meanings because of some external factors like human expertise, health, and age, which cannot be justified. Therefore, in 1974, Zadeh suggested the FT2 sets as a solution, basically, to the uncertainty associated with words like very, extremely, fairly, rather, pretty, and quite. The FT2 sets systems, then after, have been developed by many researchers like (Karnik and Mendel 2001), (Mendel and John 2002), and (Zadeh 1975b; Zadeh 1975a) to simplify their design; adaptation flexibility; uncertainty modelling; and uncertainty processing. The FT2 sets can describe uncertainty with a high degree of freedom using primary and secondary membership functions (Mendel et al. 2006). They can be presented, graphically, on three axes to form 3D-figures, as shown in Figure 1-1. Here, the *Y-axis* describes the extension of the fuzzy set elements. The vertical axis, which is commonly symbolled as (U) and called the primary membership function, ranges at [0, 1] and represents every set element primary feature satisfaction. The third axis, which is usually denoted by (\widetilde{U}) and called the secondary membership function, represents the probability of the different satisfaction levels at a specific set point. All the points which have secondary membership functions equal to one is an embedded set called the *principal set* (Karnik et al. 1999). Also, the universe of discourse, where the set elements have primary membership functions greater

than *zero* is called the *fuzzy primary set domain*. Any fuzzy type-2 discrete set element (y_n) can have different satisfaction levels over the uncertainty interval denoted by (J_{y_n}). The different satisfaction levels during this interval can have similar fixed values in the case of interval fuzzy type-2 sets.

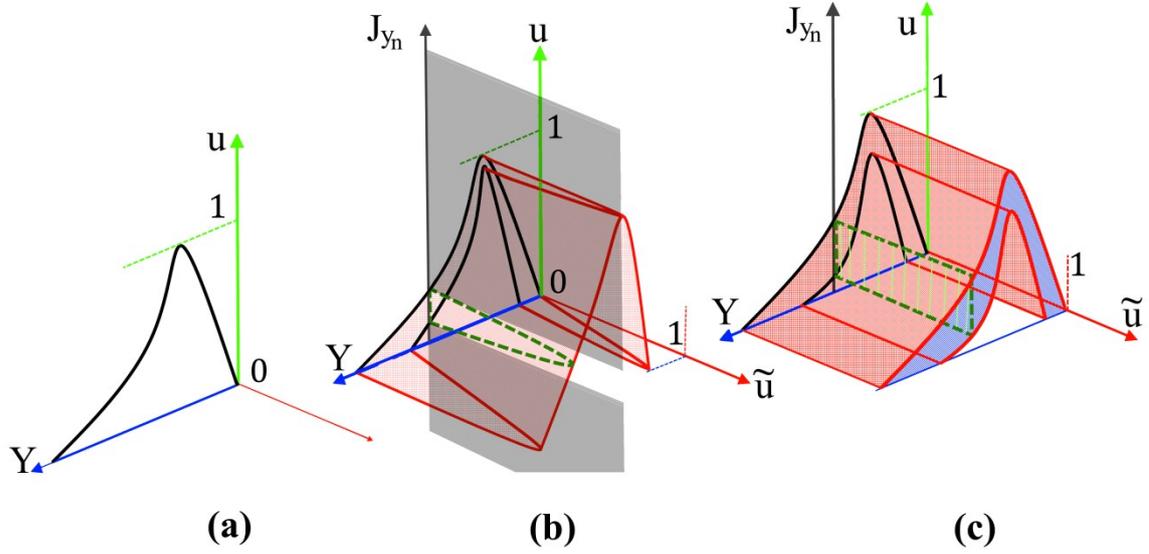


Figure 1-1 : Typical Fuzzy Sets. (a) Type-1. (b) General Type-2. (c) Interval Type-2.

Mathematically, the FT2 set is represented by the union of the set elements, where each set element is described by one tuple. The tuple contains one domain element and its associated membership function $Fu_{\tilde{A}}(y_n)$, as shown in equation (1-1). Here, the summation symbol \sum defines a union, not an addition.

$$\tilde{A} = \sum_{n=1}^N \{y_n, Fu_{\tilde{A}}(y_n)\} \quad (1-1)$$

The FT2 set (\tilde{A}) can be defined, also, in a more detailed form, as shown in equation (1-2) below, where the set domain Y has N elements, each has an uncertainty interval J_{y_n} which describes the possible membership value. If this interval is discretised to M elements, then the secondary membership function can be described as $\tilde{u}_{\tilde{A}}(y_n, J_{y_n}(u_m))$ which define the probability of every possible primary membership value.

$$\tilde{A} = \sum_{n=1}^N \sum_{m=1}^{M_n} \{y_n, J_{y_n}(u_m), \tilde{u}_{\tilde{A}}(y_n, J_{y_n}(u_m))\} \quad (1-2)$$

1.2.1. Interval Fuzzy Type-2 Sets

A simplified form of the FT2 sets has been suggested, mainly, by Mendel (Mendel 2000) to justify the secondary membership to be one value. The secondary membership domain here can be defined by upper membership function (UMF) limit and lower membership function (LMF) limit (Mendel 2007), where they have been denoted as $\overline{u}_{\tilde{A}}(y_n)$ and $\underline{u}_{\tilde{A}}(y_n)$ respectively. These two limits define the footprint of uncertainty (FOU) over the Y-U axis. An *interval FT2* set \tilde{A} can be defined mathematically as:

$$\tilde{A} = \sum_{n=1}^N \sum_{m=1}^{M_n} \{(y_n, J_{y_n}(u_m), 1)\} \quad (1-3)$$

The symbols: N and M_n define the set's primary and secondary elements count, respectively. The interval FT2 sets are used extensively and successfully in many control systems, where they have low associated computation cost and good uncertainty representation (Mendel et al. 2006; Nie and Tan 2012).

1.2.2. Embedded Fuzzy Sets

The embedded sets concept arises from the different uncertainty possibilities that a membership function can take in a FT2 set, which is called the footprint of uncertainty (FOU). In the cases of continuous FT2 set definitions, an infinite total-embedded-sets count (TES) will result (Mendel 2007; Mendel and John 2002). However, for the discretised FT2 sets, for example; \tilde{A} which has N elements; each has an uncertainty interval U discretised to M_n elements; will give a non-repeated count of the maximum

embedded sets that are equal to $\prod_{n=1}^N M_n = (M_1 \times M_2 \times \dots \times M_n \times \dots \times M_N)$, (Mendel 2007). In the case of interval FT2 sets, these embedded sets are all of type-1, but in the case of general FT2 sets, the embedded sets can also be a general FT2 set because there is at least one set element that has a secondary non-zero membership function (Mendel and John 2002). This fact has been utilized to define the general fuzzy set as a union of all the embedded sets. This generates what is known as the general *representation theorem* and the *wave slice theorem* (Mendel and John 2002), where the FT2 set \tilde{A} can be defined in terms of its embedded sets \tilde{A}_e as:

$$\tilde{A} = \bigcup_{\prod_{n=1}^N M_m} (\tilde{A}_e) = \sum_{e=1}^{E=\prod_{n=1}^N M_m} (\tilde{A}_e) \quad (1-4)$$

1.2.3. Fuzzy Controllers

In general, the fuzzy controllers consist of four stages, as shown in Figure 1-2. The first stage is the fuzzifier, which converts the input, physical crisp values, to fuzzy levels, using the input sets membership functions. These generated fuzzy levels are to be used to generate sub-actions using the fuzzy Rule-Base. The fuzzy Rule-Base contains an IF-Then rules shape the controller behaviour. These rules generate individual consequences, which have to be aggregated to get the individual action sets firing levels. Finally, the defuzzification stage has to convert the fuzzy action sets to one crisp output (Kovacic and Bogdan 2010). In the case of FT2 controllers, the inference stage generates a band of consequence levels which require a type reduction operation before the defuzzification stage. The type reduction stage has to convert down the FT2 sets to a FT1 form. This has been done in different ways, but each way has its specific accuracy and computational cost. The accurate type reduction methods are computationally expensive, while the inaccurate ones are fast and computationally cheap. Choosing the appropriate type reduction would depend on many parameters like the required accuracy; the available processing power; the required action speed; application type.

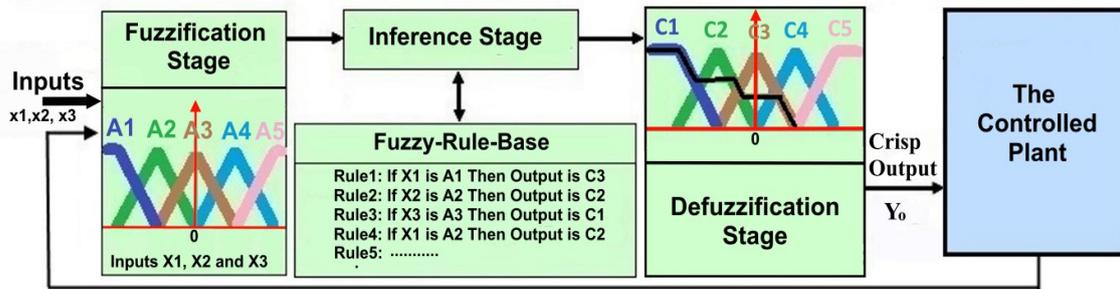


Figure 1-2: Structure of fuzzy type-1 controllers.

1.3. Research Hypothesis

This research is based on the fact that fuzzy type-2 systems are able to represent and resolve high levels of uncertainty, this makes them important for systems that are valuable to high uncertainty through their working environment, system parameters, and sensing data. However, most of those systems are always required to perform at higher speeds and do more functions using smaller digital controllers. The use of fuzzy type-2 controllers in such systems is facing a problem related to the high computation cost at the type reduction stage. This generates a bottleneck, limiting the fields where the benefits of fuzzy type-2 controllers can be harvested.

This research hypothesizes that in between all these complex computations, of the type reduction algorithms, there are some redundant operations can be eliminated to get faster performance without affecting the initial accuracy of the initial algorithms. The adaptive integration is one of the possible techniques, which is going to be investigated for its possible benefits to the type reduction methods that involve centroid computations and/or integrations. A careful study to the important type reduction algorithms can reveal the redundant computations and suggests an accelerated version of them. In addition to that, reducing multi fuzzy sets in one algorithm is another field that will be investigated in this research to eliminate its redundant computations. This approach is very important, especially for the large fuzzy-type-2 controllers.

1.4. Structure of the Thesis

This thesis can be categorized into four main parts, as follows:

- First is the introductory part, which contains Chapter 1 and the literature review of the type reductions, which is presented in Chapter 2. In addition to that, a survey regarding the fuzzy type-2 controllers that's being used for autonomous ground vehicles during the past 10 years, is conducted in chapter 3,.
- The second part describes the use of the adaptive integration in the iterative Karnik-Mendel type reduction procedure. It contains Chapter 4, which explains the related mathematical aspects and derives the error control relation. In Chapter 5, simulations and the practical experiments are conducted to evaluate the resulting time-gain and the generated error due to the proposed modifications.
- The third part contains Chapter 6, which presents an accelerated version of the Enhanced Karnik-Mendel type reduction with all the necessary tests and evaluations. In addition to this, at the end of Chapter 6, a new type reduction procedure, based on eliminating the redundancy out of the multiset reduction, is proposed, analysed, and evaluated.
- The fourth part covers the implementation aspect, which is in chapter 7, then followed by the thesis conclusions, contributions, and future work, which is seen in chapter 8.

Chapter 2: Fuzzy Type Reductions

2.1. Fuzzy Type-2 Sets Reduction

Through this section, type reduction processes (TRPs) are going to be surveyed and investigated for their computation accuracy, complexity, and ability to be solved using parallel processing. In general, any associated complex computations can form a big barrier that affects the usage of fuzzy type-2 controllers for high-speed systems. The TRP procedures, in general, are an extension of the defuzzification process, and therefore inherit their high computation cost. The defuzzification is described in general in the next section.

2.2. Defuzzification

In fuzzy systems, the defuzzification is a process that maps the type-1 set to a crisp value, sometimes called type-0 (Karnik and Mendel 1998). Many defuzzification methods exist with different accuracy levels and computational complexity (George J. Klir and Bo Yuan 1995). The most common defuzzification type is the centre of mass (COM), which is identical to the centre of gravity COG and the centre of area. Another common method is the bisecting method (COA/BSM), which divides the consequent set into two equal sections at the defuzzified point. The centre of sum (COS) (Karnik and Mendel 1998) is another method that adds the sub-consequent sets together then applies the COG process. The quality measure, defuzzification method, (QM) considers the consequent set shape changing in the weighted average COG computation. The level

grading method (LGM), which is very much like the QM method but considers the certainty level at the consequence set firing level, therefore, they are both categorized as a modified height defuzzification (Karnik and Mendel 1998). Some defuzzification methods exhibit rough behaviour like the mean of maximum (MOM); the larger of maximum (LOM); and the smaller of maximum (SOM) different versions. Such rough defuzzification methods are mostly not recommended for many control applications (Runkler 1996; Skulavik et al. 2013). Choosing a proper defuzzification technique is totally dependent on the nature of the application, where some defuzzification techniques jump between solutions and others perform more gradual interpolated transitions. For example, if a fuzzy inferring machine needed to decide the fuel type, which required for a specific machine, then it has to give an answer like gasoline or diesel or kerosene. The answer would be considered incorrect if it is going to be: use 20% gasoline and 80% diesel. But, this soft mix between the individual solutions is very important for electro-mechanical and robotics control systems to achieve stable and soft actions (Runkler 1997; Skulavik et al. 2013). The smoothness of the relation between the defuzzified values and the rules firing levels is known as the defuzzification transfer characteristic. It has been evaluated in many studies, such as (Halgamuge et al. 1996; Runkler 1997; Runkler 1996; Saade and Diab 2000; Saade and Diab 2004), for different defuzzification methods, to show that the COG and centre of area (COA) define the most preferred defuzzification techniques in control systems. These two methods are described mathematically in equations (2-1),(2-2) for fuzzy set C discretised to N elements. The centre of area is based on finding a vertical line that divides the set into two equal parts:

$$COG \text{ of set } C = C_o = \frac{\sum_{n=1}^N u(c_n) * C_n}{\sum_{n=1}^N u(c_n)} \quad (2-1)$$

$$\text{The center of area condition: } \sum_{n=1}^{n=COA} u(c_n) = \sum_{n=COA+1}^N u(c_n) \quad (2-2)$$

2.3. Type Reduction Principles

The type reduction is an extension of the defuzzification process where it is first has to convert the FT2 set to its equivalent FT1 set. The principles of embedded sets are the key ideas behind the type reduction processes (TRP). In general, any FT2 set can be defined in terms of its FT1 embedded sets and every embedded FT1 set can be defuzzified to one crisp value. These defuzzified points are to be used to form one FT1 set, which represents the type reduced set. This principle is logically simple, but computationally it is expensive because the total embedded sets count is high. For example, the total embedded sets count is $(TES) = \prod_{n=1}^N M_n$, and in the case of a general FT2 set \widetilde{A} contains N elements, each has uncertainty interval of J_{y_n} discretised to M_n points. Therefore, the total computation, which is required to defuzzify all these embedded FT1 sets, is very high even when the cost per one set is low. This process is described by equation (2-3) to generate a FT1 set $C_{\widetilde{A}}$ (Karnik and Mendel 2001). Here, every point inside every primary membership function J_{y_n} , as shown in Figure 1-1, has to be used to form the different embedded sets, as described by the first part of the equation. Each embedded set has to be defuzzified, as shown in the last part of equation (2-3). Any defuzzified point has to take the smallest membership value, between all the points that have been used in its embedded set. This operation is described by the fuzzy t-norm operation, which is usually symbolized by asterisk \star . Throughout the computation, many embedded sets may have similar defuzzification results but with different membership values. In these cases, only the highest membership values are to be taken.

$$C_{\widetilde{A}} = \int_{j_{1m} \in J_{y_1}}^{M_1} \dots \int_{j_{nm} \in J_{y_n}}^{M_1} \dots \int_{j_{Nm} \in J_{y_N}}^{M_N} [f_{J_{x_1}}(j_{1m}) \star \dots \star f_{J_{y_n}}(j_{nm}) \star \dots \star f_{J_{y_N}}(j_{Nm})] / \frac{\sum_{n=1}^N y_n \cdot j_n^*}{\sum_{n=1}^N j_n^*} \quad (2-3)$$

Another discretised form is proposed by (Coupland 2007) to calculate the type reduced set, as follows:

$$C_{\bar{A}} = \underbrace{\sum_{e=1}^{TES=\prod_{n=1}^N M_n}}_{\text{I}} \left[\underbrace{\sum_{n=1}^N \star}_{\text{II}} \mu_{\bar{A}}(j_{n,m} \Big|_{m=1}^{M_n}, y_n) \right] \Big| \underbrace{\frac{\sum_{i=1}^N j_{n,m} \cdot y_n}{\sum_{i=1}^N j_{n,m}}}_{\text{III}} \quad (2-4)$$

Here, the first summation means that calculations have to include all the possible embedded sets, each is symbolled as e . These sets range from one to the total number of embedded sets, $TES = \prod_{n=1}^N M_n$. The middle part defines how to calculate the defuzzified point membership values using a t-norm operation (Karnik and Mendel 2001). The third, right, part is the defuzzification process, for every embedded set. The highest membership function has to be considered if more than one value is being generated for one set element. Choosing the highest membership function is not described mathematically in equations (2-3) and (2-4) therefore a notice has been given. Here, high redundancy does exist in these TRP techniques; therefore called exhaustive TRP methods. The computational cost of these *exhaustive* TRP is $(N-1)$ t-norm operations plus N multiplications plus $2(N-1)$ additions plus one division, which all have to be repeated for $(TES = \prod_{n=1}^N M_n)$ times. The resulting centroids of these methods are considered ideal, but are used for referencing only, not for practical real time applications. Those expensive computations have been reduced very much by suggesting simplified FT2 sets, which are called the interval FT2 set. Here, most of the computational efforts, which have been used to processes the enormous embedded sets, are eliminated because of the possibility to process only the boundaries of the interval FT2 sets to get the same result of the exhaustive TRP methods. The traditional exhaustive TRP formula has been re-written by (Karnik and Mendel 2001) to match the interval FT2 sets where the secondary membership value becomes equal to one, $\mu_{\bar{A}}(j_{n,m} \Big|_{m=1}^{M_n}, y_n) = 1$, to get the type reduced set as follows:

$$C_{\bar{A}} = \sum_{e=1}^{TES=\prod_{n=1}^N M_n} \left[\sum_{n=1}^N \star 1 \right] \Big| \frac{\sum_{i=1}^N j_{n,m} \cdot y_n}{\sum_{i=1}^N j_{n,m}} = [C_L, C_R] \quad (2-5)$$

Here, the two points $[C_L, C_R]$ define the start and the end of the resulted FT1 set. Also, they represent the leftmost and the rightmost uncertainty extreme points. All the other points in between these points will have equal membership levels, thus there is no necessity to compute them (Karnik and Mendel 2001). This general type reduction principle will be expanded during the ensuing sections, where all the type reduction procedures (TRP) that have been proposed to reduce interval and general FT2 sets since 2001 are going to be studied and reviewed. This date is selected because the development of fuzzy control systems (FCS) has witnessed a dramatic increase since then.

2.4. Interval Fuzzy Sets Type Reductions

The importance of the interval FT2 sets arises from their good uncertainty representation combined with their relatively fast type reduction methods. Many type reduction methods have been developed for interval FT2 sets, aimed to reduce computation cost and enhance accuracy. However, some of them are sacrificing accuracy and embracing generality for the sake of high computational speed. Therefore, choosing a suitable type reduction method requires a review of the current methods and their features.

2.4.1. Iterative Karnik & Mendel (IKM) Type Reduction

Iterative search technique has been proposed by (Karnik and Mendel 2001) as a type reduction procedure (TRP) to convert the interval FT2 to an interval FT1 set defined only by its start and end points. This procedure is based on forming an embedded set having two joined parts taken from the footprint of uncertainty (FOU) that associated with the interval FT2 set. The first part is the lower edge of the footprint of uncertainty and the second part is the upper edge of the FOU. Two embedded sets are formed by joining two different edges of the FOU using two temporary movable switching

points S_L and S_r . One of these embedded sets is required to find the leftmost switching point, which is usually symbolled as (Y_L) . This set is formed by the upper edge of the FOU, symbolled \overline{FOU} , that is on the left side of the temporary switching point S_L , and the lower edge, symbolled \underline{FOU} , which is on the right side of this switching point. The embedded set, which is going to be used to locate the rightmost switching point (Y_R) , is formed by the \underline{FOU} on the left side of the rightmost temporary switching point S_r and by the \overline{FOU} that is falling on the right side of this temporary switching point. Defuzzifying these two embedded sets independently would generate two new switching points, y_l and y_r . These two points are to be used as the new switching point to construct two new embedded sets for the left and right uncertainty points. Performing these operations, of constructing new embedded set using the old results, and defuzzifying it to get new point, iteratively will settle to one point represents either the leftmost or the right most uncertainty point, based on the used embedded set. The iterations will be terminated when the new points become almost equal the old points. The required stop condition is usually in the form of: IF $(y_n - y_{n-1}) \leq \varepsilon$ THEN STOP (Karnik and Mendel 2001). The value ε represents a small acceptable error. The used embedded sets for this type reduction are shown in Figure 2-1 with their associated calculations. This is described in equations (2-6) (2-7) for a discrete interval FT2 set defined using N elements.

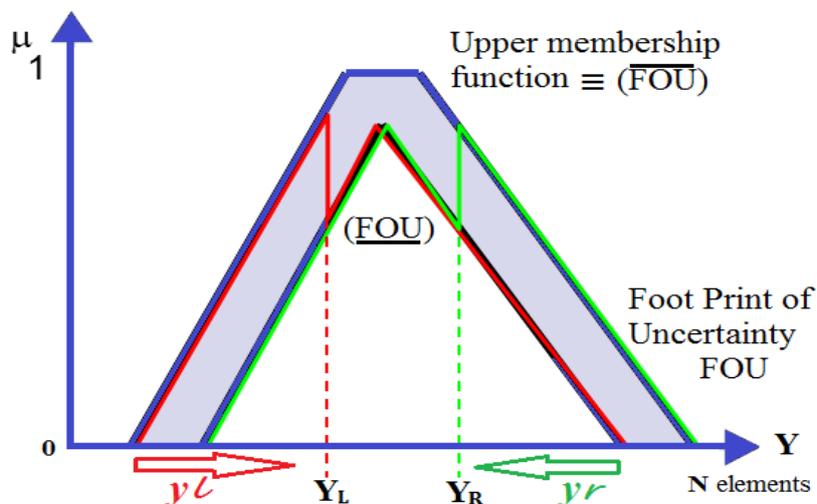


Figure 2-1 : Iterative KM type reduction over IFT2 set.

$$y_l = \frac{\sum_{n=1}^{S_L} (\overline{\mu}_n \cdot y_n) + \sum_{n=S_L+1}^N (\underline{\mu}_n \cdot y_n)}{\sum_{n=1}^{S_L} (\overline{\mu}_n) + \sum_{n=S_L+1}^N (\underline{\mu}_n)} \quad (2-6)$$

$$y_r = \frac{\sum_{n=1}^{S_r} (\underline{\mu}_n \cdot y_n) + \sum_{n=S_r+1}^N (\overline{\mu}_n \cdot y_n)}{\sum_{n=1}^{S_r} (\underline{\mu}_n) + \sum_{n=S_r+1}^N (\overline{\mu}_n)} \quad (2-7)$$

This resulted interval FT1 set defines the uncertainty associated with the output interval FT2 decision set. It can be defuzzified by averaging its two ends, as:

$$\text{Fuzzy Type2 Output} = \frac{y_l + y_r}{2} \quad (2-8)$$

The iterations that are required for each end to converge can be N iterations in worst case, but the average is $(N/2)$ iterations for initial start points chosen as the centroid of the average set: $\frac{1}{2}(\overline{FOU} + \underline{FOU})$, (Wu and Mendel 2009) or at the middle of the output set range $(\frac{N}{2})$. This TRP consumes a total of $(N \times (N \cdot \text{multiplication} + 2N \times \text{additions}))$ plus the defuzzification computation cost to generate one crisp output.

2.4.2. Approximated KM Type Reduction (AKM)

In (Karnik and Mendel 2001) a new method is proposed to approximate the type reduction results. The approximation is evaluated without passing through the type reduction iterations only for those fuzzy sets having uniform small uncertainty (Δ_{J_n}) value associated with their elements (y_n) . This approximation is performed by representing each vertical interval (J_{y_n}) using its central element (CJ_{y_n}) and its local uncertainty spread (Δ_{J_n}) . Now, for an output FT2 set, \tilde{Y} , any of its fuzzy elements (y_n) will be represented as: $J_{y_n} = (CJ_{y_n} \mp \Delta_{J_n})$. The next step is to form one embedded set called the centre of average set which contains all these central elements. This set is to be defuzzified to one point as in (2-9). The associated uncertainty with this point is approximated in equation (2-10) using all the uncertainty spreads.

$$Y_o = \frac{\sum_{n=1}^N y_n C_{J_{yn}}}{\sum_{n=1}^N C_{J_{yn}}} \quad (2-9)$$

$$S = \frac{\sum_{n=1}^N |y_n - Y_o| \Delta_{J_n}}{\sum_{n=1}^N C_{J_{yn}}} \quad (2-10)$$

The type reduced set is formed by $C_L = Y_o - S$ and $C_R = Y_o + S$. The approximation shows that the uncertainty is symmetry around the approximated center, which is not the case in general. The computational cost is $2N$ multiplications; two divisions; and $4N$ additions.

2.4.3. Wu-Mendel (Max-Min) Approximation

Another approximated type reduction procedure proposed by (Mendel 2002) is based on calculating the centre of set for the lower and upper membership functions, $\underline{f}_{(y)}$ and $\overline{f}_{(y)}$, separately, then arranging these centres in ascending order such that $\underline{y}_l = \min(C_{LMF}, C_{UMF})$ and $\underline{y}_r = \max(C_{LMF}, C_{UMF})$. These two points are considered to be the inner uncertainty boundaries for the type-reduced set. The outer uncertainty boundaries \underline{y}_l and \overline{y}_r , are defined by calculating what are termed the extra uncertainty limits, then adding those extra uncertainty limits to the existing inner boundaries, as:

$$\underline{y}_l = \underline{y}_l - \frac{\sum_{i=1}^N (\overline{f_{yi}} - \underline{f_{yi}})}{\sum_{i=1}^N \overline{f_{yi}} \cdot \sum_{y=1}^N \underline{f_{yi}}} \cdot \frac{\sum_{i=1}^N (\underline{f_{yi}} \cdot (y_i - y_1)) \cdot \sum_{i=1}^N (\overline{f_{yi}} \cdot (y_N - y_i))}{\sum_{i=1}^N (\underline{f_{yi}} \cdot (y_i - y_1)) + \sum_{i=1}^N (\overline{f_{yi}} \cdot (y_N - y_i))} \quad (2-11)$$

$$\overline{y}_r = \underline{y}_r + \frac{\sum_{i=1}^N (\overline{f_{yi}} - \underline{f_{yi}})}{\sum_{i=1}^N \overline{f_{yi}} \cdot \sum_{y=1}^N \underline{f_{yi}}} \cdot \frac{\sum_{i=1}^N (\overline{f_{yi}} \cdot (y_i - y_1)) \cdot \sum_{i=1}^N (\underline{f_{yi}} \cdot (y_N - y_i))}{\sum_{i=1}^N (\underline{f_{yi}} \cdot (y_i - y_1)) + \sum_{i=1}^N (\overline{f_{yi}} \cdot (y_N - y_i))} \quad (2-12)$$

Two of those four points are used to approximate one end of the ideal type reduced set as follows: $y_L = (\overline{y}_l, \underline{y}_l)/2$ and $y_R = (\overline{y}_r + \underline{y}_r)/2$. The final defuzzified point is calculated by averaging these two ends ($C_o = \frac{y_L + y_R}{2}$), as shown in Figure 2-2. This type reduction, using the uncertainty bounds, is called the Max-Min uncertainty bounds

method (Mendel and Liu 2008). It has achieved an acceptable real time control performance comparable to neural network type reduction techniques (Lynch et al. 2006). Here, (5N) multiplications and (10N) addition-subtraction operations are used to calculate one crisp output.

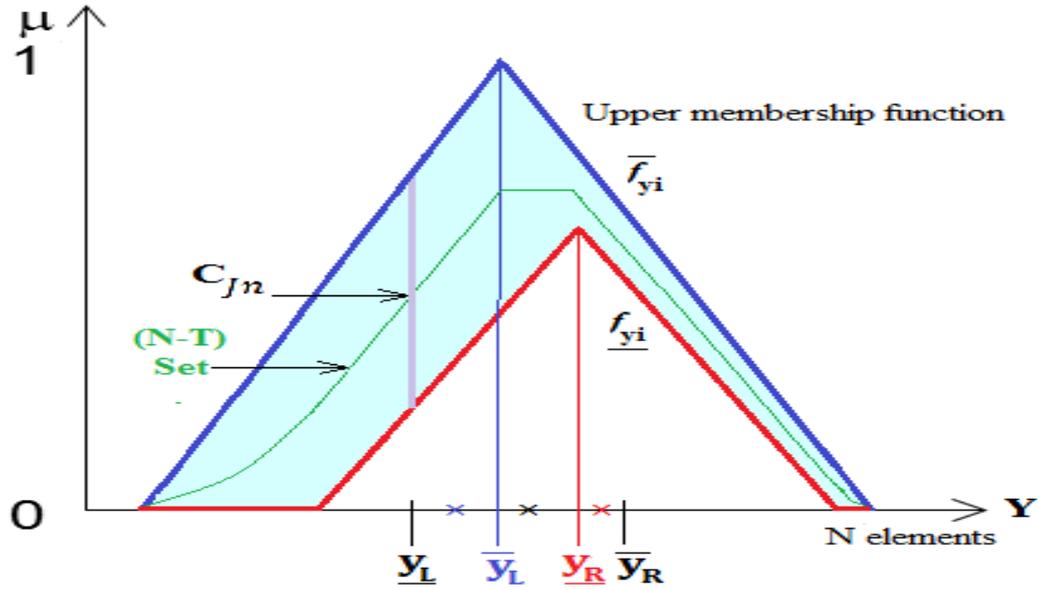


Figure 2-2: Wu-Mendel Approximation Method and N-T set.

2.4.4. Wu-Tan Type Reduction

Another type reduction procedure is presented by (Wu and Tan 2005) to overcome the type reduction bottleneck. It is based on performing a type-reduction-like process, prior to the inferring stage, of the fuzzy controllers. Here, each vertical interval at the antecedent FT2 sets is going to be replaced by one equivalent point $f_{eq}(x_n)$. Computation of these equivalent points is performed through an empirical formula, described by equation (2-13). This equation uses what have been termed as rate-correction parameters to compensate for the effects of other fuzzy sets at the input stage.

$$f_{eq}(x_n) = \bar{F}_n - \frac{1}{Se} \sum_{i=1}^{Se} rate_i \cdot x_i \cdot (\bar{\mu}_{x_n} - \underline{\mu}_{x_n}) \quad (2-13)$$

Here, $f_{eq}(x_n)$ is the point that is going to replace the n th input set interval. The $\overline{\mu_{x_n}}$ and $\underline{\mu_{x_n}}$ are the intervals upper and lower membership boundaries. The symbol Se is the count of the antecedent input sets used in the fuzzy controller. The symbol x_i represents the i^{th} input set firing level, which is affected by the controller inputs and the input set shapes. The symbol $rate_i$ defines those correction factors, which are used to compensate the effects of the other input fuzzy sets. The correction factors have to be evaluated for every set using genetic technique to utilise a cost function based on the integration of a time-weighted absolute system error. However, these factors can be easily estimated for simple set shapes. As shown in Figure 2-3, if the fuzzy systems have only basic trapezoidal sets, then its $rate_i$ factors can be approximated by: $\frac{2}{P_{i1}-P_{i2}}$ (Wu and Tan 2005), where P_{i1} and P_{i2} represents the left and the right apex points in each trapezoidal sets.

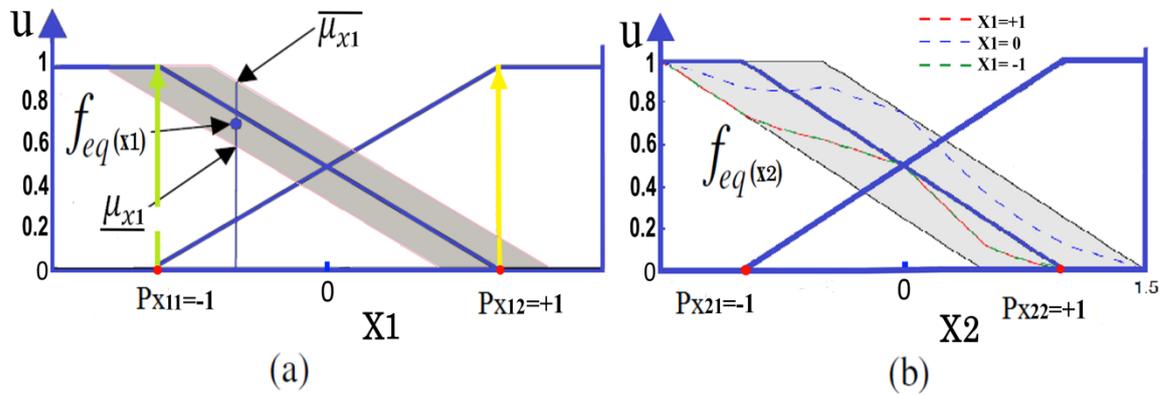


Figure 2-3: Example of interval FT2 sets reduction using Wu-Tan method. (a) Is the first input set: X_1 . (b) Is the second input set, X_1 , with its equivalent sets $f_{eq}(x_2)$ being evaluated at three different input levels: $X_1 = \{+1, 0, -1\}$.

The computation cost of this approximation is low, very much similar to this associated with FT1 systems. However, the uncertainty measure is completely missing and the $rate_i$ parameters are hard to evaluate for complex systems.

2.4.5. The Nie-Tan (N-T) method

The procedure of this type reduction is based on omitting the uncertainty awareness of the interval FT2 sets, but considering the final defuzzified value only (Nie and Tan 2008). It determines the centre of the set using the COG calculates for one embedded set containing all the average points of the vertical uncertainty slices $J_{y_n} = [\bar{u}_{y_n}, \underline{u}_{y_n}]$. The output of this type reduction-defuzzifier is one crisp value, calculated as:

$$Y_o = \sum_{n=1}^N y_n \cdot C_{J_n} = \frac{\sum_{n=1}^N y_n \cdot \left(\frac{\bar{\mu}_{j_n} + \mu_{j_n}}{2}\right)}{\sum_{n=1}^N \left(\frac{\bar{\mu}_{j_n} + \mu_{j_n}}{2}\right)} \quad (2-14)$$

The non-iterative features, which exist here, enable for more fuzzy systems analyses and tuning operations (Nie and Tan 2008). This type reduction formula gives smoother control surface in compare to IKM TRP. The computation cost is $3N$ additions, and $2N$ multiplications and divisions. This is considered a very low cost in compare to the other type reduction techniques.

2.4.6. Collapse Type Reduction

The collapse type reduction works to generate a FT1 set that has a centre of set similar to its unreduced original FT2 set (Greenfield, Chiclana, Coupland, et al. 2009). The reduction principle is based on choosing one special interval set that has one blurred point, let's say at y_i , where the blurring level is equal to: $b_i = UMF_i - LMF_i$, as shown in Figure 2-4. This special interval set has an identical LMF and UMF except at the blurred point. A temporary reprehensive set (R_i), of type-1, is going to be derived such that its centre of set is similar to the special interval set, which has one blurred point. This representing set has been chosen to be similar to the lower LMF except at the blurred point, where the membership value r_i is calculated to be:

$$r_i = \frac{b_i \cdot \|L\|}{b_i + 2 \cdot \|L\|} \quad , \quad \text{The scalar cardinality is } \|L\| = \sum_{n=1}^N \mu_{LMF}(y_n) \quad (2-15)$$

Here, the symbol $u_{LMF}(y_i)$ represents the FT2 lower membership values. Repeating the operation of choosing a blurred set and finding its represented FT1, while considering the converted and the non-converted parts of the initial FT2 set, will lead to reducing the whole FT2 set, point by point successively, to one equivalent FT1 set. This is described in equation (2-16), where the symbol $uR_k(y_i)$ represents the temporary equivalent set at the i^{th} point, on the k^{th} cycle. As shown in Figure 2-4, the process started from the left-most elements and ended at the point y_i . The collapse type reduction may generate different representing sets, but all of them will have symmetrical centres. This may happen if different indexing patterns are used, as shown in Figure 2-4 part (a), for backward, inward or outward indexing patterns (Greenfield, Chiclana and John 2009).

$$uR_k(y_i) = u_{LMF}(y_i) + \frac{b_i \cdot (\|L\| + \sum_{j=1}^{i-1} r(y_j))}{b_i + 2 \cdot (\|L\| + \sum_{j=1}^{i-1} r(y_j))} \quad (2-16)$$

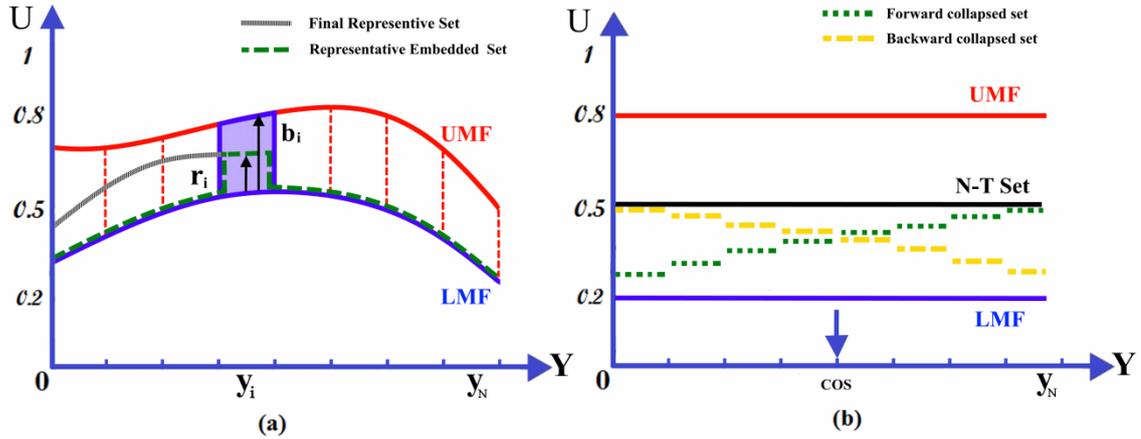


Figure 2-4 : (a) Collapse type reduction principles. (b) Different progress patterns effect.

The collapse type reduction will generate a Nie-Tan set, described on page 19, if a very fine set discretization level is used (Greenfield 2012). This happens because the $\|L\|$ value will become very high. This makes the value $\left(\frac{b_i}{\|L\|}\right)$ equal to zero, as shown below, generating r_i definition similar to that in Nie-Tan set, which uses the set's mid-points.

$$r_i = \frac{b_i}{2 + \left(\frac{b_i}{\|L\|}\right)} \Rightarrow r_i = \frac{b_i}{2} \quad (2-17)$$

The uncertainty measure in the collapse procedure is omitted, like the N-T type reduction technique. The computations required here are $3N$ multiplication and divisions, plus $4N$ additions to get the final representative set. The conversion to crisp value will require $2N$ additions and N multiplications, if the COG defuzzification is being used, which is preferred in control systems.

2.4.7. Enhanced Karnik–Mendel (EKM) Type Reduction Procedure

The type reduction cost has been reduced noticeably in the procedure proposed in (Wu and Mendel 2007). Much of the redundant computations of the KM type reduction have been eliminated by suggesting two main enhancements (Wu and Mendel 2009) (Maowen 2011). The first enhancement is achieved by proposing better initial switching points, $\frac{N}{2.4}$ for the left and $\frac{N}{1.7}$ for the right. This is made instead of the set average point, which had been used in the iterative KM procedure, as shown in equation (2-18).

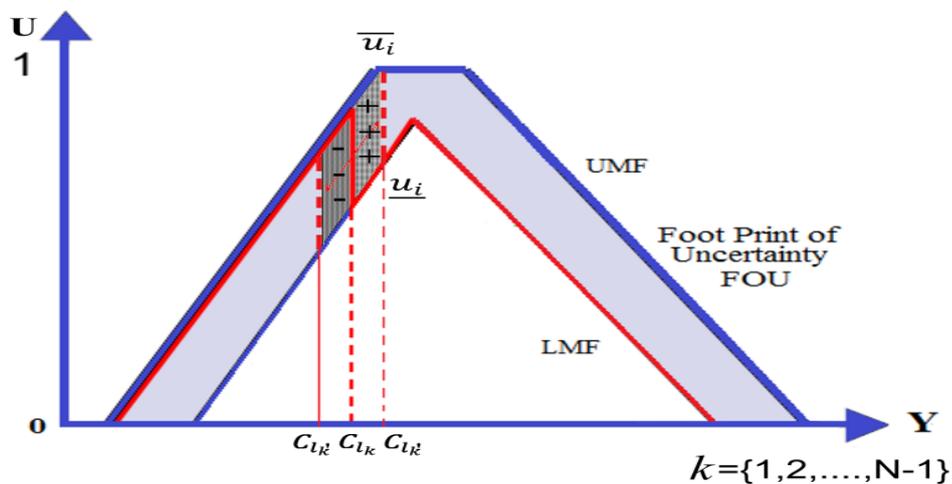


Figure 2-5 : EKM iterations principles.

Using this initialization has led to fewer iterations, because these new starting points have been deduced using statistical analysis for different set shapes and their type reduction results, thus they are very close to the real solutions. Using these points may require some extra tuning in cases of complex or non-common set shapes (Wu and Mendel 2009).

The second enhancement has achieved by eliminating some redundant computations out of the KM type reduction, especially after the first KM iteration. Here, the suggestion is to make the first iteration in full, as shown in equations (2-19). Then, all of the following iterations are going to use the previous result and a correction value, which has to be added or subtracted, according to the difference between the current switching point and its predecessor, as shown in equation (2-20) and Figure 2-5.

$$C_{L0} = C_{R0} = \frac{\sum_{i=1}^N y_i (\overline{u_{y_i}} + \underline{u_{y_i}})/2}{\sum_{i=1}^N (\overline{u_{y_i}} + \underline{u_{y_i}})/2} \quad (2-18)$$

$$C_{l_0} = \frac{a}{b} = \frac{\sum_{i=1}^k \overline{u_{y_i}} \cdot y_i + \sum_{i=k+1}^N \underline{u_{y_i}} \cdot y_i}{\sum_{i=1}^k \overline{u_{y_i}} + \sum_{i=k+1}^N \underline{u_{y_i}}} \quad (2-19)$$

$$C_{i_{k'}} = \frac{a + \mathbf{sign}(k' - k) \sum_{i=\min(k',k)}^{\max(k',k)} y_i \cdot (\overline{u_{y_i}} - \underline{u_{y_i}})}{b + \mathbf{sign}(k' - k) \sum_{i=\min(k',k)}^{\max(k',k)} (\overline{u_{y_i}} - \underline{u_{y_i}})} \quad (2-20)$$

The total computational cost required to find the left centroid is $2N$ additions and N multiplications and one division plus the calculations required for a move from the initial start point, $N/2.4$, to zero, which in the worst case is about $4 \times (N/2.4)$ additions-subtractions, plus $N/2.4$ multiplications. Widespread acceptance of this EKM type reduction procedure has encouraged researchers to solve many related issues, as for example (Garcia 2012), which has suggested using interpolation to discretise interval FT2 sets that have missing vertical interval ends, $\{\overline{J_x}, \underline{J_x}\}$, which can result during a horizontal alpha cut process.

2.4.8. EKM Type Reduction Using Incremental Formula

The EKM has been reformulated by (Duran et al. 2008) to an *incremental formula* (EKM-IF) with enhanced stop condition, which eliminates search fluctuation around the final left and right switching points. Fluctuation can result because of digitization error and the convexity nature of the search space, which can exist in almost all fuzzy sets. The practical tests performed by (Liu et al. 2012) had shown that fluctuation happens frequently in the EKM algorithm, which is demanding bulk computations for redundant operations. The incremental formula starts by calculating separately the numerator and the denominator of the centroid formula of the upper and lower membership functions, as in equations (2-21) and (2-22).

$$\frac{\overline{Rnum}_0}{\overline{Rden}_0} = \frac{\sum_{i=1}^N y_i \cdot \overline{u_{y_i}}}{\sum_{i=1}^N \overline{u_{y_i}}} \quad (2-21)$$

$$\frac{\underline{Lnum}_0}{\underline{Lden}_0} = \frac{\sum_{i=1}^N y_i \cdot \underline{u_{y_i}}}{\sum_{i=1}^N \underline{u_{y_i}}} \quad (2-22)$$

At initialization, these nominators and denominators are given index zero. Then a sequential left and right switching points search have to be performed, as shown in Figure 2-6, giving the first resulting left and right centroids an index $k=1$, as shown in equations (2-23) and (2-24).

$$\begin{aligned} \overline{Rnum}_k &= \overline{Rnum}_{k-1} - y_k \cdot (\overline{u_{y_k}} - \underline{u_{y_k}}), \quad \overline{Rden}_k \\ &= \overline{Rden}_{k-1} - (\overline{u_{y_k}} - \underline{u_{y_k}}) \end{aligned} \quad (2-23)$$

$$\begin{aligned} \underline{Lnum}_k &= \underline{Lnum}_{k-1} + y_k \cdot (\overline{u_{y_k}} - \underline{u_{y_k}}), \\ \underline{Lden}_k &= \underline{Lden}_{k-1} + (\overline{u_{y_k}} - \underline{u_{y_k}}) \end{aligned} \quad (2-24)$$

$$C_{Rk} = \frac{\overline{num}_k}{\overline{den}_k}, \quad C_{Lk} = \frac{\underline{num}_k}{\underline{den}_k} \quad (2-25)$$

The sequential search operation ends when minimum C_{Lk} and maximum C_{Rk} points are found. The computation cost is $9N$ additions and $3N$ multiplications.

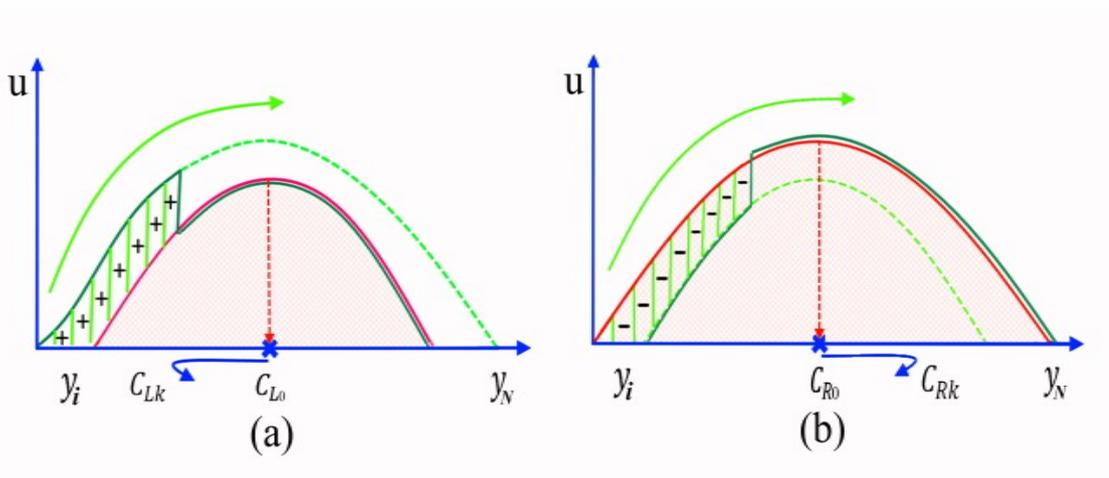


Figure 2-6: EKM Type Reduction in Incremental Formula. (a) Left switching point search. (b) Right switching point search.

2.4.9. Type Reduction Using Type-1 OWA

The ordered weighted average (OWA) operation is a tunable fuzzy aggregation process suggested by (Yager 1988). It is based on a unit weighting operator $W = \{\omega_1, \omega_2, \dots, \omega_N\}^T$ consisting of sub weights; each of them is in the range of $\omega_i \in [0,1]$, while their summation is one $\sum_{i=1}^N \omega_i = 1$. Processing an (n) variables vector $\{a_1, a_2, \dots, a_n\}$ in OWA means generating one crisp output by the process $\sum_{i=1}^N \omega_i a_{\sigma(i)}$, where $\sigma(i)$ is a sorting operator, which arranges the input vectored elements a_i 's, such that $a_{\sigma(i)}$ is the i -th largest element in the set. Examples of the OWA operators are maximum operator $W \uparrow = \{1,0,0, \dots\}^T$; minimum operator $W \downarrow = \{\dots,0,0,1\}^T$; and average operator $W \sim = \{\frac{1}{N}, \dots, \frac{1}{N}\}^T$.

This aggregation process has been extended by (Zhou et al. 2008) to aggregate FT2 sets. The new extended aggregation is called type-1 OWA. Here the operator now is a two-dimensional array described as linguistic fuzzy sets, $\{W_i\}_{i=1}^N$, using the unity universe of discourse. The lingual fuzzy sets $\{A_i\}_{i=1}^N$, which are going to be processed, can be defined over their localized universes X_k without the need to re-map them to the unity universe of discourse. The result of processing N fuzzy type-1 sets using the type-

1 OWA operation is N points, which define one FT1 set, Y . These N points and their related membership values have been evaluated by (Zhou et al. 2008), as follows:

$$y = \sum_{i=1}^N \bar{\omega}_i a_{\sigma(i)} \mid \omega_i \in \text{support } W_r, a_i \in \text{support } A_k \quad (2-26)$$

$$\text{Where } \bar{\omega}_i = \frac{\omega_i}{\sum_{i=1}^N \omega_i} \quad (2-27)$$

$$\text{and } \mu(y) = \sup_y \left(\mu_{A_1}(a_1) \star \dots \star \mu_{A_N}(a_N) \star \mu_{W_1}(\omega_1) \star \dots \star \mu_{W_N}(\omega_N) \right) \quad (2-28)$$

Equation (2-27) can be substituted in equation (2-26) to get the formal COG equation, but here, the weighting arrays of the type-1 OWA describe the uncertainty over the vertical set slices, which usually called the primary membership functions.

Using this type-1 OWA to reduce an interval FT2 set, \tilde{A} , which contains N elements will require describing its elements as vectors, $\{A(X)_i\}_{i=1}^N$. Here, each victor is defining the uncertainty, which may exist in the set elements. The weighting operators $\{W_i\}_{i=1}^N$ have to describe the primary and the secondary membership functions as: $W_i = \{\mu_{W_i}(\omega_k) \mid \omega_k \in J_{x_i}\}$, for each set element x_i . A general type-1 OWA operation is performed by (Chiclana and Zhou 2011) as in equation (2-27), where all the possible combinations of a_i and w_i , have to be used to determine the uncertainty boundaries, that define the start and end points of the type reduced set. The computation cost of this operation is similar to the exhaustive TRP.

A better type-1 OWA procedure had been proposed by (Zhou and Chiclana 2009) named the α -cut OWA, which requires defining the sets $\{A_i\}_{i=1}^N$ and the operators $\{W_i\}_{i=1}^N$ to become a group of intervals. That is performed using an equally spaced values of $\alpha \in [0,1]$ to generate these intervals as: $\{W_{i_\alpha}\}_{i=1}^N = \{\omega \mid \mu_{W_i}(\omega) \geq \alpha\}$ and $A_{i_\alpha} = \{x \mid \mu_{A_i}(x) \geq \alpha\}$. The $\{W_{i_\alpha}\}_{i=1}^N$ and $\{A_{i_\alpha}\}_{i=1}^N$ sets have to be aggregated using equations (2-26), (2-27), and (2-28) then a union between the different results of the different α -cut OWA has to be performed to get the final type reduced set $Y = \bigcup_{0 \leq \alpha \leq 1} \alpha \cdot Y_\alpha$. Any of the individual α -cut OWA result set is an interval can

be fully described using its ends, as $Y_\alpha = \{Y_{\alpha-}, Y_{\alpha+}\}$, thus a good computation cost reduction has been achieved by processing these ends only.

$$Y_{\alpha-} = \min_{\substack{W_{\alpha-}^i \leq \omega_i \leq W_{\alpha+}^i \\ A_{\alpha-}^i \leq a_i \leq A_{\alpha+}^i}} \frac{\sum_{i=1}^N \omega_i a_{\sigma(i)} / \sum_{i=1}^N \omega_i}{}, \mu(y) = 1 \quad (2-29)$$

$$Y_{\alpha+} = \max_{\substack{W_{\alpha-}^i \leq \omega_i \leq W_{\alpha+}^i \\ A_{\alpha-}^i \leq a_i \leq A_{\alpha+}^i}} \frac{\sum_{i=1}^N \omega_i a_{\sigma(i)} / \sum_{i=1}^N \omega_i}{}, \mu(y) = 1 \quad (2-30)$$

The ends are determined for each α -level set iteratively (Zhou and Chiclana 2009; Chiclana and Zhou 2013), very similar to IKM algorithm, where the search starts using the first set element, indexed 1, and stops if $\rho_{\alpha-}^s \geq A_{\alpha-}^{\sigma(s)}$ and $\rho_{\alpha+}^s \geq A_{\alpha+}^{\sigma(s)}$, respectively, for the left and the right uncertainty points.

$$\rho_{\alpha-}^s \triangleq \frac{\sum_{i=1}^{s-1} \mu_{W_{\alpha-}}(i) \cdot A_{\alpha-}^{\sigma(i)} + \sum_{i=s}^N \mu_{W_{\alpha+}}(i) \cdot A_{\alpha-}^{\sigma(i)}}{\sum_{i=1}^{s-1} \mu_{W_{\alpha-}}(i) + \sum_{i=s}^N \mu_{W_{\alpha+}}(i)} \quad (2-31)$$

$$\rho_{\alpha+}^s \triangleq \frac{\sum_{i=1}^{s-1} \mu_{W_{\alpha+}}(i) \cdot A_{\alpha+}^{\sigma(i)} + \sum_{i=s}^N \mu_{W_{\alpha-}}(i) \cdot A_{\alpha+}^{\sigma(i)}}{\sum_{i=1}^{s-1} \mu_{W_{\alpha+}}(i) + \sum_{i=s}^N \mu_{W_{\alpha-}}(i)} \quad (2-32)$$

Each $\{W_{i_\alpha}\}_{i=1}^N$ set uses its associated J_{xi} endpoints, thus it is exactly like the IKM algorithm for IFT2 set type reduction with the same computation cost. In general, the type reduction is a special case of the OWA type-1 aggregation (Chiclana and Zhou 2013).

2.4.10. Weighted Enhanced Karnik–Mendel Type Reduction Procedure

The weighted enhanced KM algorithm (WEKM), suggested in (Liu et al. 2012), has achieved computation cut, over the EKM, by two modifications: (i) Using a mathematical formula to estimate better initial start points; (ii) Weighting every primary membership point in a way that can reduce the EKM iterations. The initial point's mathematical formula mainly depends on the difference between the UMF and the LMF,

where $y_{lo} = N/(1 + \sqrt{\rho})$ and $y_{ro} = N/(1 + \sqrt{1/\rho})$. The ρ equals to $\sum_N \bar{\mu} / \sum_N \underline{\mu}$. This formula gives $y_{lo} = N/2.4$ and $y_{ro} = N/1.7$ for the case of $\rho = 2$, which indicates that most of the practical tests that had been performed by Karnik–Mendel in (Wu and Mendel 2009) to estimate the initialization points have this ratio and it is coop with more practical cases. The second enhancement has been achieved by using the weights of the Trapezoidal, Simpson, and 3/8-Simpson integration rules to weight the fuzzy sets elements. In case of Trapezoidal rules, each primary membership has to be weighted by one value, according to its index as $\{w_{i=1,N}=0.5, w_{i \neq N,1}=1\}$. In case of Simpson rule, the following weights are used $\{w_{i=1,N}=0.5, w_{i=even \text{ and } \neq N}=2, w_{i=odd \text{ and } \neq 1}=1\}$. While, the 3/8-Simpson rule requires the weights $\{w_{i=1,N}=0.5, w_{i=even \text{ and } \neq N}=2, w_{i=odd \text{ and } \neq 1}=1\}$ for each μ_{xi} . This weighting process adds more computations to the EKM formula but it leads to getting less iteration to reach the left and right uncertainty points. The overall computation cost is shown to be less, for the case of Trapezoidal weights only (Liu et al. 2012).

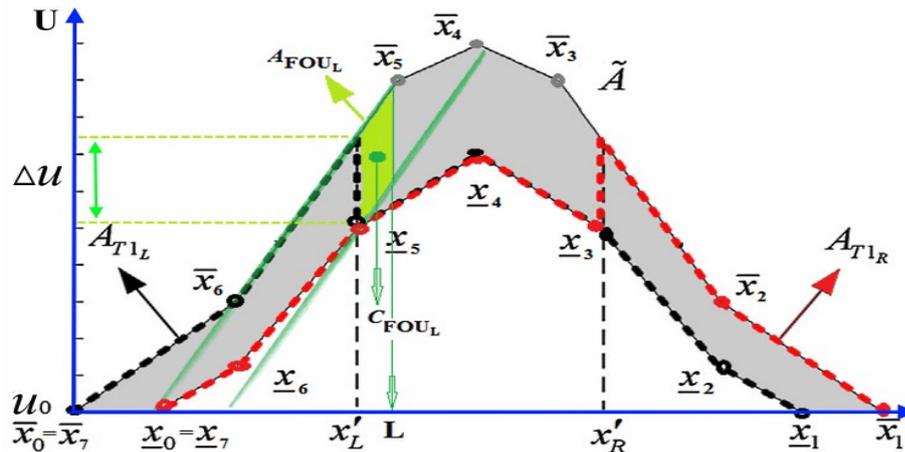


Figure 2-7 : Piecewise Linear IFT2 Set reduction.

2.4.11. Piecewise Linear Interval Fuzzy Type-2 Set Reduction

The piecewise linear (PWL) interval FT2 sets reduction has been proposed using a closed formula, based on the traditional derivation principles to find maximum and

minimum optimal. This has utilised to find the uncertainty boundaries using closed mathematical formula without iterations. A high speed performance and a high accuracy were achieved (Ulu et al. 2013) due to the well-defined geometry of piecewise linear interval sets which can be presented using two polygons, one for the upper and one for the lower membership function. Here, the end points, C_l and C_r , of a type reduced set have been found by calculating centroids of two specially selected type-1 fuzzy sets, A_{T1L} for the left switch point and A_{T1R} for the right switch point, as shown in Figure 2-7 in black and red dots respectively. The sets A_{T1L} and A_{T1R} are described in a non-intersecting closed polygon form using N_L and N_R vertices, for the left and the right respectively. The initial centroids are chosen to be the closest vertices points to the 2.4 and 1.7, which are the EKM initials. This closest is defined by a minimum absolute distance search as: $x_L = \min_{\underline{x}_i} |\underline{x}_i - (\bar{x}_0 + (\frac{\bar{x}_1 - \bar{x}_0}{2.4})|$ and $x_R = \min_{\underline{x}_i} |\underline{x}_i - (\bar{x}_0 + \frac{\bar{x}_1 - \bar{x}_0}{1.7})|$, for the left and the right initialisation points respectively. Here, the symbol \underline{x}_i defines the lower membership vertices, which has been used in the search because the lower membership has usually less vertices. The symbol \bar{x}_i defines the uppers, as shown in Figure 2-7. The \bar{x}_0 and \bar{x}_1 are the leftmost and the rightmost the vertices points of the upper membership function on the horizontal set axes, while for lower set boundaries vertices are \underline{x}_0 and \underline{x}_1 . The area and centroids calculations for A_{T1L} , $C_{A_{T1L}}$ and A_{T1R} , $C_{A_{T1R}}$ have been performed using Surveyor's area formula for a polygon contains N vertices (Ulu et al. 2013) (Braden 1986), as follows:

$$A_{Poly} = \frac{1}{2} \sum_{i=0}^{N-1} (x_i u_{i+1} - x_{i+1} u_i) \quad (2-33)$$

$$C_{Poly} = \frac{1}{6A_{Poly}} \sum_{i=0}^{N-1} (x_i + x_{i+1})(x_i u_{i+1} - x_{i+1} u_i) \quad (2-34)$$

These centroids' initializations are based on choosing the closest vertices to the ideal centroids initials, the (2.4) and (1.7) points. The accurate centroids require a correction parts defined by two movable points L and R, as follows:

$$C_l(\tilde{A}) = \frac{A_{A_{T1L}} \cdot C_{A_{T1L}} + (L^2 - (x_L)^2) \cdot \Delta\mu_{\tilde{A}}(x_L)/2}{A_{A_{T1L}} + (L - x'_L) \cdot \Delta\mu_{\tilde{A}}(x'_L)} \quad (2-35)$$

$$C_r(\tilde{A}) = \frac{A_{A_{T1R}} \cdot C_{A_{T1R}} + (R^2 - (x_R)^2) \cdot \Delta\mu_{\tilde{A}}(x_R)/2}{A_{A_{T1R}} + (R - x_R) \cdot \Delta\mu_{\tilde{A}}(x_R)} \quad (2-36)$$

The (L) point has to minimise the value $C_l(\tilde{A})$, while the (R) point has to maximise $C_r(\tilde{A})$. These positions have been evaluated by deriving with respect to (dL) for the left point and with respect to (dR) for the right point. Then equating the derivations to zero, getting:

$$L = x_L + \frac{-A_{A_{T1L}} + \sqrt{A_{A_{T1L}}^2 + 2\Delta\mu_{\tilde{A}}(x_L) \cdot A_{A_{T1L}} \cdot (C_{A_{T1L}} - x_L)}}{\Delta\mu_{\tilde{A}}(x_L)} \quad (2-37)$$

$$R = x_R + \frac{A_{A_{T1R}} - \sqrt{A_{A_{T1R}}^2 - 2\Delta\mu_{\tilde{A}}(x_R) \cdot A_{A_{T1R}} \cdot (C_{A_{T1R}} - x_R)}}{\Delta\mu_{\tilde{A}}(x_R)} \quad (2-38)$$

The symbol $\Delta\mu_{\tilde{A}}(x)$ represents the vertical uncertainty slice for set element x . This solution is proposed for fixed FOU width during the periods $[x_L, L]$ and $[x_R, R]$ and it can be more complex if the width of the FOU, $\Delta\mu_{\tilde{A}}(x_R)$ is not constant (Ulu et al. 2013). If this is so, many possible cases can exist which depend on the slope of the lower and the upper FOU boundaries and the difference between the initial centroid points x_L, x_R and the approximated centroid L and R. These cases are categorized, as shown in Figure 2-8, to:

- (a) The UMF slope is less than LMF slope, $(|UMF'| - |LMF'|) < 0$, then
 - (i) If $L > x_L$ (denoted L_R) then subtract the red triangle from the calculations to enhance the result.
 - (ii) If $L < x_L$ (denoted L_L) then add the black triangle to the calculations to enhance the result.
 - (iii) If $R > x_R$ (denoted R_R) then add the black triangle to the calculations to enhance the result.
 - (iv) If $R < x_R$ (denoted R_L) then subtract the red triangle from the calculations to enhance the result.
- (b) The UMF slope is greater than LMF slope, $(|UMF'| - |LMF'|) > 0$, then
 - (i) If $L > x_L$ (denoted L_R) then add the black triangle to the calculations to enhance the result.
 - (ii) If $L < x_L$ (denoted L_L) then subtract the red triangle from the calculations to enhance the result.

- (iii) If $R > x_R$ (denoted R_R) then subtract the red triangle from the calculations to enhance the result.
- (iv) If $R < x_R$ (denoted R_L) then add the black triangle to the calculations to enhance the result.

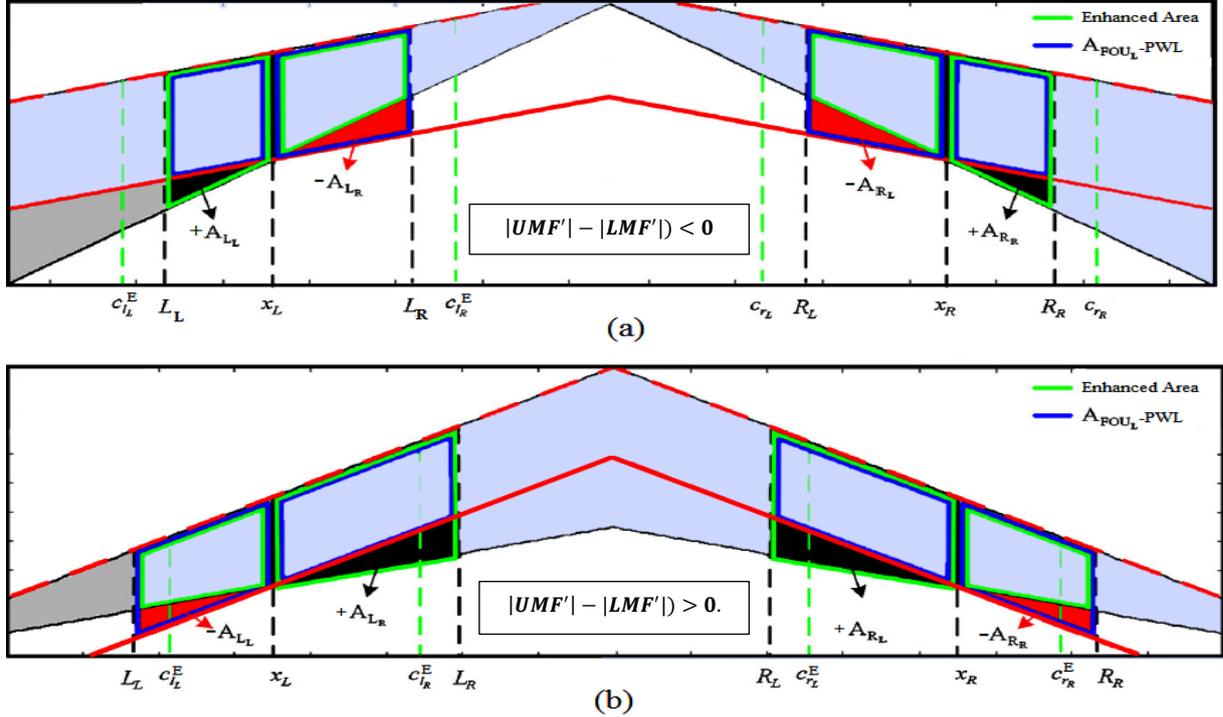


Figure 2-8 : Enhanced PWL type reduction cases. (a) Left and right centroid enhancement for $(|UMF'| - |LMF'|) < 0$. (b) Left and right centroid enhancement when $(|UMF'| - |LMF'|) > 0$.

Here, the computational cost cannot be evaluated in term of the horizontal set discretization points N because the PWL sets are defined using N segments; but the empirical speedup results showed time cut reached to $1/3$ of the EKM execution time.

2.4.12. Linearly Approximated Karnik-Mendel Type reduction

In (Salaken et al. 2015), a linear regression approach is used to approximate the result of the iterative KM type reduction procedure. They followed a tradition approach of systems learning by monitoring a large bunch of input/output data for each fuzzy set, over the most important operating range. Then each fuzzy set will be associated with two

different polynomials, for the left and right uncertainty points. These polynomials have to be tuned such that to reduce any error between the ideal and the approximated points. Each polynomial has a degree of three, in the following form:

$$h(x) = \sum_{i=0}^3 k_i \cdot x^i \quad (2-39)$$

The lower and upper fuzzy firing levels \bar{f} and \underline{f} are used as inputs to generate a mapping, with the assist of the normal KM results, while offline, such that to avoid going into the KM iterations while the controller is on running. These resulting closed form relations will have an execution time very much like the Wu-Tan formula, which has to be pre-tuned, offline, for every fuzzy set.

2.5. Special Cases Type Reduction

Special type reductions have been proposed for special set types. It is useful to cover them here and to get a comprehensive literature review.

A- LR Interval Fuzzy Type-2 Set Reduction

An LR fuzzy set is a normal and convex set having UMF greater than LMF over its universe of discourse have been proposed by (Zimmermann 1991) for fuzzy systems. (Chen et al. 2013). They contain three sections, left, central, and right, that each can be described fully using a closed form equation (Zimmermann 1991) as:

$$LR = \begin{cases} F_L \left(\frac{m_L - x}{\alpha} \right) & \text{if } x < m_L \\ k & \text{if } m_L \leq x \leq m_R \\ F_R \left(\frac{x - m_R}{\beta} \right) & \text{if } x > m_R \end{cases} \quad (2-40)$$

Here, the symbols α and β define the spread of the left and right side, respectively. The constant k is a fixed value represents the current fuzzy firing level. These LR interval sets are important and have been used in many fuzzy systems (Kuo and Chen 1998), (Chen et al. 2013). Reducing such sets has been proposed by (Chen et al. 2013) using a closed form equation to find the reduced set ends. The proposal can be implemented for the cases of linear, Gaussian, and mixed set sides. The proposed reduction uses the integrations of the set sub-sections to compute the uncertainty boundaries, or the KM-switch points, Y_L and Y_R . The integration forms have six possible cases, which depend on the position of the switching points with respect to each other, as shown in Figure 2-9. The switching points position can be changed because of the relative distribution of the set parameters $(\overline{\alpha, \beta, Z_l, Z_r})$ and $(\underline{\alpha, \beta, Z_l, Z_r})$, and the fuzzy firing levels, \overline{f} and \underline{f} , as shown in Figure 2-9. Locating the left and the right uncertainty points are performed iteratively but not like the EKM. Here, the switching points have to be tested on each cycle to determine the equations and the parameters required to be considered in the next iteration. This algorithm processes each consequent set individually because it is impossible to process one final complex consequent set which can be non-convex. Therefore, the final defuzzified value is to be computed using the weighted average aggregation technique based on the individual centroids of the output sets.

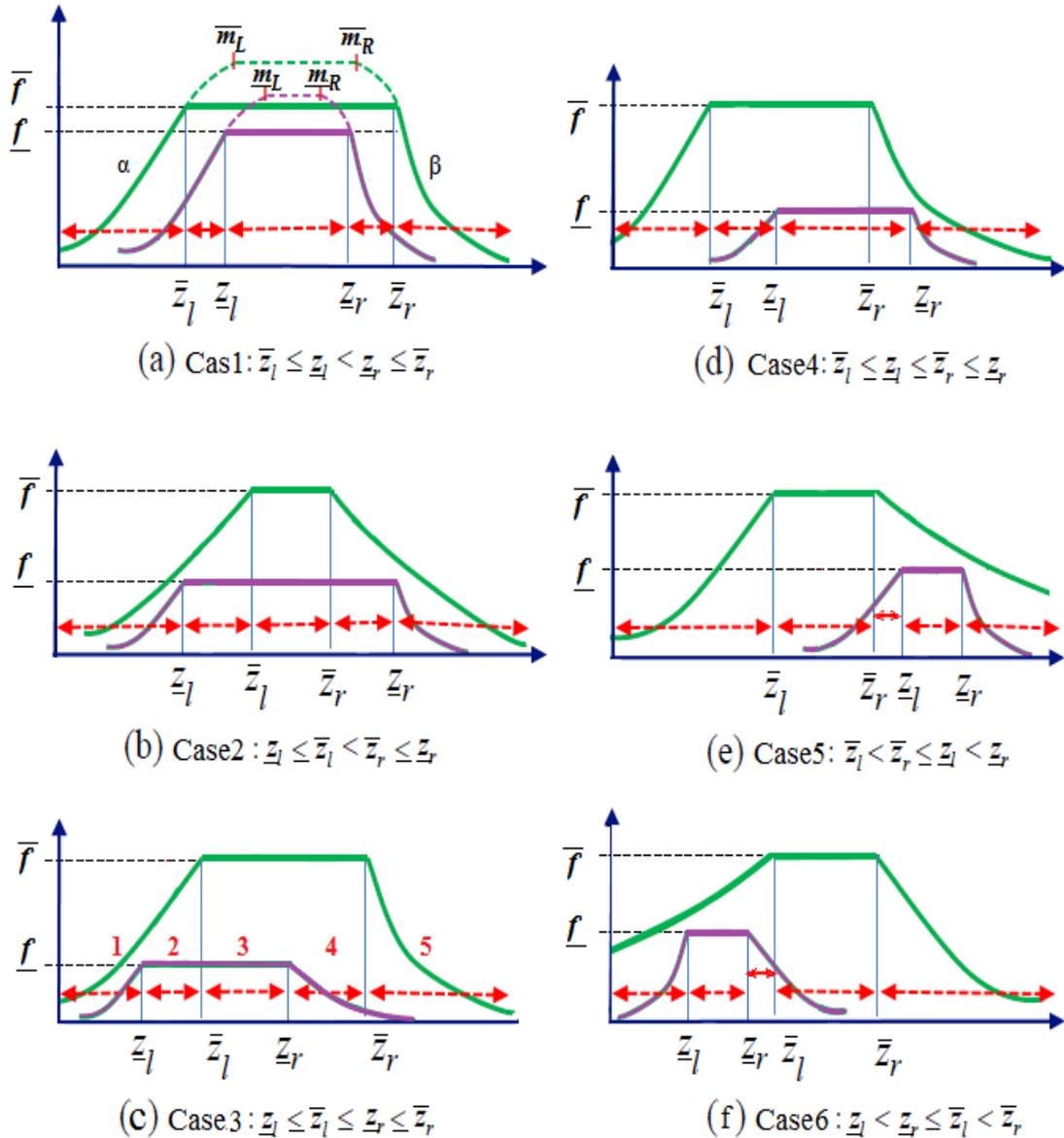


Figure 2-9: LR Interval FT2 sets possible cases and the possible switching point locations (1, 2, 3, 4, 5).

B- Reducing Interval FT2 Sets with Spikes

In some modelling situations and with some aggregators, an interval FT2 set with spikes may exist (Aisbett and Rickard 2014). Such spikes can create very big error if any of them falls exactly at a discretisation step position (Aisbett and Rickard 2014). The correct type reduction, in this unique situation, cannot be satisfied easily using normal type reductions techniques. One solution has been proposed in (Aisbett and Rickard 2014) that is based on processing, as a first step, a smoothed version of such interval FT2

set, totally removing any spike of the set. This smoothed set type reduction, is to be performed using a normal EKM algorithm. The second step is to identify the peak points, μ_{MaxL} and μ_{MaxR} , which are the highest points of the FT1 sets that being used in the EKM procedure. The effective spikes, in each FT1 set, are only those whom attached to upper membership sections and having values higher than the peak of the smoothed set version. Thus, spikes that fall to the left of Y_L or to the right of Y_R , and have enough height are the only ones to be taken into account, as shown in Figure 2-10. The new centroid of the FT1 set, which is going to contain the effective spikes, is calculated as below. The membership symbol with apostrophe, $\mu'(y_i)$, describes the smoothed set version.

$$C_L = \frac{\sum_{i=1}^{L-1} \overline{\mu'}(y_i) \cdot y_i + \sum_{i=L}^N \underline{\mu'}(y_i) \cdot y_i + \sum_{s \leq L-1} (\overline{\mu}(y_s) - \mu_{MaxL}) \cdot y_s}{\sum_{i=1}^{L-1} \overline{\mu'}(y_i) + \sum_{i=L}^N \underline{\mu'}(y_i) + \sum_{s \leq L-1} (\overline{\mu}(y_s) - \mu_{MaxL})} \quad (2-41)$$

$$C_R = \frac{\sum_{i=1}^{R-1} \underline{\mu'}(y_i) \cdot y_i + \sum_{i=R}^N \overline{\mu'}(y_i) \cdot y_i + \sum_{s \geq R} (\overline{\mu}(y_s) - \mu_{MaxR}) \cdot y_s}{\sum_{i=1}^{R-1} \underline{\mu'}(y_i) + \sum_{i=R}^N \overline{\mu'}(y_i) + \sum_{s \geq R} (\overline{\mu}(y_s) - \mu_{MaxR})} \quad (2-42)$$

The small spikes have been proved to have no effect on the centroid result, thus they have to be neglected (Aisbett and Rickard 2014). The process computation cost is very similar to the EKM cost, but that can be affected by the number of the existed spikes.

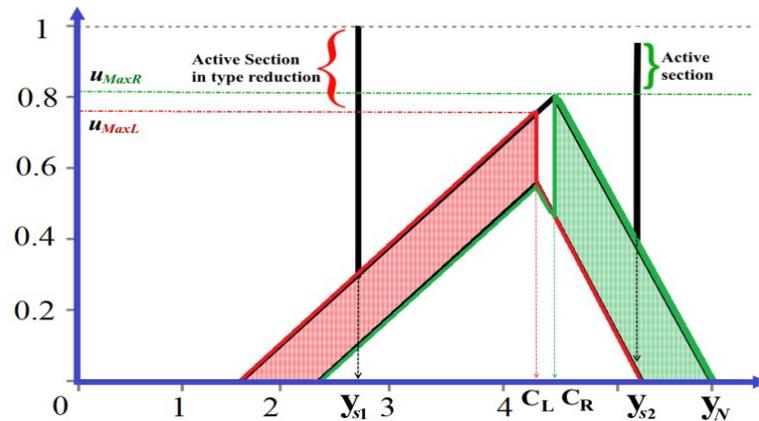


Figure 2-10: Type reduction principles of an interval FT2 set with spikes.

2.6. Reduction of The General Fuzzy Type-2 Sets

The general fuzzy type-2 (GFT2) systems are capable of better performance because they can model uncertainty in a better way. However, the type reduction of the GFT2 sets represents a barrier in real time systems, like in autonomous vehicles and robotic systems (Wagner and Hagnas 2010). The GFT2 sets reduction techniques complexity arises from its complex representation. Many efforts to simplify the set representation have been proposed through the use of α -cuts and z-Slicing techniques, where the general FT2 set can be decomposed to form simpler sub-sets based on sub features. Decomposition techniques have some related definitions; these have to be presented first before going to use them for GFT2 sets reduction.

A- Vertical Slicing: Each set domain element (x) is associated with a vertical plane formed of the primary and the secondary membership functions of the element. This vertical plane is usually defined by the fuzzy axis U and \tilde{U} .

B- The Alpha-Cut (α -cut): Is an important concept, initially proposed by Zadeh (1971) under the name of *resolution identity* (Hamrawi 2011) to decompose the fuzzy set based on its features to simplify its processing. The α -cuts of a FT1 set (A) are crisp, classical sets (Hamrawi 2011), and each represents specific set's feature, as: $A_\alpha = \{x \in X \mid \mu_A(x) \geq \alpha, \alpha \in [0,1]\}$. Each alpha set A_α has an indicator set (George J. Klir and Bo Yuan 1995), which is defined as following:

$$I_{A_\alpha}(x) = \begin{cases} 1 & \text{if } x \in A_\alpha \\ 0 & \text{if } x \notin A_\alpha \end{cases}.$$

A detailed set definition using α -cut is:

$$A_\alpha = \alpha \cdot I_{A_\alpha}(x) = \{(x, \alpha) \mid \forall x \in A_\alpha\}.$$

C- The α -Cut Representation Theorem: Any fuzzy set can be represented through the union of its α -cut sets as: $A(x) = \bigcup_{\forall \alpha} \alpha \cdot I_{A_\alpha}(x)$, where each of them is an embedded constrained set.

D- The α -cut Properties: For a convex FT1 sets A and B the following are true (George J. Klir and Bo Yuan 1995):

- 1- $A_{\alpha^+} \subseteq A_\alpha, \forall \alpha.$
- 2- If $A \subseteq B$ then $A_\alpha \subseteq B_\alpha, \forall \alpha.$
- 3- If $A = B$ then $A_\alpha = B_\alpha, \forall \alpha.$

- 4- $(A \cup B)_\alpha = A_\alpha \cup B_\alpha$.
- 5- $(A \cap B)_\alpha = A_\alpha \cap B_\alpha$.
- 6- $A'_\alpha = (A_{(1-\alpha)^+})'$, where the apostrophe indicates a complement of the fuzzy set.

E- The α -Plane, Horizontal Slices: Applying the α -cut on a fuzzy type-2 set generates a horizontal slice that falls in the XU plane. Each α -plane contains the set elements which have a secondary grade equal or greater than α . Mathematically, this can be described for each α -plane as below:

$$\widetilde{A}_\alpha = \{x, \mu_{\widetilde{A}}(x) \mid \widetilde{u}_{\widetilde{A}}(x) \geq \alpha, \forall x \in X, \forall \mu_{\widetilde{A}}(x) \in J_x\}.$$

The union of these planes describes the initial FT2 set (Mendel et al. 2009).

F- Fuzzy Type-2 α -Identifier set: This describes an interval fuzzy type-2 set associated with each α -plane (George J. Klir and Bo Yuan 1995), (Mendel et al. 2009) has a secondary membership value equal to 1 for every element in it, as follows:

$$I_{\widetilde{A}_\alpha}(x) = \begin{cases} 1 & \text{if } (x, \mu_{\widetilde{A}}(x)) \in \widetilde{A}_\alpha \\ 0 & \text{if } (x, \mu_{\widetilde{A}}(x)) \notin \widetilde{A}_\alpha \end{cases}.$$

G- The z-Slice: This is a representation technique, very similar to alpha planes, formed by horizontally slicing the z-dimension, of the general fuzzy type-2 sets, to equally spaced grades each falling in the range of $[0, 1]$. The Z-slice creates an interval FT2 set with secondary grade equal to $\frac{i}{z_i}$, $1 \leq i \leq I$. Each generated interval FT2 sets, has specific interval domain, J_{z_i} , defined by its two end points $\overline{J_{z_i}}$ and $\underline{J_{z_i}}$. Mathematically, the z_i -slice, of a general FT2 set (\widetilde{A}) which has been sliced to (I) slices, can be described as in (Wagner and Hagraas 2010) by: $\widetilde{A}_{z_i}(x) = \sum_{x \in X} \sum_{u_i \in [\underline{J_{z_i}}, \overline{J_{z_i}}]}(x, u_i, z_i)$. The whole set is formed by a union of the individual slices as: $\widetilde{A}_z(x) = \sum_{i=1}^I \widetilde{A}_{z_i}(x)$. Note that, the z_0 slice is not considered because it has a secondary membership value equal to zero which does not affect the system decision.

2.6.1. General Fuzzy Sets Reduction Using Vertical Slice

Every set element x_i in a general fuzzy type-2 set, has a vertical slice (VS) formed by its primary and secondary membership functions. This defines a plane projected on the $U_{\tilde{A}}$ and $\tilde{U}_{\tilde{A}}$ axis, which usually known as J_x in interval FT2 sets. Each plane describes the uncertainty associated with its set element; thus, reducing the uncertainties that are associated with all of the set elements to get crisp values, can create one FT1 set. This idea is being adopted in (Lucas et al. 2007) to reduce the general FT2 sets by reducing all the FT1 sets which shapes the uncertainty in the VS_i planes, as shown in Figure 2-11.

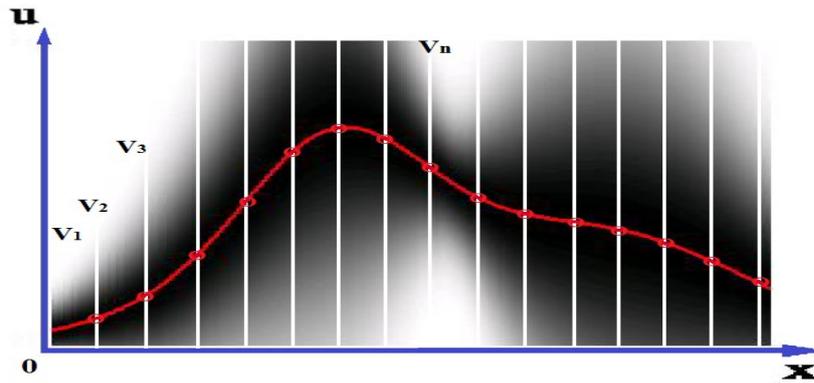


Figure 2-11: Vertical Slice Type Reduction of a General Type-2 Set.

Here, the dark parts represent high membership value and the light parts represent low values. This type reduction is very much like the N-T procedure to reduce the interval sets, but applied here to general FT2 sets. This vertical slicing type reduction (VSTR) omits the uncertainty measurements of the fuzzy type-2 set (Lucas et al. 2007), just like the N-T technique (Nie and Tan 2008). The computation cost for the VSTR process is $(N \times 2M_n)$ additions and $(N \times M_n)$ multiplications, when applied to discrete set that has N elements, each element is associated with a vertical slice discretised to M_n elements.

2.6.2. The α -Planes Type Reduction Method

Based on the elemental logic of the α -cuts on fuzzy sets, like $(A \cup B)_\alpha = A_\alpha \cup B_\alpha$ and $(A \cap B)_\alpha = A_\alpha \cap B_\alpha$, one can easily recognize that to reduce computational redundancy, the fuzzy operations can only be performed on those similar uncertainty levels (George J. Klir and Bo Yuan 1995), (Hamrawi 2011). This has been utilized in (LIU 2008), (Mendel et al. 2009) to decompose general FT2 set to embedded sets, as shown in Figure 2-12, then processing only those embedded sets, which have similar features, to achieve efficient type reduction. The reduced FT1 set which results from this general FT2 set reduction is formed by a union of the individual FT1 reduced sets, which have been resulted from the alpha planes sets reduction, as: $C_0 = \sum_{\alpha \in [0,1]} (\alpha, [C_{\alpha L}, C_{\alpha R}])$. The notation $(\alpha, [C_{\alpha L}, C_{\alpha R}])$ denotes an interval FT1 set with domain $[C_{\alpha L}, C_{\alpha R}]$ and membership equal to α . The symbol \sum describes a discrete set union, which is to be performed using the maximum operation.

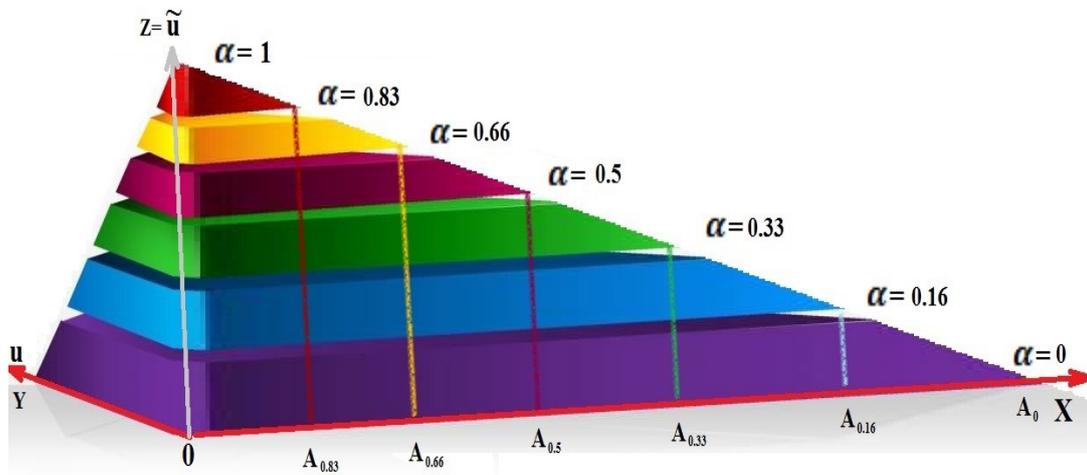


Figure 2-12: General fuzzy type-2 Set decomposed by alpha planes at the values 0, 0.16, 0.33, 0.5, 0.66, 0.83 and 1.

The final crisp value is computed by defuzzifying the envelope which contains all the FT1 sets, C_0 , (Wagner and Hagsras 2010) using weighted average.

In the alpha plane type reduction, more computations that are redundant have been excluded in the case of *fully symmetrical fuzzy type-2 sets*, which have symmetrical

secondary primary membership functions around some point, $x = m$. This reduction is achieved by considering point $y_0 = m$ as the defuzzified value if the system does not require uncertainty details (Mendel et al. 2009). For convex general FT2 sets case, which have normal secondary membership function, it is possible to calculate the centroid and the defuzzified value by only processing α -planes at values 0 and $\tilde{\mu}_{Max}$ and still achieving “excellent approximation” (Mendel et al. 2009). Also, it had been shown that, for higher accuracy and complex uncertainty figures, the use of less than (10) alpha levels is sufficiently acceptable for control applications (Mendel et al. 2009). However, this accuracy cause still has no mathematical proof or illustration (Mendel et al. 2009).

The computational cost of the alpha plane type reduction, if decomposition is made using ($k \leq 10$) levels, is k times the cost of reducing one interval FT2 set. The good point of this type reduction is the possibility of using independent $2k$ parallel processors to perform the centroid computations.

2.6.3. Random Sampling Method

The random sampling type reduction-defuzzification (RSTR-DF) method has been proposed by (Greenfield et al. 2005). The method gets use of the high redundancy nature of the embedded sets. Any FT2 set of $1/N$ horizontal discretization level and $1/M_n$ secondary discretization level will contain $\prod_{n=i}^N M_n$ embedded sets. This number can be very large even for small digitization levels. As for example if $N=10$ and $M=10$, then embedded sets count is $(10)^{10}$. This high population has been replaced by random small samples that can be used to approximate the type reduced set (Greenfield 2012). In this situation, the Central Limit Theorem (Greenfield et al. 2011), which states that for any huge populations, there is a mean (m) and a standard deviation σ , that can be estimated using a sample of size N elements, which will have the same mean (m) of the bigger population, and a variance of $\frac{\sigma^2}{N}$. This mean value can represent the centroid of the defuzzified set (Greenfield et al. 2011). Thus, instead of processing huge embedded sets, a small set sample can be used to generate an accurate centroid too. The process contains the following main steps: (i) For every domain division (x_i), select a random

primary membership value in the vertical slice (J_{xi}) and construct one embedded set; (ii) Calculate the centroid of this randomly selected embedded set and associate it with minimum secondary membership; (iii) Construct the type reduced set using these generated tuples. The accuracy of this type reduction, according to practical experiments performed by (Greenfield et al. 2011) using common fuzzy sets like Gaussian and trapezoidal, shows a generate accuracy, even for low discretization level, as summarised in the bar chart Figure 2-13.

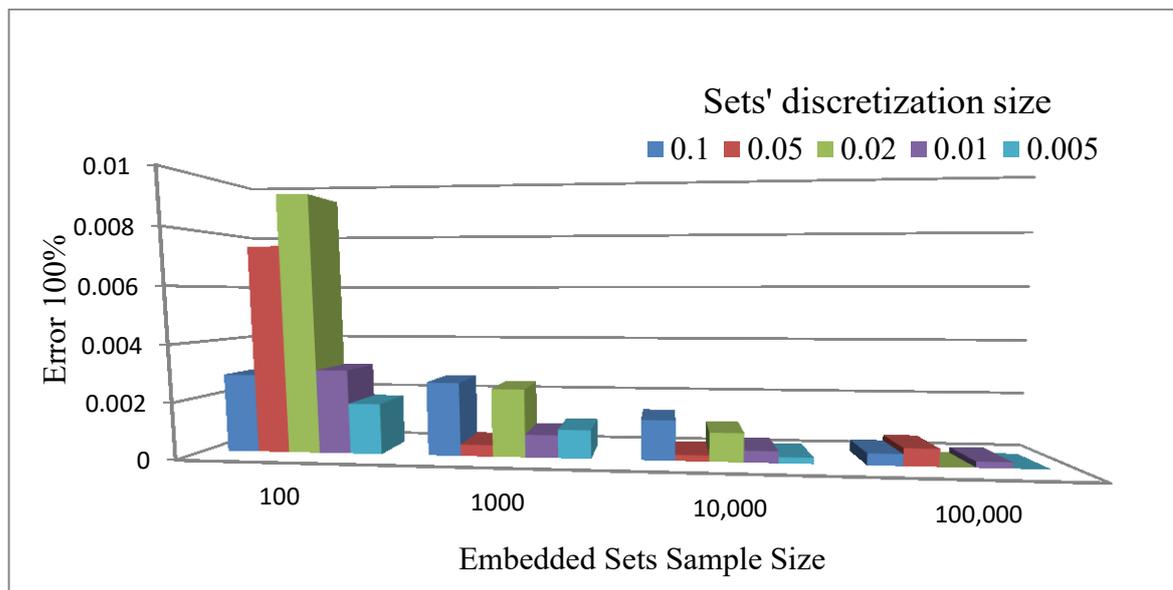


Figure 2-13 : Random sampling, type reduction-defuzzification, accuracy.

The computation cost is equal to (Sample size/discretization level) of multiplications, plus twice of that as additions.

The random sampling type reduction process has been developed further by (Linda et al. 2010), by defining a sampling importance factor (SIF) using the bounded Gaussian probability density function (PDF) to limit the embedded sets sampling spread. Here, the Gaussian PDF mean is calculated simply by averaging the FOU limits. The standard deviation has been modified to an adjustable form by subtracting the upper and lower boundaries of the FOU and dividing over 2^β , as: $\sigma = (\overline{FOU} - \underline{FOU}) / 2^\beta$. Thus, (β) can control the Gaussians PDF-spread to important regions only, as shown in Figure 2-14. Through this development, more calculation reduction has achieved with smoother control surface (Linda et al. 2010). However, the uncertainty boundary becomes more undefined.

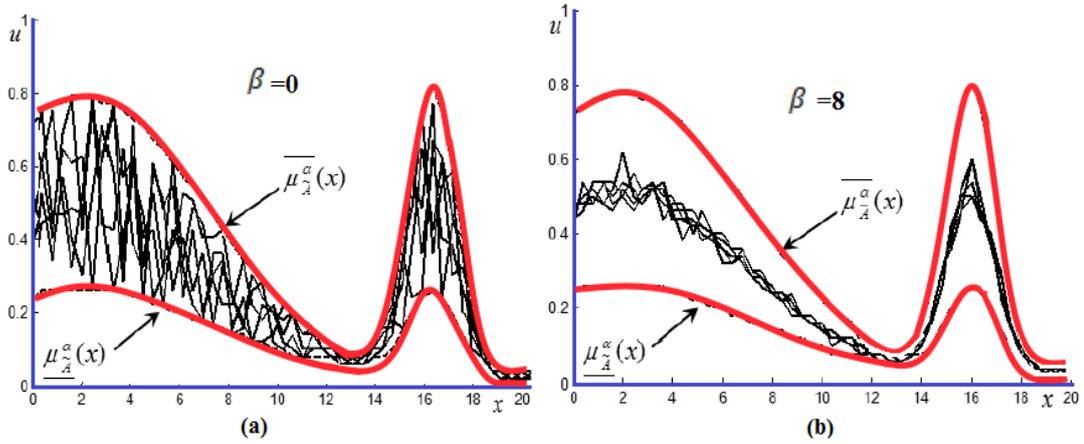


Figure 2-14: Importance Random Sampling type reduction principles. (a) Using importance control factor $\beta = 0$. (b) Using importance control factor $\beta = 8$.

2.6.4. The Enhanced α -Plane Type Reduction ($E\alpha$ -Plane)

More efforts to enhance the type reduction of the general FT2 set using α -planes have been made by (Yeh et al. 2011), (Wu and Nie 2011), and (Linda and Manic 2012), were based on enhancing the initial start points of the EKM TRP on each α -plane. The first α -plane, which falls at α_{Max} , has been suggested to use an initial point, like the EKM initialization, defined as the principal set mean point. The following α -planes, where $k = \{1, 2, \dots, K\}$, $\alpha_k > \alpha_{k+1} > 0$, have proposed to use its predecessor plane uncertainty ends, as the initialization to the current α -plane's EKM calculations.

2.6.5. Quasi-Fuzzy Type-2 Sets Reduction

In (Mendel and Liu 2008), it has been shown that some special general fuzzy type-2 sets, which have triangular or trapezoidal secondary membership functions, named Quasi-Fuzzy Type-2 sets, can be reduced through decomposing them using just two α -planes, at $\alpha=0$ and $\alpha=1$. In the case of triangular secondary membership function, the resulted set at $\alpha=1$ will be a type-1 fuzzy set. The resulting type reduced set is defined only by three points $C_{L,\alpha=0}$, $C_{R,\alpha=0}$, $C_{o,\alpha=1}$ which form a triangular FT1 set. This set defuzzification generated the final crisp output. In the case of trapezoidal

secondary membership function, a trapezoidal (like) type reduced set will be generated which can be defuzzified very easily using its vertices points. The type reduction, for each α -plane, can be performed using the approximate Max-Min closed uncertainty formula (Mendel 2002) to achieve a high speed fuzzy system that suits real time applications (Mendel and Liu 2008).

2.6.6. Geometric Type-2 Reduction-Defuzzification

For a general FT2 set defined using straight segments, it is possible to describe its outer surface using primitive triangular shapes, and then calculate their centroid, like any geometric body. That is what has been utilised by (Coupland 2007), where the triangles have been used to cover the surface of the sets and calculates the final set centroid using a weighted average, of the individual centroids, as:

$$C_0 = \frac{\sum_{t=1}^T C_t \cdot A_t}{\sum_{t=1}^T A_t} \quad (2-43)$$

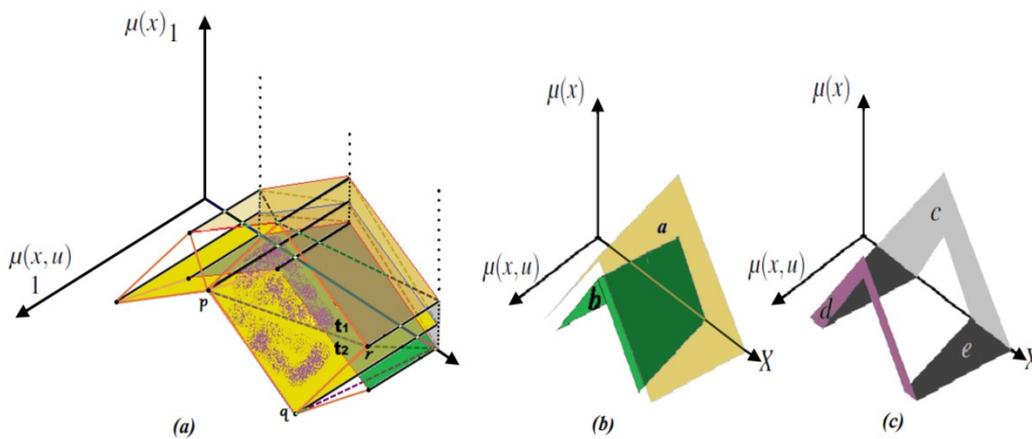


Figure 2-15: Geometric type reduction-defuzzification. (a) Two triangles shape the secondary membership function in a general FT2 set. (b-c) The possible sides to be described using triangles are a, b, c, d and e.

Where C_t and A_t are the centroid and the area respectively, for each individual triangle (t), of a total (T) triangles. The set five sides (a, b, c, d, and f) shown in Figure 2-15 have to be covered by triangles. Every triangle is to be formed using successive fuzzy elements and one of their membership functions. Any triangle area is to be evaluated as half the determinant of two of its edges' cross products. The triangles' centroids are to be evaluated by averaging the horizontal coordination of every triangle vertex, as for example the points p - q - r shown in Figure 2-15:(a).

$$C_t = \frac{x_p \cdot x_q \cdot x_r}{3} \quad \text{and each triangle area is } A_t = \frac{|V_q - V_p \times V_r - V_p|}{2} \quad (2-44)$$

The computational cost for every discretization step, which contains two triangles, is: two centroid calculations, which use 4 multiplications; two area calculations, which require 12 multiplications; all that have to be performed for the set's 5 sides. So, the total computational cost is in the range of $(16 \times 5 \times 2 \times N)$ multiplications which is still high if compared to the EKM calculation cost.

2.7. Interval Set TRPs Comparison

Some type reductions are suitable for general fuzzy sets, while others are designed for special set types. To acquire a clearer view about what has been done, a comparison is made, shown in Table-1, based on the type reduction nature, accuracy, best-fit set, and computational cost, of the interval and the general fuzzy sets. In addition, Figure 2-16 shows a comparison between the costs of the interval sets reduction. This computational cost has been estimated by assuming a fuzzy system uses 10 consequence sets, each is discretised to 100 samples. The addition cost is considered one clock cycle and the multiplication and the addition cost are considered as three clocks. These costs are typical for Intel-Pentium processors (Fog 2004), which it is one of the most common and general processors

Table- 1: Comparison of the interval fuzzy type-2 sets reduction techniques

Features Procedure Name	Set type	Accuracy	Process cost	Initiali- za-tion	Uncer- tainty	Closed formu- la	Maxim- um iteratio- ns	Search techni- que	Parall- el process	Year of appearance & Reference
Exhaustive TRP	Any	Very High, considered as a reference process	$(N \cdot mul + 2N \cdot add), \prod_{n=1}^N M_n$	1	Yes	Yes	$\approx M^N$	Sequential search	Yes	(Karnik and Mendel 2001)
IKM TRP	Any	Very High	$2 \cdot (N \cdot mul + 2N \cdot add), N/12$	$\frac{N}{2}$	Yes	No	$2 + \frac{N}{12}$	Iterative search	No	(Karnik and Mendel 2001)
Approximate KM TRP	Uniform width set	Low	$2N \cdot mul + 4N \cdot add$	-	Yes	Yes	1	No	No	(Karnik and Mendel 2001)
Max-Min bounds	Any	High	$5N \cdot mul + 10N \cdot add$	-	Yes	Yes	1	No	No	(Mendel 2002)
Wu-Tan approximation	Trapezoidal	Good	$(N \cdot mul + 2N \cdot add) + (2 \cdot Sets \cdot add + 2 \cdot Sets \cdot mul)$	-	No	Yes	1	No	Yes	(Wu and Tan 2005)

Table- 1 - (Continued): Comparison of the interval fuzzy type-2 sets reduction techniques

Features	Set type	Accuracy	Process cost	Initialization	Uncertainty	Closed formula	Maximum iterations	Search technique	Parallel process	Year of appearance & Reference
Procedure Name										
EKM TRP	Any	High	$2(N \cdot mul + 2N \cdot add) + \frac{N}{12}(mul + 3 \cdot add)$	$\frac{N}{2.4}$ $\frac{N}{1.7}$	Yes	No	$2 + \frac{N}{12}$	Iterative and search	No	(Wu and Mendel 2007)
EKM-Incremental Formula	Any	High	$3N \cdot mul + 9N \cdot add$	1	Yes	No	2	Search	No	(Duran et al. 2008)
N-T Method	Any	Low	$2N \cdot mul + 3N \cdot add$	-	No	Yes	1	No	No	(Nie and Tan 2008)
Collapse TRP	Any	Low	$(4N \cdot mul + 6N \cdot add)$	-	No	No	2	Sequential	No	(Greenfield, Chiclana, Coupland, et al. 2009).
Type-1 OWA TRP	Any	Very High	$2 \cdot (N \cdot mul + 2N \cdot add) \cdot N/10$	1	Yes	No	$2 + \frac{N}{10}$	Iterative search	No	(Chiclana and Zhou 2011).
Trapezoidal - WEKM TRP	Any	High	$2(N \cdot mul + 2N \cdot add) + \frac{N}{14}(mul + 3 \cdot add)$	$\frac{N}{2.4}$ $\frac{N}{1.7}$	Yes	No	$2 + \frac{N}{14}$	Iterative and search	No	(Liu et al. 2012)

Table- 1- (Continued): Comparison of the interval fuzzy type-2 sets reduction techniques

Features	Set type	Accuracy	Process cost	Initializa- tion	Un- certa- inty	Closed formu- la	Maxim- um iteratio- ns	Search techni- que	Parall- el process	Year of appearance & Reference
Procedure Name										
LR IFT2 set Reduction	LR with linear and Gaussian segments.	High	$\approx 1KM/10$	$\frac{N}{2}$	Yes	Mix	10	Closed and Iterations	No	(Chen et al. 2013)
PWL	Trapezoidal	High	$\frac{1}{3} \cdot EKM$ $\approx 1.25 \times EKM$	-	Yes	Yes	2	No	No	(Ulu et al. 2013)
IFT2S+Spikes Reduction	Any	Good	$\approx 1.25 \times EKM$	$\frac{N}{2.4}$ $\frac{N}{1.7}$	Yes	Mix	2N+1	Iterative and search	No	(Aisbett and Rickard 2014).

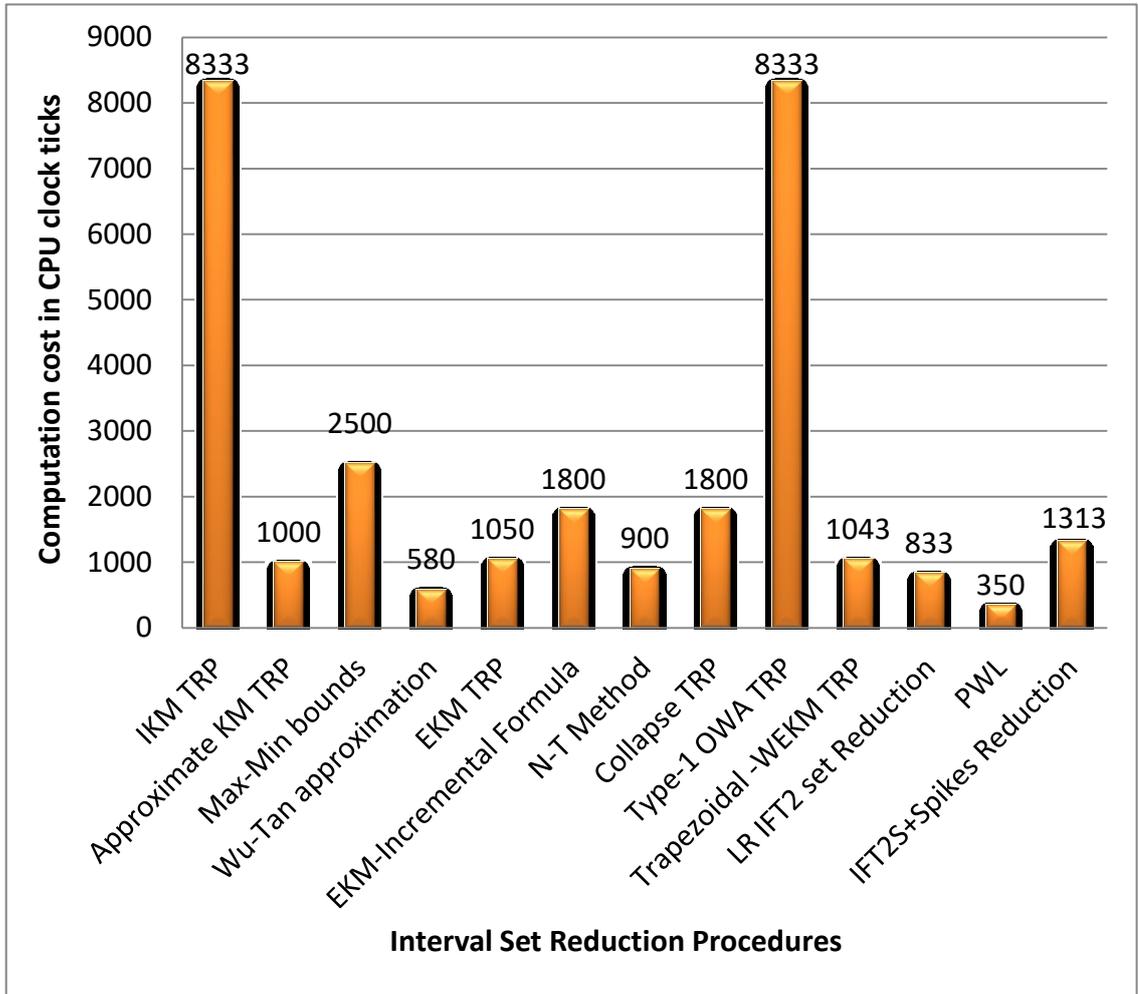


Figure 2-16: Interval FT2 sets type reduction procedures cost comparison.

2.8. General Set Reductions Comparison

The general fuzzy type-2 sets reduction process consumes a very high level of computation to eliminate the high uncertainty of these sets. The alpha cut decomposition and the Z-slicing have reduced the type reduction cost by processing the symmetrical uncertainty levels only. Using the quasi general FT2 sets (Mendel et al. 2009) is another approach offering low type reduction cost and good uncertainty modelling, in comparison to interval FT2 sets. The tabulated information, shown in Table-2, summarizes the nature of the general FT2 sets type reduction and their computational cost and the targeted fuzzy sets. This computational cost has been estimated by assuming a fuzzy system uses 10 consequence sets, each is discretised to 100 samples using the typical Intel-Pentium processors (Fog 2004) computation cost. The bar graph, in Figure 2-17, compares the computational cost of the general fuzzy type-2 reduction, taking in account their historical development. It can be seen that the best compromise between computational cost and uncertainty presentation is in using the quasi-fuzzy sets.

Table- 2: Comparison of the general fuzzy type-2 sets reduction techniques.

Features	Best set type	Accuracy	Process cost	Initialization	Uncertainty indication	Closed formula	Maximum iterations	Search technique	Parallel process	Year of appearance & Reference
Procedure Name	Any	Good	$(N.mul+2N. add).100$	1	Yes	No	200	No	Yes	(Greenfield et al. 2005)
	Trapezoidal	High	$\approx 150 \times N$	1	No	Yes	1	No	Yes	(Coupland 2007)
Geometric TRP	Any	High	$\approx 11 \cdot EKM$	$\frac{N}{2.4}$ $\frac{N}{1.7}$ for all planes	Yes	No	22N	Iterative and search	Yes	(LIU 2008)
Quasi FT2 sets TRP	Quasi Sets	Good	$\approx 1.5 \times EKM$	$\frac{N}{2.4}$ $\frac{N}{1.7}$	Yes	No	3	Iterative and search	Yes	(Mendel and Liu 2008)
Importance Random TRP	Any	Good	$(N.mul+2N. add).50$	1	Yes	No	100	No	Yes	(Linda et al. 2010)
Ea-Planes TRP	Any	High	$\approx 10 \cdot EKM$	$\frac{N}{2}$	Yes	No	20	Iterative and search	Yes	(Yeh et al. 2011)

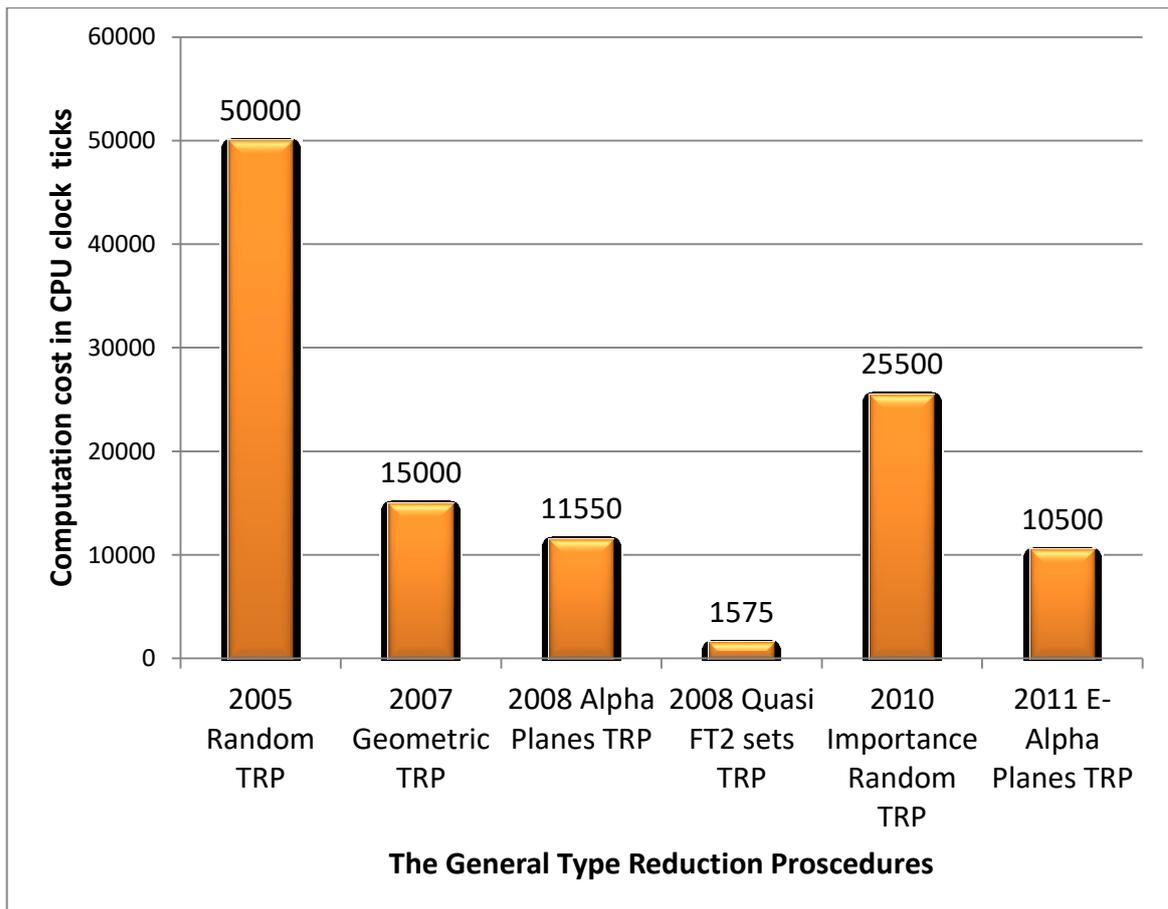


Figure 2-17: Computational cost comparison of the general FT2 set reductions.

2.9. Conclusions

Concerning the current type reduction methods, it can be seen that either they are slow, general, and accurate, or; they are fast but sacrifice accuracy, uncertainty, and generality. Each specific technique has its own features, which can't be changed. However, it is very useful to build a type reduction procedure or suggest a method that can slide between high accuracy and complexity to low accuracy and complexity. The features sliding property will offer a selective accuracy and computation complexity control, which will help to do, only, the necessary computations that fulfil the fuzzy system requirements at a specific situation and time.

In addition to that, because of the high usage of integrations in the accurate type reductions, there is a high chance to exclude redundant computations that are distributed inside such procedures by using adaptive integration methods, which support selective accuracy. Locating and eliminating those computations can result a shorter execution time without affecting the accuracy of the initial algorithms. Also, the computation costs of the type reductions is seen to be very much dependent on the initialisations of the type reduction procedures, like the EKM and IKM algorithms. Thus, there is a good possibility to enhance the type reduction cost by finding better initializations.

Chapter 3: Autonomous Vehicle Fuzzy Type-2 Controllers

Using high order fuzzy sets in digital autonomous vehicle controllers facilitates higher uncertainty modelling and better utilization of human expertise. This is important because autonomous vehicles usually face high uncertainty levels, which arise from: environment noise; possible inaccurate sensor readings; deterioration of the actuator's performance; and the variation in humans knowledge and experience (Mendel 2000).

In this chapter, the two main fuzzy type-2 system architectures that are commonly used in autonomous vehicle controllers will be briefed to have a better understanding and clear conclusions while surveying the autonomous vehicle controllers, conducted in the following sections.

3.1. The Fuzzy Type-2 Mamdani Controller

If uncertainty and non-linearity dominate a system, then most of the control techniques will either fail to perform adequately or will require high design effort to achieve the requirement, while fuzzy type-2 controllers can be constructed easily and achieve satisfactory even for model-less systems (Castillo 2011). Also, most of the traditional controller techniques applied for linear and nonlinear systems can be applied using fuzzy techniques achieving better uncertainty process. The two main fuzzy type-2 logic controller structures are the Mamdani (Mamdani 1977) and Takagi-Sugeno-Kang (Takagi and Sugeno 1985) architectures. The Mamdani fuzzy type-2 controller uses linguistic variables and commands to describe system inputs, actions, and outputs in a

method very similar to human decision-making process, thus they encouraged wide usage.

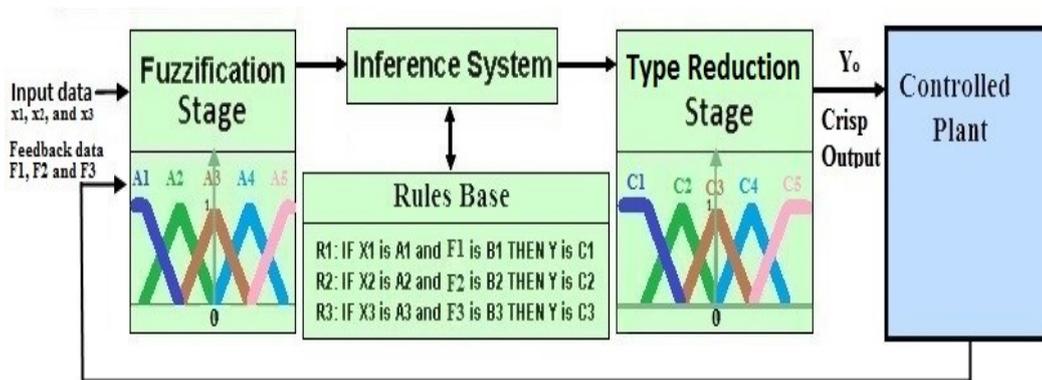


Figure 3-1: Main units of the fuzzy type-2 controllers.

The main units of the Mamdani fuzzy type-2 controller, shown in Figure 3-1, are:

- The fuzzification stage is a mapping from the input space to the fuzzy unity space $\mathbb{I} = U: [0,1]^1$. This mapping is controlled by the membership functions $u_{A_n}: X \rightarrow [0,1]$ of the input set (Thiele 1994). In case of FT2 systems, each input variable can generate multiple fuzzy levels according to the uncertain footprint of the input set.
- The inference stage is an operation of generating one decision according to the input fuzzy states, using (if-then) relations. Different input states vectors can be defined, and each can have different size K_i . Every state has to be processed using one rule to generate one decision represented by fuzzy level, which describes the decision importance (Wang and Wang 2001). Thus, every rule defines a mapping from K_i dimensional space to one-dimensional space, as: $[0,1]^k \rightarrow \mathbb{I}$. Putting different rules altogether would create one rule-base to shape the fuzzy controller performance. The deduction of one rule, which is based on a specific state, has to be extended to wider input range, as shown below:

Given Rule 1: *IF antecedent_1 is $A_1(x)$*
And *IF antecedent_2 is $A_2(x)$*
And *IF antecedent_N is $A_N(x)$ THEN consequent is $C_1(y)$*

Deduction 1: *IF antecedent_1 is $A'_1(x)$*
And *IF antecedent_2 is $A_2(x)$*
And *IF antecedent_N is $A_N(x)$ THEN consequent is $C'_1(y)$*

Here, inferencing of $C'_1(y)$ is performed using $A'_1(x)$, which is a slightly modified version of the main state set $A_1(x)$. This is performed using Zadeh's *implication process* $Z(x \rightarrow y)$ as following.

$$\begin{aligned} \text{Given,} \quad & Z(u_A(x) \rightarrow u_C(y)) \\ &= \begin{cases} \overline{u_A(x)} \vee u_C(y) & \text{if } u_A(x) \leq u_C(y) \\ u_A(x) \vee \overline{u_C(y)} & \text{if } u_A(x) > u_C(y) \end{cases} \end{aligned} \quad (3-1)$$

$$\text{Deduce } C'(y) = \sup_{x \in X} [u_{A'}(x) * Z(u_{A_n}(x) \rightarrow u_{C_n}(y))] \quad (3-2)$$

This Zadeh implication process has been shown to be not suitable for engineering control applications (Mendel 1995) because of its high non-linearity. Therefore, it has been replaced by the Mamdani implication formula using (minimum) operation only (Mamdani 1977).

$$M(u_A(x) \rightarrow u_C(y)) \triangleq \text{Min}(u_A(x), u_C(y)) \quad (3-3)$$

$$\equiv M(u_A(x) \rightarrow u_C(y)) \triangleq u_A(x) * u_C(y) = u_{A \rightarrow C}(x, y) \quad (3-4)$$

This deduction formula has been generalized by Larsen (Mendel 1995) using t-norm operations as in equation (3-4). Generally, aggregating minor aspects in different consequents is performed using supremom and t-norm implications, which are abbreviated as one (SUP-STAR) operation, to act as follows:

$$C'(y) = \sup_{x \in A'} [u_{A'}(x) * u_{A \rightarrow C}(x, y)] \quad (3-5)$$

However, to get correct deductions from a discretised fuzzy controllers, the operations *minimum* and *maximum* are used, respectively, to perform the t-norm and supremom operations (Karnik et al. 1999). If more than one rule is triggered, then combining their sub-decisions is possible using t-conorm operations. The t-conorm, sometimes-called S-norm, is defined as a category of fuzzy operations consists of different operations (Shi 2009) like; $\text{maximum}(x,y)$; probabilistic sum = $(x + y - xy)$; bounded sum = $\min(x + y, 1)$; and drastic sum = $\{\max(x, y), \text{if } x = 0 \text{ or } y = 0 ; \text{else} = 1\}$. The t-norm is another fuzzy category (Shi 2009) and includes operations like; $\text{minimum}(x,y)$; algebraic multiplication; bounded product = $\max[0, x + y - 1]$; drastic product = $\{\min(x, y), \text{if } x = 1 \text{ or } y = 1; \text{else} = 0\}$; and nilpotent minimum (left-continuous t-norm) = $\{\min(x, y), \text{if } y > \bar{x}; \text{else} = 0\}$, where $\bar{x} = (1 - x)$.

- Last stage in the FT2 controller is the type reduction and the defuzzification operations. Both of them are using similar principles to convert the fuzzy type-2 consequent sets to FT1 sets, then to a crisp type-0 values.

3.2. The Takagi-Sugeno Fuzzy Controller

Building a fuzzy controller using the Takagi-Sugeno-Kang (TSK) technique is based on clustering the controller to multi linear or semi-linear sub-systems (Takagi and Sugeno 1985) (Sugeno and Kang 1988), (Ren et al. 2006), (Takagi & Sugeno 1985; Sugeno & Kang 1988; Ren et al. 2006). The mathematical model of each sub-system is to be defined and engaged with specific single tone consequent set. The output can be calculated as a weighted sum of consequent sets, each set is scaled by its related rule firing strength. The final output is an interpolation of sub-systems outputs.

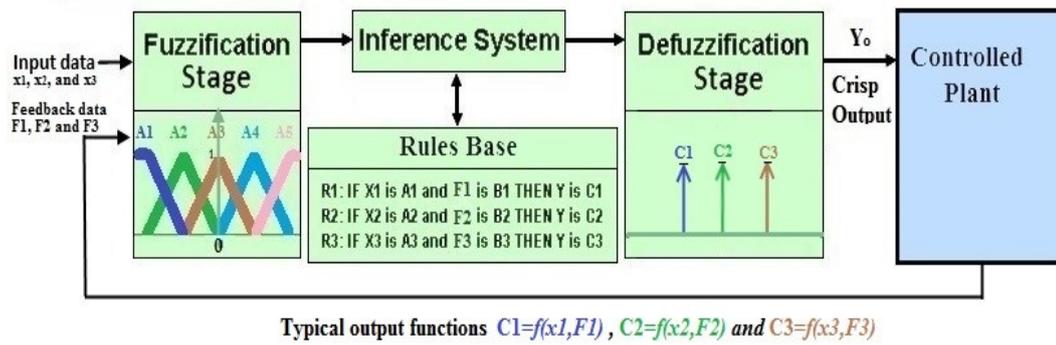


Figure 3-2 : Structure of the Takagi-Sugeno Fuzzy Control System.

The TSK fuzzy controller is categorized to three models known as model-I, model-II and model-III, depending on the type of the antecedent and consequent sets (Liang and Mendel 1999). Model-I has antecedent sets of type-2 and consequent sets are interval FT1 sets. Model-II has antecedent sets of type-2 while the consequent sets that are crisp values. Model-III is considered as a FT1 system where the input and the output are modelled using type-1 fuzzy sets only. The fuzzification and inferring process are exactly as in the Mamdani model. The type reduction process in the TSK fuzzy controller is simple because the consequence sets membership functions are linear or constants (Liang and Mendel 1999).

3.3. The Fuzzy Type-1 Controller

The FT1 controllers have very similar structures to those of the FT2 controllers but they have simpler fuzzy sets of type-1, which cannot present uncertainty. Each element of the FT1 sets has its own well-defined membership level, thus uncertainty presentation is minimum. In such case, no type reduction is required, only a defuzzification is process is necessary at the output stage of the FT1 controllers. However, the structure of the rule base and the inferring process of the FT1 controllers are identical to those of the FT2 systems. Nevertheless, in some cases, it may require

involving more rules to compensate for the missing uncertainty process that exists in FT2 systems.

3.4. Autonomous Vehicles

The autonomous vehicles are robotic systems which utilize intelligent controllers, human expertise, advanced sensing techniques, and adaptive algorithms to interact correctly with the surrounding environment while performing autonomous missions. They can replace the humans in hazardous and inaccessible places (Trabia et al. 2006); perform military and criminal confronting missions (Theodoridis and Hu 2012); and do routine tasks, in agricultural and industry sectors (Kayacan et al. 2013). These autonomous vehicles platforms can take different forms like unmanned ground vehicles (UGV); unmanned air vehicles (UAV); unmanned water-surface vehicles (USV); and autonomous underwater vehicles (AUV). The autonomous navigation missions require mutual cooperation between different sub-functions such as: obstacle avoiding; path, lane, and wall following; goal seeking; path planning; trajectory planning; traps and ambiguous status resolving, which have to be performed under the constraints of the vehicle's kinematics and dynamics. Enhancing the performance of autonomous vehicles is a continuous challenge for a range of reasons, such as environment uncertainty; high environment dynamics; environment complexity; limitation of processing speed; limitation of portable power sources; system cost limitation, plus many other difficulties. To reduce the uncertainty, different advanced sensing types can be used, such as: touch detector; ultrasonic distance estimator; infrared sensors; laser distance meter; radar scanner; thermal, optical and/or stereovision cameras, at the expense of costing, power and space. Therefore, it is good to use simple and cheap sensors equipped with good uncertainty resolving process in the control algorithms. In this study, an emphasis on enhancing the fuzzy type-2 controllers, which are efficient in resolving uncertainty, is to be conducted thus to get faster controller can do better for autonomous vehicles and similar systems that have high uncertainty and require fast action rates. In addition to that, fast fuzzy type-2 systems will enable pumping more functions in small digital controllers while still giving a short response time and achieving small errors.

3.5. Autonomous Vehicle Fuzzy Type-2 Controllers Survey

The fuzzy type-2 controllers for autonomous vehicles are to be surveyed for their controller's architecture, sensors, applied type reduction, and any other technology fused in them. The results are to be summarized, discussed, and presented graphically to identify the success and the problems associated in using the FT2 controllers in autonomous vehicles.

The first FT2 controller, performed on autonomous vehicles, was built by (Hagras 2004a; Hagras 2004b) using two different autonomous vehicle platforms. The first vehicle was equipped with 7 ultrasonic sensors and one infrared scanner. It was derived and directed by two differential independent front wheel DC motors. The second vehicle was a down-scaled car structure with GPS receiver. They were modelled on 2D bases to perform goal seeking, obstacle avoidance and wall-following behaviours for both indoor and outdoor environments. FT2 and FT1 controllers were built and optimized using genetic techniques to facilitate even comparison between them. Different obstacles and surrounding walls, with different ultrasonic absorption rates, were used in the comparison tests. The performance of that FT2 controller overcomes the FT1 controllers even when less fuzzy sets were being used. The centre of sets (COS) type reduction process was applied, as it offers less computational cost, which was considered as a priority, even when its output is not very extensive. The hierarchical decision structure has been applied to process the high priority tasks first. The tests achieved did not show the performance in highly dynamic obstacle situations and did not process the vehicle dynamics in high-speed examples.

In (Figueroa et al. 2005), a control operation is performed on a soccer robotic vehicle to track a ball using its coordination, which is gained by a simple image processing algorithm. A nonlinear interval fuzzy type-2 functioning as a proportional-differential (PD) controller, using two sets for each input, was constructed to overcome the uncertainty in the ball and the robot position data. The ball catching function is performed using tracking process, where no dynamic, motion, estimation to the ball or other soccer moves were performed. The comparison was performed with an FT1 controller, showing error deviation enhancement of 50%.

In (Baklouti and Alimi 2007), a fuzzy type-2 TSK model-I system architecture was used to perform three main operations: sub-goal access, local obstacle avoidance, and priority organiser. The sub-goals have been determined using the generalised local Voronoi diagram (Aurenhammer and Klein 1991) to cluster the obstacles according to the robot dimensions, before going on to decide the correct safe path. The obstacles are described by their local minimum and maximum limit points only. The robot receives the local minimum point as local goal to calculate the angle of the next move. The second operation is the local obstacle avoidance, which depends on six laser sensors; each is described using two fuzzy trapezoidal sets to calculate the proper obstacle-avoidance angle. The final output is coordinated using a FT1 controller with 64 rules.

In (Liu et al. 2007), continuous walking gait switching nonlinear general FT2 controllers were designed to control the three nonlinear operation modes; single support, double support and the transient control, of a biped humanoid robot. A developed fuzzy clustering modelling technique, based on variant estimation, was used to reflect the variables uncertainty effect over the fuzzy secondary membership function. The gradient decent tuning algorithm was performed to tune the system parameter. The hip and ankle gait generation and switching were controlled to preserve the stability by maintaining the zero moment point (ZMP) into the allowable boundaries. The proposed controller performance reveals a very low integral square error (ISE) and a superiority, even with the injection of high noise level, in comparison to controllers based on reinforced-learning; type-1 fuzzy sets; slide mode technique; traditional PID; and fuzzy type-2 with normal fuzzy C-mean clustering algorithm (FCM). The proposed technique captures the dispersion around the mean to achieve high accuracy, but this can be considered as an over tuning which may degrade the system's robustness. Further development is required to support the complex soft switching required for efficient running and jumping, for example what has been done by (Ahmed 2011).

In (Astudillo et al. 2007), a path following process has been performed using interval FT2 controller in a back stepping form. The differential-driven wheeled mobile kinematic constraints were considered using a mathematical model to find the instantaneous orientation. The interval FT2 controller, Mamdani architecture, calculates the driving torque for each wheel. A test with periodic disturbance, injected over the fuzzy controller output, showed a very small, negligible, effect on the performance. But,

applying that disturbance at the feedback path was not considered or evaluated. The trajectory speed was gained from a separate controller, so the fuzzy controller did not achieve the full system control.

In (Nurmaini and Hashim 2008), a FT2 controller performing edge following and obstacle avoidance operations were constructed using two wheeled, differential driven, mobile robot contains 8 ultrasonic sensors. Each sensor's inputs and the two control signals were described using three IFT2 sets. The processing power, required to overcome the type reduction delay, has been offered by AT89x55 microcontroller and 8 peripheral interface controllers, type (PIC1684a), for the ultrasonic sensors.

In (Zhang et al. 2008), a hierarchical fuzzy control architecture was proposed and presented to solve the problem of the exponential growth of fuzzy rules with respect to the controller input's count. The example of the autonomous robot, with 7 ultrasonic sensors, has been given, where each sensor signal is described using three FT2 sets. So, the representation of the full working state requires $3^7 = 26487$ rules. The processing time issue of such a huge rule base has potentially been solved by decomposing the robot tasks to low level local sub-tasks and high level strategic tasks. Two FT2 controllers were used to perform obstacle avoidance and goal seeking, based on calculating robot-obstacle and robot-goal angle's errors. A third fuzzy controller has been designed to act as an organiser or coordinator which prioritises operations at any specific time. This hierarchical control has been suggested to achieve high performance, but no systematic design procedure has yet been described.

In the work of (Chen and Yao 2011; Chen and Yao 2009), an autonomous wall-following behaviour was implemented on the Pioneer, a differential driven, robotic vehicle with 16 ultrasonic sensors. An interval FT2 set has been used and tuned based on the mean data and deviation; to get smooth action can increase gear life and reduce power consumption. A simplified centroid type reduction was proposed based on using single-tone consequence set firing levels in a procedure similar to the KM to achieve high throughput. Also, the vehicle kinematics and its dynamics have been modelled in a separate low level controller.

In (Hsiao et al. 2009), an interval FT2 controller was built for a nonholonomic wheeled robot, using sliding mode control technique, to perform trajectory tracking

while considering vehicle kinematics and dynamics. The gain adaptation was used to enhance disturbance rejection and to achieve required asymptotic stability using Lyapunov stability conditions (Jean-Jacques E. Slotine and Weiping 1991; Mohammad Khansari-Zadeh and Billard 2014).

In (Wagner and Hagraas 2009), the wheeled robot Pioneer-2 has been evaluated on-street to perform wall-following and obstacle-avoiding functions using a Z-sliced General FT2 controller tuned for the indoor environment. The front and side ultrasonic sensors were used to detect obstacles and pavement. Each sensor signal is modelled using two trapezoidal sets, for near and far. The used consequent sets were turn-right and turn-left. The general FT2 sets were formulated of four Z-slices only. The zero level slices describe the footprint of uncertainty (FOU). The general FT2 vehicle controller tests, with Z-sliced set, had shown better performance, when compared to controllers with interval FT2 and FT1 sets, using wet asphalt roads that are covered with leaves, pebbles, and dust.

In (Siti Nurmaini et al. 2009; Nurmaini and Hashim 2009; S. Nurmaini et al. 2009), a weightless neural network object identifier (Aleksander et al. 2009) has been fused with an interval FT2 controller to enhance obstacle avoidance and wall-following autonomous behaviours. The usage of this object classifier is important to reduce uncertainty and system process cost. The weightless neural classifier, which supports one-shot training, collects all sensors data to identify the obstacle shape. Actions were taken accordingly to avoid collision possibility. Low hit rate had been achieved by fusing the two controllers, in compare to the FT1 and the interval FT2 controllers, in complex and noisy environments.

In (Kang et al. 2009), a stereo vision ego-motion estimation compensator for an autonomous humanoid robot to enhance recognition level was built using a fuzzy type-2 controller. The direct least squares technique (Fitzgibbon et al. 1999) is used to fit an ellipse around the extracted useful feature to calculate the rotation and the displacement between the objects, in the successive images. The FT2 compensator accuracy and the computation time were compared to the iterative closest point technique and the scale-invariant feature transformed technique to show better performance. The achieved accuracy and good processing time promote the proposed technique for the correction of the autonomous vehicle's vision.

In both (Leottau and Melgarejo 2010) and (Sidhu et al. 2012; Leottau and Melgarejo 2011) a similar work has been performed using an interval FT2 controller to overcome the non-holonomic vehicles steering constraints. The relation between the relative horizontal position error and the orientation error has been described as a FT2 controller. The required steering angle is being fed to a low-level controller, which represents the vehicle constraints to achieve it while the vehicle is moving. In (Leottau and Melgarejo 2011), the controller is tested with different types of set reduction methods taking into account the error and the stability. These tests have shown superior performance of the centroid type reduction over other reduction types.

In (Linda and Manic 2011), a fuzzy type-2 wall-following controller was built with Lego-Robot platform and tuned by measuring and modelling the sensors uncertainty. Each ultrasonic sensor uncertainty has been evaluated, in its active range, for different reflective materials. This FT2 controller shows low absolute error in comparison to a traditionally designed FT2 controller and FT1 controller. The FT1 controller sets, also, have been tuned using the sensor uncertainty. The uncertainty propagations throughout the system have been evaluated and used to define the consequent set boundaries.

In (Mbede and Melingui 2012) autonomous goal-seeking and obstacle avoidance were performed using a controller with Z-sliced general fuzzy type-2 sets, such as to avoid static and dynamic obstacles. The slices, being used, were just two, at 0.5 & 1, and still achieved better performances than interval FT2 controllers. The omnidirectional, three wheel, Robotino robot, which is supported with 9 distance sensors, was used as a platform for the tests. A single colour camera was used to identify the goal, while no obstacles prediction was performed for the dynamic obstacle avoidance.

In (Baklouti et al. 2012), the KheperaII robot, which has two wheels and is equipped with six infrared distance sensors, had been used and simulated using SIMROBOT toolbox software to evaluate different FT2 controllers. TSK and the Mamdani models were used for the different interval set widths and the total performance being evaluated for path smoothness, distance to the goal, and task time. The best performance has been achieved using Mamdani architecture with interval sets width of 40%.

The work in (Hsu and Juang 2012), has been proposed using a simple and efficient ant-colony optimization (ACO) technique to tune the global parameters of the interval FT2 controller, using cost function dependent on system error. They first discretized each system parameter to N sections then formed a segments array of size $\times D$, where D is the system total variables to be tuned, that being included in the ant path. The exploration and exploitation of the ACO algorithm uses the cost-fermion function, which gradually leads to a global optimal solution. The system tests, using autonomous wall-following, have shown low error deviation.

In (Chang et al. 2013), robotic team members are controlled individually by two fuzzy controllers to follow one leader which orders them, while avoiding collision, conjunction, and being trapped in local minima. The first interval FT2 controller performs team pattern construction process, dynamically. It is mixed with neural network layers to substitute the type reduction and to achieve tuning using the gradient descent method. The second interval FT2 controller uses a potential based separation method to keep a safe distance between the team members and to take the robot out of any local minima that generate a stop case. Best response has gained using the FT2 controller, in comparison to the traditional consensus potential-based algorithm.

In (Kumbasar and Hagraas 2013), two FT2 proportional-integral-differentiator controllers (FT2PID) were designed to control non-holonomic robotic vehicles in the path following process. The external FT2PID, the first controller, calculates the required vehicle steering angle while it propagates at a constant speed to its target, which is defined by x, y, θ . The second controller preserves the required angle by controlling the torque necessary to correct the angle at different loadings. Performance error has compared to traditional-PID, FT1-PID, and Interval FT2PID controllers, to show the superiority of the interval FT2-PID technique. All the controllers were optimized using Big-Bang Big-Crunch technique (Erol and Eksin 2006) to minimize the integral absolute error function. The robot kinematic constraints were included but not the dynamic constraints.

In (Kayacan et al. 2013), an extensive piece of work was undertaken towards achieving an autonomous agricultural robotic tractor, using sliding mode technique in an interval FT2-Neural Network controller. The tractor kinematic and dynamic constraints were considered to minimize errors during the trajectory-following process. A state

estimator, to calculate the tractor orientation, was built using extended Kalman filter. The Real Time Kinematic Global Positioning System (RTK-GPS) is used for coordination and steering angle estimation, achieving less than 1mm error. The real-time learning was used by tuning the neural parameters while preserving the Lyapunov stability criteria. The neural network type reduction was chosen to achieve high throughput reached to 20Hz. The work is great, but did not use any obstacles sensing or cameras to interact with the surrounding environment because it is assumed that all the navigation is to be done in the open fields; thus, it is still dangerous and more work will have to be done to include all the autonomous sub-functions.

In (Melingui and Chettibi 2013), an omnidirectional mobile robot was used to perform target seeking while avoiding obstacles, left/right corners, corridors and U-shapes, in the indoor dynamic environment. These operations were performed using four techniques; (i) Artificial potential field path planning (APF), (ii) Interval FT2 controller, (iii) Switched interval FT2 controller plus artificial potential field (IFT2C+APF), (iv) Fused artificial potential field and interval FT2 controllers (T2FP). The robot has 9 distance infrared sensors, for obstacle detection, and one coloured camera to identify the goal position. Their tests have shown a better performance in regard to seeking time and path smoothness for the fused T2FP controller over the other experimented techniques. The work covers different strategies, but still there is no learning strategy implemented to overcome tricky obstacles.

3.6. Survey Summary of Autonomous Fuzzy Type-2 Controllers

The first implementation of FT2 controllers, in autonomous vehicles, was done by (Hagras 2004a; Hagras 2004b) to achieve obstacle avoidance, wall-following, and goal-seeking behaviours using ultrasonic sensors and hierarchical structure, which was tuned using a genetic algorithm. The hierarchical FT2 structure has been further investigated for autonomous vehicles in (Zhang et al. 2008) to achieve lower computational cost. A moving object tracking autonomous process has performed using FT2PD controlled in (Figueroa et al. 2005). Then, the first TSK FT2 controller was

designed to perform obstacle avoiding and goal seeking by using Voronoi diagrams (Aurenhammer and Klein 1991) to cluster the obstacles (Baklouti and Alimi 2007). In (Liu et al. 2007), the FT2 controller was used to supervise autonomous biped motion, while in (Kang et al. 2009) the ego-motion was isolated to enhance the stereoscopic objects recognition. In (Leottau and Melgarejo 2010) (Sidhu et al. 2012; Leottau and Melgarejo 2011), the steering constraint of the non-holonomic autonomous vehicle has been processed using a FT2 controller for forward and backward moves. The autonomous vehicle controller design is proposed by (Linda and Manic 2011) based on modelling the sensor uncertainty. The backstepping controller has been built by (Astudillo et al. 2007) using the FT2 technique to perform path following process. In (Nurmaini and Hashim 2008), the microcontroller AT89x55 has been used to perform obstacle avoiding and goal seeking. In (Chen and Yao 2009), the FT2 controller smooth action, during the autonomous operations, has been achieved to increase the vehicle's gear life and to reduce power consumption. In (Wagner and Hagrais 2009), the z-sliced general FT2 sets has been used in autonomous vehicle, which achieved great results with just two sensors, while in (Mbede and Melingui 2012) a very simple general FT2 set, z-sliced only to two layers, has been designed and compared to controllers using interval FT2 sets. In (Siti Nurmaini et al. 2009) (Nurmaini and Hashim 2009) (Nurmaini and Tutuko 2011), a neural network identifier has been used to enhance the FT2 autonomous actions. In (Hsu and Juang 2012), the ACO algorithm has been used to optimise the parameter of the FT2 autonomous vehicle controller. The robotic team controller has been successfully achieved in (Chang et al. 2013) using FT2 techniques to execute different movements while avoiding any collisions and local minima situations. The autonomous path following process is performed in (Kumbasar and Hagrais 2013), using two cascaded FT2PID controllers to simplify processing the vehicle nonlinearity. The autonomous agricultural tractor has been controlled in (Kayacan et al. 2013) to move autonomously at the fields using a general FT2 controller. In (Melingui and Chettibi 2013), complex obstacles have been avoided by fusing the potential field path planning with the interval FT2 controller.

3.7. Autonomous Fuzzy Type-2 Controller Survey Statistics

The survey of the FT2 autonomous vehicle controllers, presented in Figure 3-3 of the last 10 years, shows great involvements of the Interval FT2 sets, which are taking place as Mamdani FT2 controllers. This is mainly due to the controller simplicity, simple human expert utilization, and ability to process uncertainty. In the cases where only input data has uncertainty and the system can be presented using linear sub-sections, the TSK fuzzy model has been preferred because of its fast type reduction process. But, when human expertise and the mathematical model are not available, then Mamdani fuzzy architecture is preferred and to be tuned using either manual technique; genetic algorithms (GA); Neural-Networks (NN); Swarm intelligence; or bio-intelligent techniques (BIT). These Mamdani FT2 architectures are preferred, but still, they require high processing power, during the type reduction computations. The type reduction cost went down dramatically after 2007 when the Enhanced Karnik–Mendel (EKM) type reduction algorithm has been proposed. The impact of this type reduction algorithm on autonomous vehicle controllers can be seen in the histogram chart Figure 3-4 during the period from 2008 till 2009 where such researches have increased rapidly. However, the total number of researches during the last decade is still small due to the bottlenecks exist in the Mamdani fuzzy type-2 controllers, which mainly caused to the type reduction computations. This is, strongly, recommends and encourages finding better type reduction algorithms can perform more efficient in fast systems and enable using more tasks in smaller and cheaper processors haveing low processing power.

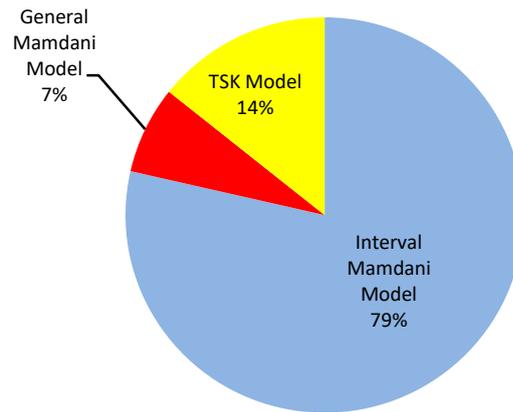


Figure 3-3: Interval FT2 sets usage rate in autonomous vehicles control researches, during 1994-2014

Also, it can be seen in the pie-chart Figure 3-5, how important the centroid type reduction technique is, which can achieve smooth controller output that required in autonomous vehicles. In addition, the autonomous controller survey shows, during the last three years, that most of the researches do focus on using general fuzzy type-2 controllers and their tuning techniques. This research shows that how more efficient techniques are required to reduce the type reduction cost, especially when there are real-time actions or optimization algorithms.

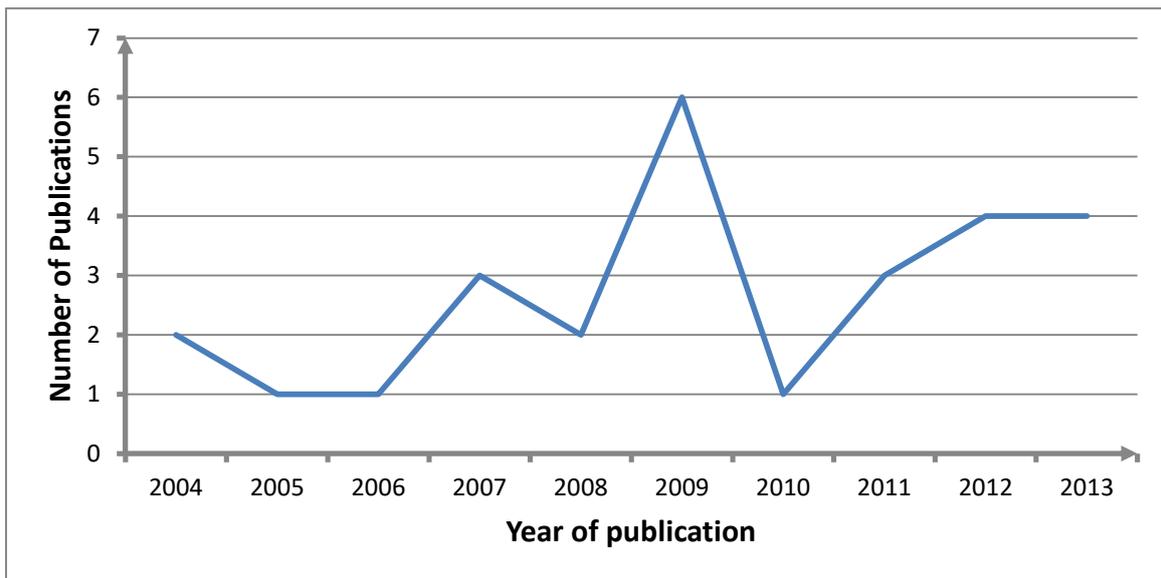


Figure 3-4: Autonomous FT2 Controllers researches during the past ten years.

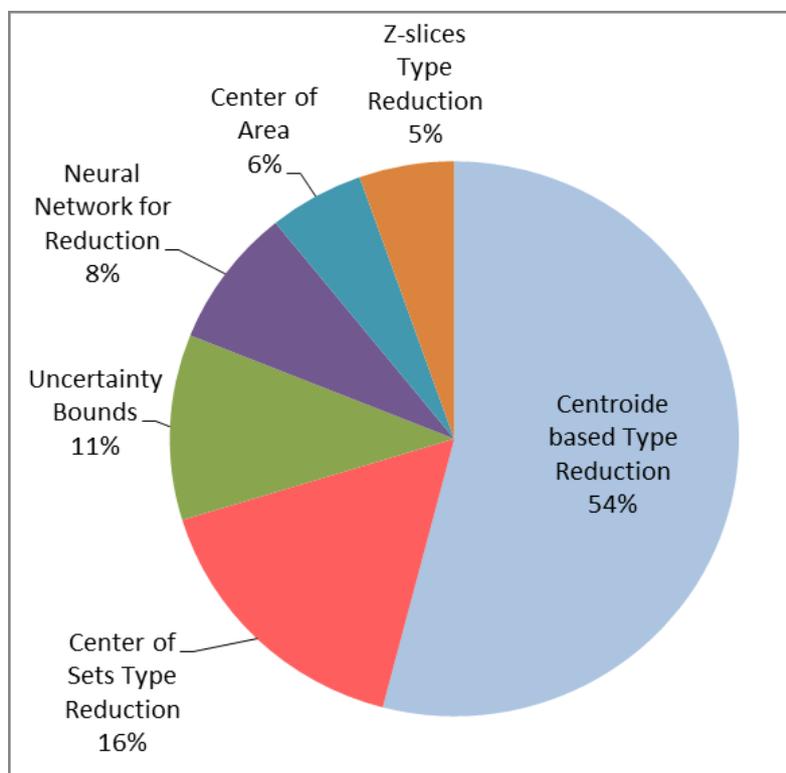


Figure 3-5: Type reduction suitability for autonomous FT1 control applications.

Chapter 4: Type Reduction Approximation

4.1. Research Goals and Methodology

The digital fuzzy type-2 controllers have many attracting features required for autonomous vehicles such as high reliability, uncertainty process, simple utilisation to human expert, adaptability, and high nonlinearity process. The main drawback of the FT2 controllers is its intensive computational cost during the type reduction process (Liu et al. 2012). Therefore, a type reduction with low execution time is required for high-speed systems like autonomous vehicles to assure performing faster to achieve low error and enabling more functions in smaller and cheaper processors. This research is aiming to develop better type reduction methods to be used in the Mamdani fuzzy type-2 controller architectures. The Mamdani fuzzy type-2 controllers are of the most common and preferred architectures; supporting high uncertainty resolving, and simple knowledge representations, and understandable deduction methods very similar to humans' way of thinking. The enhancements are to be achieved firstly by using the adaptive integration techniques into the type reduction stage thus to cut the computations cost while keeping the required system error at the designed levels. In addition, it is aimed to carefully studying the most important and efficient type reduction methods thus to locate and eliminate any possible redundant calculations. The proposed type reduction methods are going to be evaluated regarding their performance speed and their calculations' error. The evaluations of the new type reductions will be performed with respect to their original type reduction algorithms to get an unbiased vision about the resulting enhancements, in the speed, and any payed penalties that appear as error. The performance evaluations are going to be performed at the type reduction stages then at the full system level. The autonomous ground vehicle fuzzy type-2 controller is chosen

for the full system evaluations. This type of controller is suitable because of its vulnerability to high uncertainty level and its need to perform at high speed, which will require a fast controller throughput to maintain a low performance error.

4.2. Type Reduction Development

Significant efforts have been made, which can be seen in the literature review of chapter 2, to reduce the computation cost of the type reduction in the fuzzy type-2 systems but the flexible type reductions techniques like the EKM still have long delays. Also, it can be seen that the core computation behind most type reduction and defuzzification techniques is the COG which is used to generate a weighted average decision, as:

$$COG = \frac{\int_{y_0}^{y_N} y \cdot f(y) dy}{\int_{y_0}^{y_N} f(y) dy} \approx \frac{\sum_{n=1}^N y_n \cdot \mu_n}{\sum_{n=1}^N \mu_n} = \frac{I_{Num}}{I_{Den}} \quad (4-1)$$

The output set discretisation level would define the computation cost of the fuzzy controllers, where two additions and one multiplication have to be performed on every discrete set element. Practically, an accuracy of 1×10^{-3} , of the full operating range, is required for electromechanical and autonomous vehicle controllers (Zhu et al. 2012; Gomez and Jamshidi 2010; Kovacic and Bogdan 2010; Saleh et al. 2009; ActivMedia Robotics 2006). This accuracy is the degree of system achievement measured with respect to the total sense divisions in the working-range. This means a digital discretization level of 1000 elements is sufficient to achieve most of the practical accuracy cases. Thus, a tabulated discrete set definition of 1000 elements as $\mu_n = f(y_n)$ is simple and adequate to be used for the COG evaluation within a fuzzy controller. The COG can be evaluated using closed integration forms, as in (Zimmermann 1991), for cases of well-defined fuzzy shapes like Gaussian sets. But, the most general cases for fuzzy sets are non-simplex set shapes, especially those ones whom been shaped by tuning algorithms, as in (Hsu and Juang 2012). The integration cost can be reduced by approximating the slightly curved segments to straight lines, so as the under curve area

can be evaluated using fewer points while maintaining the integration accuracy (Gonnet 2012). To evaluate the fuzzy controller output error, which caused by such approximation, let's have a look on the numerical integration error formula using N points, over a function range $[a, b]$ using Riemann right point sum, which is used in many fuzzy decision making due to its simplicity.

$$I = \int_a^b f(y) dy = \sum_{n=1}^N f(y_n) \cdot \Delta y ; \quad y_0 = a, y_N = b \quad (4-2)$$

$$\Delta y = \frac{(b - a)}{N} \quad (4-3)$$

The Riemann sum integration error for a smooth continuous function over a period $[a, b]$ will proportion to the step size and the function average slope $\tilde{f}'(x)$.

$$error = \frac{\Delta y}{2} \cdot (f(b) - f(a)) = \frac{\Delta y}{2} \cdot |\tilde{f}'(x)| \quad (4-4)$$

This Riemann sum integration error is considered high if it is compared to the Trapezoidal rule, which has a low error found to be close to the Simpson-rule error in many practical cases (Cruz-Uribe et al. 2002). Also, the Trapezoidal rule's total performance speed, in COG related type reductions, is shown to be better than Simpson-rule and some other high order rules (Liu et al. 2012).

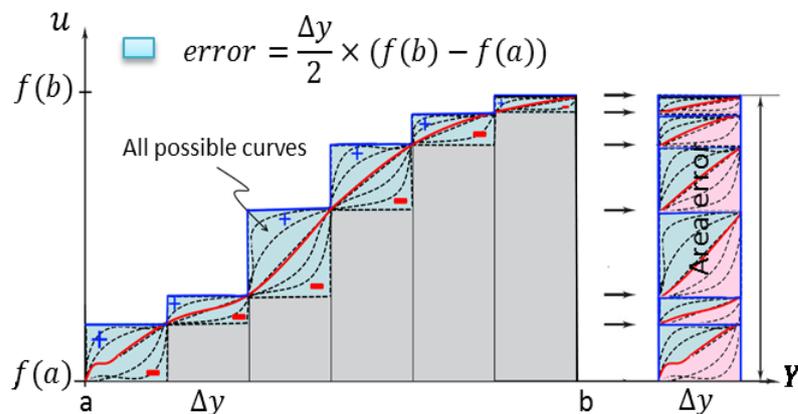


Figure 4-1: Riemann sum maximum error limits for evenly distributed points.

4.3. Basic Numerical Quadrature Rules

There are different quadrature rules and each one has accuracy and costs specific to its computation. The most common types, ordered according to their complexity, are the first order Newton Cotes; the Simpson rule; the Trapezoidal rule; and Riemann midpoint sum. They approximate the definite integration using equally spaced segments. The high order quadrature rules have low approximation error, but high computation cost, which can cause high delay if used in fuzzy type-2 controllers. The trapezoidal rule computations for N discrete points require dividing the first and the last points by two then adding them to the rest of the points, as:

$$I_N^T = \frac{(b-a)}{N} \left[\frac{f(y_1) + f(y_N)}{2} + \sum_{n=2}^{N-1} f(y_n) \right] \quad (4-5)$$

The integration error, of the Trapezoidal rule over the period $[a, b]$, is defined as in equation (4-6).

$$Error_N^T = \frac{|(a-b)^3|}{12N^2} |f''(y)| \quad (4-6)$$

This error has been evaluated upon assumption of smooth function and that average second order derivative $|f''(y)|$ over the integration period does exist and continues while any higher order derivatives are zero or negligible.

The Simpson's 1/3 (3/8) quadrature rule is based on approximating parts of the total curved function $f(y)$ using 3(4) points for each part and performing the following calculations individually, as:

$$I_3^{S1/3} = \frac{\Delta y}{3} (f(y_0) + 4f(y_1) + f(y_2)) \quad (4-7)$$

$$I_4^{S3/8} = \frac{3\Delta y}{8} (f(y_0) + 3f(y_1) + 3f(y_2) + f(y_3)) \quad (4-8)$$

In this case, each point has to be pre-scaled then added to the rest points. This idea is generalised in the high order Newton-Cotes quadrature closed formulas, where the numerical integration uses chunks of weighted sums, as below, where each chunk contains $N+1$ points to approximate a function of order N .

$$I_N^{NQ} = \sum_{n=1}^N w_n \cdot f(y_n) \quad (4-9)$$

Each point weight can be calculated for the function of order N using Lagrange polynomial as:

$$w_n = \int_a^b \left[\prod_{\substack{j=0 \\ j \neq n}}^N \frac{x-x_j}{x_n-x_j} \right] dx, j = 0,1,2, \dots, N \quad (4-10)$$

These weights w_n are to be evaluated for every sub-interval $[a, b]$, of the total integration range $[A, B]$; thus, high delay will result if this is used for fuzzy type reduction because fuzzy set shapes are changing continually by the rules firing levels.

4.4. Adaptive Quadrature Rules

Adapting each quadrature rule period size according to the function curvature is a useful technique that can reduce the total computation cost. Adaptive quadrature (AQ) methods can be performed in different ways based on how to evaluate the integration error (Gonnet 2009). A long start period can be chosen at the beginning; then, if the error is significant, then integration periods have to be subdivided into two equal periods. The integration and the error, for the new periods, have to be re-evaluated. The initial interval integration can usually be used to evaluate the next smaller intervals in order to maintain high calculation efficiency. The error evaluation of the adaptive quadrature methods at each step requires extra computation which is considered a penalty that may be compensated if enough smooth sub-sections are executed within the total integrant period. Practically, fuzzy sets contain a mix of sharp and smooth sections, therefore, a

positive time gain using AQ can be generated. However, using the AQ methods have a risk of delivering incorrect results (Gander and Gautschi 2000) if care is not taken off the integrant function characteristics like its degree and its maximum slope change. Such errors are considered in this research, thus a closer examination of it has to be made, starting from the analog trapezoidal rule integration and its error, over an interval $[a, b]$. The integration error can be evaluated for a function $f(y)$ using the bi-part integration technique, utilising the function derivations only (Cruz-Urbe and Neugebauer 2003), as follows:

$$I_{ab} = \int_a^b f(y)dy \Rightarrow I_{0h} = \int_0^h f(a+y)dy \quad ; \quad h = b - a \quad (4-11)$$

and

$$I_{ab} = I_{h0} = \int_h^0 f(b-y)dy \quad (4-12)$$

but

$$e_{T[a,b]} = \frac{1}{2}(I_{0h} + I_{h0}) - \frac{h}{2}(f(a) + f(b)) \quad (4-13)$$

getting

$$e_{T[a,b]} = \frac{(b-a)^3 [f''(b) - 2f''(a)]}{12} \quad (4-14)$$

Substituting $f''(a)$ and $f''(b)$ by $f''(\bar{x})$, which represents the average of the function's second derivative over the interval $[a, b]$, then it is possible to re-write the trapezoidal integration error as:

$$e_{T[a,b]} = \frac{-(b-a)^3}{12} f''(\bar{x}) \quad (4-15)$$

This can be expressed for interval of size equals to $h = \frac{b-a}{N}$ over a total of N sub intervals, as:

$$e_{TN} = N \times e_{Th} = N \frac{-(b-a)^3}{12N^3} f''(\bar{x}) = \frac{-(b-a)^3}{12N^2} f''(\bar{x}) \quad (4-16)$$

This error is in an absolute form and does not depend on integration results or integration period length. Using this global error to form adaptive quadrature rules has a negative effect on the calculation speed, but a positive effect on integration reliability where a search operation has to be performed in every cycle to find the highest error segment (Gonnet 2009). A local error-locating scheme is another technique, which is less

expensive, and has been used successfully by many researchers as in (Gonnet 2009) (Barden 2013) (Doncker et al. 1996). It is based on downscaling the global error linearly, according to the local segment size, in relation to the total integration period, thus eliminating the search process. Here, the sum of the local integration errors is equal to or smaller than the global required error. This error localisation is a distribution process, known to be safer, and can perform more adequately except for functions containing a singularity point (Doncker et al. 1996). Thus, using local error technique for the fuzzy sets type reduction, and aggregation is totally safe because the fuzzy sets do not contain any singularities. Considering the results given by (Liu et al. 2012), which shows that trapezoidal rule is outperforming the other high order quadrature rules, if it is used for fuzzy type reductions, due to its good practical accuracy and its low computation cost. A similar result in (Carluccio and Albani 2011) shows that the speed of adaptive quadrature rules for low order functions is higher than that at high order functions. Thus, one can expect that using an adaptive trapezoidal rule for fuzzy type reduction will enhance their computation cost while maintaining the required system accuracy.

4.5. Suggested Error Evaluation

The usual dividing of the integration period into two equal intervals, during adaptive integrations, when the quadrature local error exceeds its limit (Gonnet 2009), means that performing two integrations one of them can be unnecessary. Searching for the appropriate interval length successively to assure that integration local error is preserved can be another possible approach. In the next sections, this is going to be analysed and evaluated for fuzzy type reduction purposes. The trapezoidal rule error proportional nature, where integration error is zero for straight segments and high for highly curved segments, facilitates using the successive search by evaluating the local integration error on a variable integration period. It is required to find the maximum interval length, which still has an integration error within the required limit. In the successive search technique (McNeill et al. 2011); if the required criteria level is not exceeded, then the search step size is doubled, otherwise, it is halved. This algorithm has been used frequently for high speed Analog to Digital Converters (ADC) because of its

simplicity and ability to be realized using simple hardware, consuming low power (Saberi and Lotfi 2014).

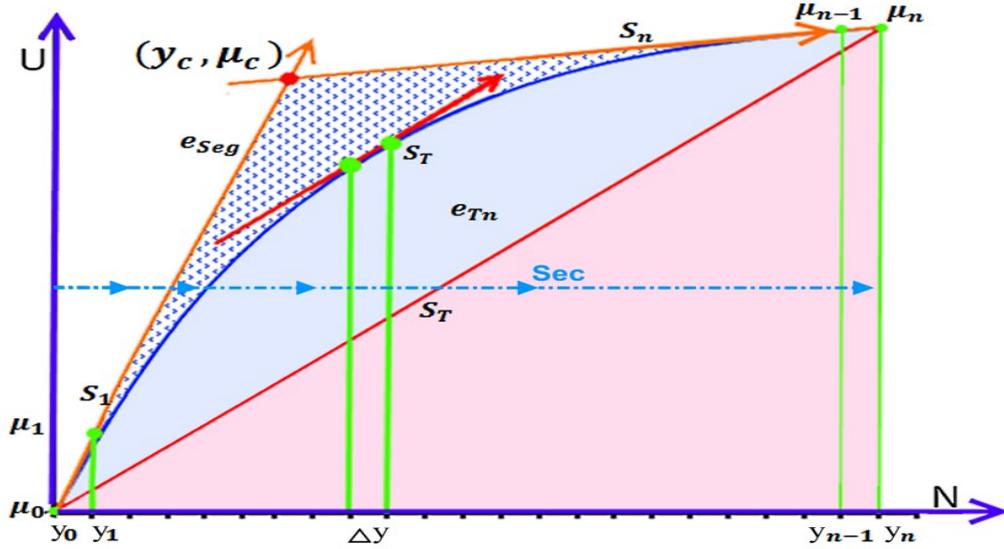


Figure 4-2: Trapezoidal error over one long interval.

This is going to be used here to find the correct interval length in the successive adaptive quadrature algorithm while assuming that the fuzzy membership is defined using discrete equally spaced points as: $f(y_n) = \{\mu_0, \mu_1, \mu_2, \dots, \mu_N\}$, which it is one of the simple and most common definitions that's being use for digital systems. The goal is to find the maximum possible interval length successively within a convex fuzzy set. This can be done by using the well-known trapezoidal-rule error over a period $[a, b]$, which is defined by equation (4-15), then re-formulating this for a temporary segment section S_{sc} by using its first point $(y_{0_{sec}}, \mu_{0_{sec}})$ and its last point $(y_{n_{sec}}, \mu_{n_{sec}})$. This re-shaping begins from the definition of the slope at the start point of the temporary section S_{sc} , as follows:

$$S_{1_{sec}} = \frac{(\mu_{1_{sec}} - \mu_{0_{sec}})}{(y_{1_{sec}} - y_{0_{sec}})} = \frac{(\mu_{1_{sec}} - \mu_{0_{sec}})}{\Delta y} \quad (4-17)$$

This section can be defined in different lengths as multiplicands of the system discretisation level $(n_{sec} \cdot dy)$. Its slope at the end-point is equal to:

$$S_{n_{sec}} = \frac{(\mu_{n_{sec}} - \mu_{(n-1)_{sec}})}{(y_{n_{sec}} - y_{(n-1)_{sec}})} = \frac{(\mu_{n_{sec}} - \mu_{(n-1)_{sec}})}{\Delta y} \quad (4-18)$$

Then, as shown in Figure 4-2, from the average second-derivation of the curved function over this section:

$$f''(\alpha) = \frac{S_{n_{sec}} - S_{1_{sec}}}{(n - 0)_{sec} \cdot \Delta y} \quad (4-19)$$

Can get the average slope of the straight line that joining the start and end points of the curved section to be;

$$S_{T_{seg}} = \frac{(\mu_{n_{sec}} - \mu_{0_{sec}})}{(y_{n_{sec}} - y_{0_{sec}})} = \frac{(\mu_{n_{sec}} - \mu_{0_{sec}})}{n_{sec} \cdot \Delta y} \quad (4-20)$$

This slope, according to the mid-point interpolation theorem (Rao 2007), can be approximated by:

$$S_{T_{sec}} \approx \frac{S_{n_{sec}} + S_{1_{sec}}}{2} \quad (4-21)$$

Getting: $S_{n_{sec}} \approx 2S_{T_{sec}} - S_{1_{sec}}$ (4-22)

Substitute in (4-19) to get; $f''(\alpha) \approx \frac{2(S_{T_{sec}} - S_{1_{sec}})}{n_{sec} \cdot \Delta y}$ (4-23)

Re-write using the start and end points: $f''(\alpha) \approx \frac{2[(\mu_{n_{sec}} - \mu_{0_{sec}}) - n(\mu_{1_{sec}} - \mu_{0_{sec}})]}{n^2 \cdot (\Delta y)^2}$ (4-24)

Using equation (4-15) to get: $e_{Tn} = \frac{-n^3 \cdot (\Delta y)^3}{12} f''(\alpha)$ (4-25)

Re-write, using the section terminal points:

$$e_{Tn} \approx \frac{-n \cdot \Delta y [n(\mu_{1_{sec}} - \mu_{0_{sec}}) - (\mu_{n_{sec}} - \mu_{0_{sec}})]}{6} \quad (4-26)$$

The local error limit is the unit-length error multiplied by the local interval length. The unit-length error is the global error divided by the equally spaced total integration divisions, as follows:

$$e_{unitL} = \frac{e_{Trapez}}{N \cdot \Delta y} \quad (4-27)$$

Thus, error threshold for an interval of length $n \cdot \Delta y$ should be bounded as:

$$e_{Tn} \leq e_{unitL} \cdot n \cdot \Delta y \quad (4-28)$$

Substituting in equation (4-26) to get the local integration error constraints as:

$$|n(\mu_{1_{Sec}} - \mu_{0_{Sec}}) - (\mu_{n_{Sec}} - \mu_{0_{Sec}})| \leq \frac{6 \cdot |e_{Trapez}|}{N \cdot \Delta y} \quad (4-29)$$

$$n(\mu_{1_{Sec}} - \mu_{0_{Sec}}) - (\mu_{n_{Sec}} - \mu_{0_{Sec}}) \leq |6 \cdot e_{Tp.u.l}| ; \quad for \Delta y = 1 \quad (4-30)$$

This error constraint is correct for functions that have gradual slope change over their domain. However, this is not the case with fuzzy sets that have horizontal cuts, which caused by different fuzzy threshold levels. The error, in this case, is bounded as shown in Figure 4-2 by the dotted area, which has to be considered in order to keep the total error under the required limit. The maximum possible integration error for such shapes with horizontal cuts can be in its maximum value when the point y_c falls in the middle distance between y_0 and y_n . The integration error for this case can be evaluated as:

$$e_{Seg} = (y_{C_{Sec}} - y_{0_{Sec}}) \frac{(\mu_{C_{Sec}} - \mu_{0_{Sec}})}{2} + (y_{n_{Sec}} - y_{C_{Sec}}) \frac{(\mu_{n_{Sec}} - \mu_{0_{Sec}}) + (\mu_{C_{Sec}} - \mu_{0_{Sec}})}{2} - (y_{n_{Sec}} - y_{0_{Sec}}) \frac{(\mu_{C_{Sec}} - \mu_{0_{Sec}})}{2} \quad (4-31)$$

Re-formulating and using the slope of the start and end points:

$$S_0 = \frac{(\mu_c - \mu_0)}{(y_c - y_0)} = \frac{(\mu_c - \mu_0)}{0.5(y_n - y_0)} \quad (4-32)$$

$$S_n = \frac{(\mu_n - \mu_c)}{(y_n - y_c)} = \frac{(\mu_n - \mu_c)}{0.5(y_n - y_0)} \quad (4-33)$$

Dividing them by each other equation to get:

$$\frac{S_0}{S_n} = \frac{(\mu_c - \mu_0)}{(\mu_n - \mu_c)} \quad (4-34)$$

To get point μ_c definition as:

$$\mu_c = \frac{S_n \cdot \mu_0 + S_0 \cdot \mu_i}{S_0 + S_n} \quad (4-35)$$

Using equation (4-21) to get μ_c in term of total slope S_T as below:

$$\mu_c \approx \frac{2S_T \cdot \mu_0 - S_0 \cdot \mu_0 + S_0 \cdot \mu_n}{2S_T} \quad (4-36)$$

This can be simplified to:

$$\mu_c \approx \mu_0 + \frac{S_0}{2S_T} (\mu_n - \mu_0) = \mu_0 + \frac{n}{2} (\mu_1 - \mu_0) \quad (4-37)$$

Substitute in equation (4-31) to get:

$$e_{seg} \approx \frac{n\Delta y}{4} [n(\mu_1 - \mu_0) - (\mu_n - \mu_0)] \quad (4-38)$$

Thus, the error will stay under the required boundaries, if the following relation is held:

$$n(\mu_1 - \mu_0) - (\mu_n - \mu_0) \leq \frac{4 \cdot |e_{Trapez}|}{N \cdot \Delta y} \quad (4-39)$$

Or,
$$n(\mu_1 - \mu_0) - (\mu_n - \mu_0) \leq |4 \cdot e_{Tp.u.l}| ; \quad \text{for } \Delta y = 1 \quad (4-40)$$

This relation defines the Trapezoidal error in stricter form if compared to relation (4-30), which it is the result of a traditional trapezoidal error equation (4-15).

4.6. Error in Fuzzy Type Reduction

The fuzzy controller's accuracy can be decided depending on the controlled plant parameters. The final system's accuracy cannot be much better than the mechanical plant accuracy even if high accuracy digital controller or interpolation techniques are used. Such cases can be seen in many practical autonomous vehicle controllers, like those in (Hu and Li 2014; Xue et al. 2012; Fang et al. 2011; Zhao et al. 2012), and in many electromechanical systems as in (Abdelmajid et al. 2011), where their ultimate electro-mechanical goal accuracy was 1/1000. Also, the accuracy of any digital controller is bounded by the word size of its digital-to-analog (DAC) and analog-to-digital (ADC) converters. These ADC and DAC can be found at low prices and good performance speed mostly in the ranges of 8,10,12,14 and 16 bits (Analog Devices Inc. 2005). Thus, to avoid any ineffectual extra computation in the fuzzy type-2 controllers, an output word size of 10 bits is to be chosen. This word size offers a digitization level of 1024 elements, which is sufficient for the mechanical systems' accuracy, described above. However, the IFT2 controller output is evaluated by averaging its left and right switching points; thus, output error associated with one ideal fuzzy output Out_0 is defined by:

$$Out_0 + e_{OutF2} = \frac{(y_l \mp e_l) + (y_r \mp e_r)}{2} = \frac{y_l + y_r}{2} + \frac{|e_l \mp e_r|}{2} = Out_{FT2} \quad (4-41)$$

Here, e_l and e_r are the absolute errors associated with the left and the right extreme points of the type reduced set. The controller output absolute error is e_{OutF2} .

$$e_{OutF2} = \frac{|e_l| + |e_r|}{2} \quad (4-42)$$

$$\text{If } |e_l| = |e_r| \quad \text{Then } e_{OutF2} = |e_l| = |e_r| \quad (4-43)$$

The error of these points is also affected by the allowance of the stop condition of the fixed-point iterations used in the KM type reduction. Their calculation, which has to be performed iteratively, is the (COG). The COG denominator integration error allowance e_{D_atw} , which is associated with the ideal integration result I_{Den0} , can be defined, for simplicity, using a normalized discrete step size of $\Delta y = 1$ as follows:

$$I_{Den0} + e_{D_atw} = \sum_{n=1}^N (\mu_n + e_{\mu_n}) \cdot \Delta y = \sum_{n=1}^N \mu_n + \sum_{n=1}^N e_{\mu_n} \quad (4-44)$$

The symbol e_{μ_n} defines the error associated with every set sample, which is the calculation allowance for each unit length. The error in defining the set membership levels is assumed zero; then only the error limit per unit length $e_{Dp.u.l}$ will dominate the calculation.

$$e_{\mu_n} = e_{Dp.u.l} = \frac{e_{D_alw}}{N} \quad (4-45)$$

Performing the same calculations for the numerator, as shown below, using the fuzzy membership error e_{μ_n} and the normalized $\Delta y = 1$ to get:

$$\begin{aligned} I_{Num} + e_{N_alw} &= \sum_{n=1}^N (\mu_n + e_{\mu_n}) \cdot y_n \cdot \Delta y \\ &= \sum_{n=1}^N y_n \cdot \mu_n + \sum_{n=1}^N y_n \cdot e_{Dp.u.l} \end{aligned} \quad (4-46)$$

And, by using the following approximation for a discretised output axis contains N elements y_n can get:

$$\sum_{n=1}^N n \cdot e_{Dp.u.l} \approx \frac{N^2}{2} e_{Dp.u.l} = \frac{N}{2} e_{D_alw} \quad (4-47)$$

Resulting:

$$I_{Num} + e_{N_alw} \approx \left[\sum_{n=1}^N y_n \cdot \mu_n \right] + \frac{N}{2} e_{D_alw} \quad (4-48)$$

However, the final COG error has to be evaluated using the numerator and the denominator errors in percentage form because they are associated with two dividend operands (Castrup and Castrup 2010), as:

$$\begin{aligned} \frac{|e_{COG}|}{|COG|} &= \frac{|e_{Num}|}{|I_{Num}|} + \frac{|e_{Den}|}{|I_{Den}|} \\ &= \left| \frac{\frac{N}{2} e_{D_alw}}{I_{Num}} \right| + \left| \frac{e_{D_alw}}{I_{Den}} \right| \\ &= \left| \frac{I_{Den}}{I_{Den}} \times \frac{N}{2} \frac{e_{D_alw}}{I_{Num}} \right| + \left| \frac{e_{D_alw}}{I_{Den}} \right| \end{aligned} \quad (4-49)$$

Get:

$$e_{COG}\% = \frac{N/2}{I_{Num}/I_{Den}} \times e_{Den}\% + e_{Den}\% \quad (4-50)$$

Thus, it is possible to control the total COG error limit using the denominator error boundaries only. The term I_{Num}/I_{Den} can be approximated for $\widetilde{COG} = N/1.7$ and for $\widetilde{COG} = N/2.4$, which are the average values of the right and left switching points, respectively, as in the following cases:

$$1- \text{ For right switching points: } e_{\widetilde{COG}_R}\% \approx 1.85 \times e_{Den_R}\% \quad (4-51)$$

$$2- \text{ For left switching points: } e_{\widetilde{COG}_L}\% \approx 2.2 \times e_{Den_L}\% \quad (4-52)$$

The different absolute denominator errors, which can be considered during the COG calculations, are:

$$\text{The general form: } e_{Den} = \frac{|e_{COG} \times I_{Den}|}{0.5 \times N + COG} \quad (4-53)$$

$$\text{For the right switching point: } e_{Den_R} \approx \frac{|e_{\widetilde{COG}_R} \times I_{Den_R}|}{1.1 \times N} \quad (4-54)$$

$$\text{For the left switching point: } e_{Den_L} \approx \frac{|e_{\widetilde{COG}_L} \times I_{Den_L}|}{0.9 \times N} \quad (4-55)$$

$$\text{For } \widetilde{COG} \approx N/2 : e_{Den} = \frac{|e_{COG} \times I_{Den}|}{N} \quad (4-56)$$

If the $|e_{COG}|$ is set to be half of one-discrete element, then the minimum value of $|e_{Den}|$ is resulting if the COG is at its maximum value, which equals to N for discretised outputs.

$$\text{Min } e_{Den} = \frac{|I_{Den}|}{3N} \quad (4-57)$$

The maximum possible value of the error $|e_{Den}|$ can result if the COG is minimum, which it is zero in the case of digital discretised outputs, to get:

$$\text{Max } e_{Den} = \frac{|I_{Den}|}{N} \quad (4-58)$$

This minimum error limit, which is one-third of the maximum error limit, has to be used to assure that COG error is going to be within required limit whatever the COG result is. Thus, to achieve absolute output error smaller than one discrete element, an absolute

error of half discrete element has to be used in the evaluation of the left and right switching points, as defined in equation (4-42) - page 80. However, to get the absolute value of $|e_{Den}|$, the value I_{Den} has to be approximated for the first computation cycle of the iterative KM type reduction family. The average membership level of the set $\frac{N}{2}$ can be used for convex shaped fuzzy sets that have N discrete elements on their vertical and horizontal axis, as:

$$I_{Den0} \approx N_{Horizontal} \times \frac{(\mu_{min} + \mu_{max})}{2} \approx \frac{N_{Horizontal} \times N_{Vertical}}{2} \quad (4-59)$$

Assuming for simplicity that $N_{Horizontal} = N_{Vertical}$ and considering the COG is on its maximum value of (N) then substituting this in equation (4-53) to get:

$$[e_{Den}]_0 \approx |e_{OutF2}| \times \frac{N}{3} \quad (4-60)$$

The evaluation of e_{Den} within the KM's iterations after the first cycle will be more accurate, than the initial guess, because the previous integration's actual result will be used in the new error evaluation.

$$[e_{Den}]_i = \frac{e_{OutF2} \times |I_{Den}|_{(i-1)}}{0.5 \times N + COG} \quad , \quad for \ i \geq 1 \quad (4-61)$$

The $|I_{Den}|_{(i-1)}$ represents the latest result of the numerator integration in the iterative KM type reductions. The minimum error, which can result if $COG \approx N$, is as shown below:

$$[e_{Den}]_i = \frac{e_{OutF2}}{1.5 \times N} |I_{Den}|_{(i-1)} \quad , \quad for \ i \geq 1 \quad (4-62)$$

Substituting the initial error value from equation (4-60) into error relation (4-30), page 78, to get the simple initial error limit as follows:

$$|n(\mu_1 - \mu_0) - (\mu_n - \mu_0)| \leq |2e_{OutF2}|_0 \quad (4-63)$$

Using equation (4-62) in the next evaluation cycles, to get:

$$|n(\mu_1 - \mu_0) - (\mu_n - \mu_0)|_i \leq \frac{6}{N} [e_{Den}]_{(i-1)} = 4e_{OutF2} \frac{|I_{Den}|_{(i-1)}}{N^2} \quad (4-64)$$

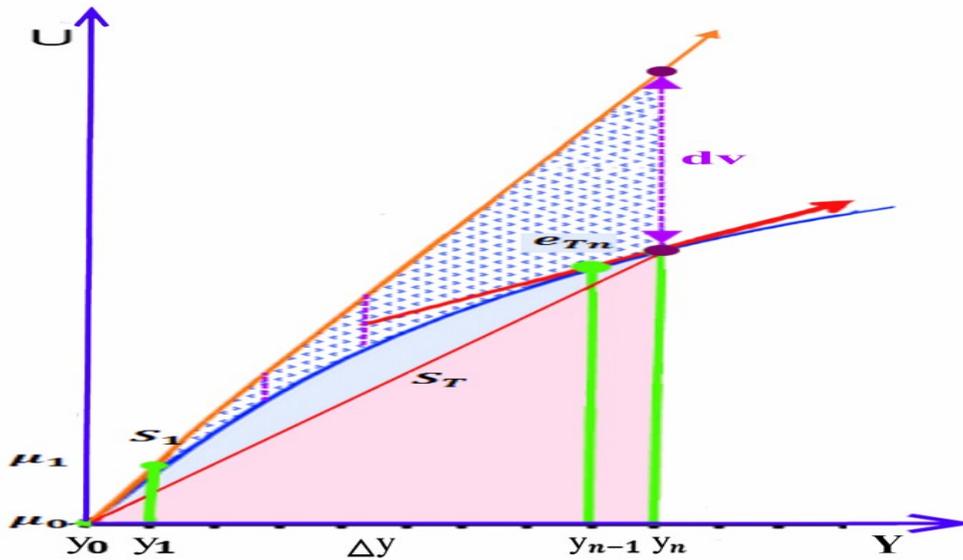


Figure 4-3: Estimating the possible error limit of the trapezoidal rule by monitoring the difference between a straight line and the curved section.

The value $|n(\mu_1 - \mu_0) - (\mu_n - \mu_0)|$ represents the vertical difference (dv) between the curved fuzzy set and the straight line that starts from the first point of the integration period and has a slope equal to the initial slope of the curve, as shown in Figure 4-3.

4.6.1. The COG's Error Dependency

The COG error, which is controlled by relation (4-62) is dependent on the denominator result $|I_{Den}|$ of the previous cycle. Thus, if a previous cycle result is smaller than the result of the running calculation then the fuzzy controller's output error is going to be smaller than the required error limit, which it is acceptable.

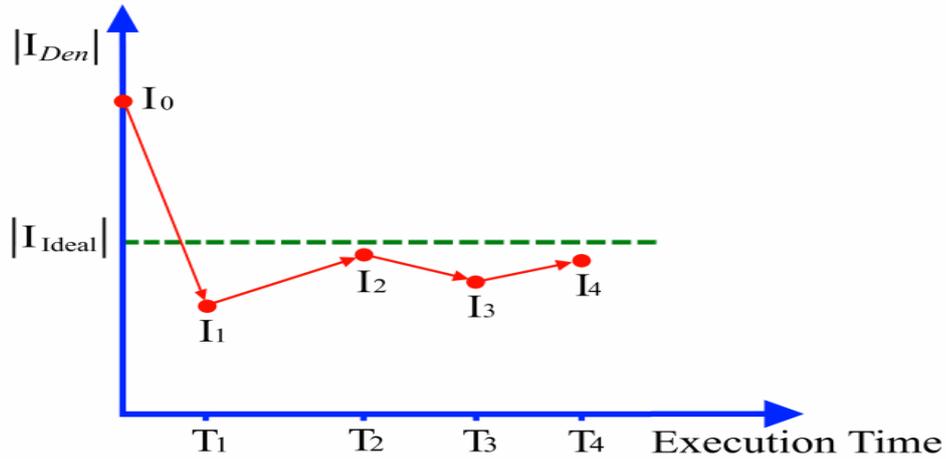


Figure 4-4: Integration result fluctuation because of error dependency.

However, if the current integration result is smaller than the previous integration result then the error of this integration can exceed the required system limit. If this is the case, then it can be fixed by repeating the integration using the last integration result, which is generated due to using the previous integration result. Analytically, it can be seen as in Figure 4-4 that using only the previous integration result to define the calculation allowance will cause results fluctuation. As an example, consider the old integration result $|I_0|$ is big and the 2nd integration result $|I_{ideal}|$ is small. Thus, in the first evaluation cycle, the allowance is high, the execution time T_1 is short, and the last integration result $|I_1|$ is smaller than $|I_{ideal}|$. This smallest integration result is generated because the fuzzy sets have convex shapes, thus any approximation to their curved shapes using straight segments will cut away part of the area under the curve and gives smaller total area of the underside of the curves. Because of that, all the integration results $I_1, I_2, I_3,$ and I_4 are smaller than the ideal integration result $|I_{ideal}|$. But, as can be seen in Figure 4-4, the best result is I_2 , because its error falls within the required controller's error and it takes a short execution time to be calculated. Any further cycles would consume a longer execution time and increase the integration error.

4.7. Correct Interval Length Search

The trapezoidal rule error, which is defined by equations (4-15) and (4-30), is directly proportioned to the interval length of the integration, in the case of pure quadratic functions. This property enables the use of successive binary search technique to find the correct intervals that maintain the trapezoidal rule error limit. The successive binary search does approach the required criteria using a coarse accuracy at the beginning, which then to become finer and finer, step after step, as shown Figure 4-5. The first step size in a search domain contains (N) discrete elements will be ($N/2$). The successive steps will be halved until reaching a step size equal to one discrete level at the end of the search process. However, here, the search interval length is non-radix-2. Therefore, the step size also is non-radix-2 and dividing it by 2 would require rounding either to floor or ceiling. In this case, it is possible to round the step size to floor thus to get final interval length smaller than the ideal case (Ogawa et al. 2011; Zeloufi et al. 2015). Such smaller intervals will certainly have an error limit falls within the required boundaries. Also, it is necessary to divide the domain of the output fuzzy sets at the cross-section points between the adjacent sets and at the peak points of the sets, as shown in Figure 4-6. This will guarantee to get in each subsection a quadratic shape that is going to stay quadratic during every possible fuzzy firing level combination. Re-sectioning has to be performed again if the shapes of the output sets are changed by some tuning process.

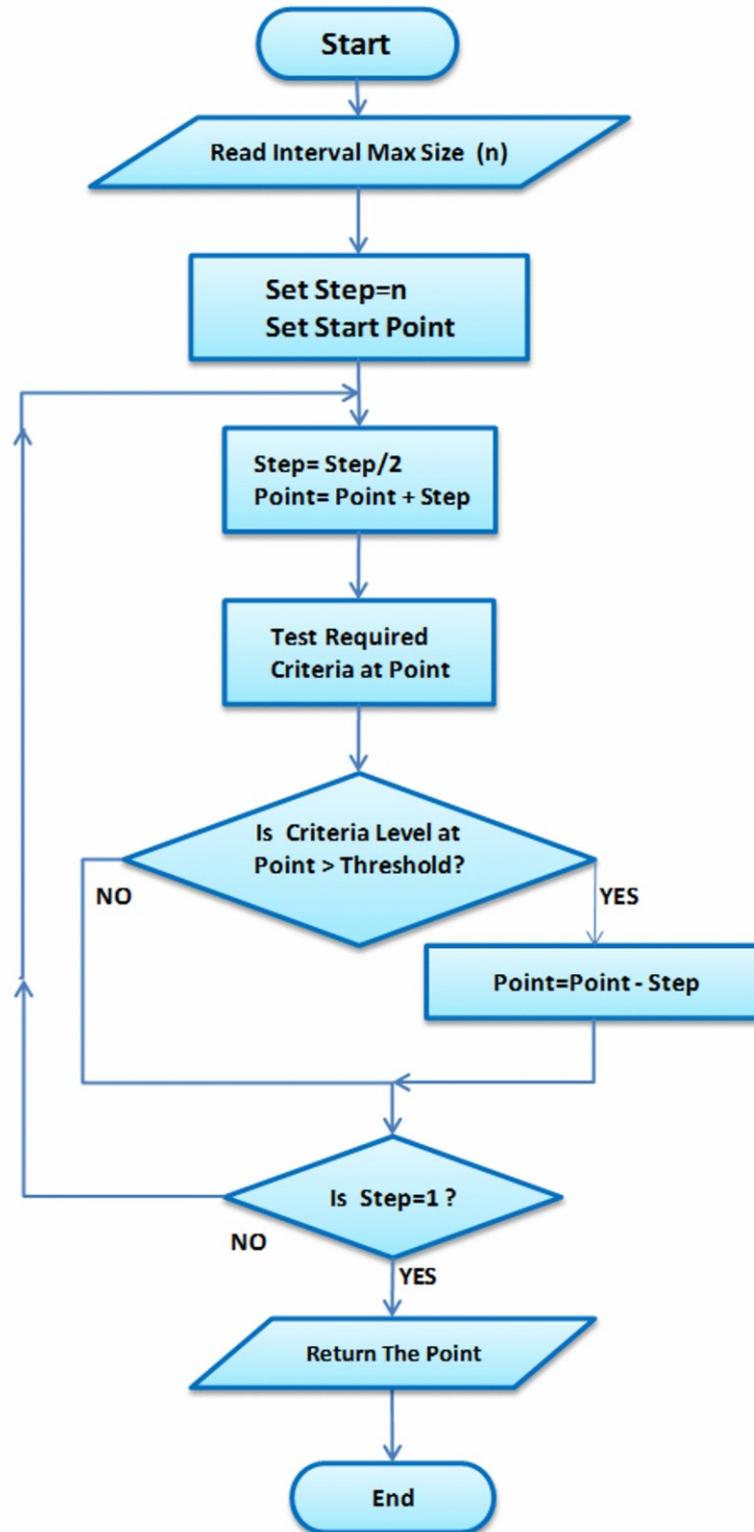


Figure 4-5: Traditional successive binary search.

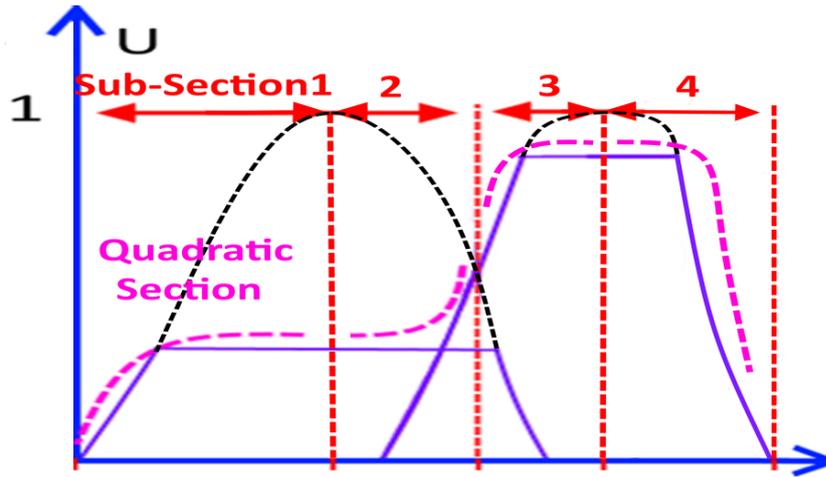


Figure 4-6: Output fuzzy sets sub-sectioning example.

4.8. COG Using Adaptive Trapezoidal

The type reduction and defuzzification operations that can generate smooth output are mostly performed through a weighting average operation, which is the COG calculation. Any COG calculation contains two integrations; one is for the denominator and the other for the numerator. The denominator integration can be performed using trapezoidal integration over N discrete points as follows:

$$I_{Den} = \int_{y_0}^{y_N} f(y) dy \approx \sum_N (y_n - y_{n0}) \frac{(\mu_n + \mu_{n0})}{2} \quad (4-65)$$

The numerator integrations can be performed as:

$$I_{Num} = \int_{y_0}^{y_N} y \cdot f(y) dy \approx \sum_N (y_n - y_{n0}) \frac{(y_n \mu_n + y_{n0} \mu_{n0})}{2} \quad (4-66)$$

Here, y_{n0} denotes the local segment start point and y_n defines its end. The symbols μ_{n0} and μ_n define the fuzzy membership level at the segment ends. The lengths of these different integration segments affect the total COG calculation error. Controlling the segments, lengths can be achieved using relation (4-64) to monitor the error of the denominator integration only then executing the two COG integrations. However, it is important to analyse what is going to happen if an approximation process is merged with a KM iterative type reduction family procedure. These iterations have the well-known iterative form $y_{n+1} = f(y_n)$, where $f(y_n)$ is replaced by $\text{COG}(y_n)$ where y_n is the movable switching point that joins the left and the right parts of the fuzzy set to be reduced, see paragraph 2.4.1 and Figure 2-1. The iterations are going to settle or end at what is known as the fixed point (FP). This will happen when the left equation part, y_{n+1} , which can be described as an independent equation, defines a straight line, and has a slope of one, as shown in Figure 4-7. This equation is going to equate the right part $f(y_n)$ at the FP. However, this would require that the function $f(y)$ has an absolute slope less than or equal to one, and has a sufficient extend around the FP. That extend is important to including the initial guessed point (Burden and Faires 2011). However, all the possible progressing patterns to reach the FP are dependent on the function slope between the initial guess and the FP.

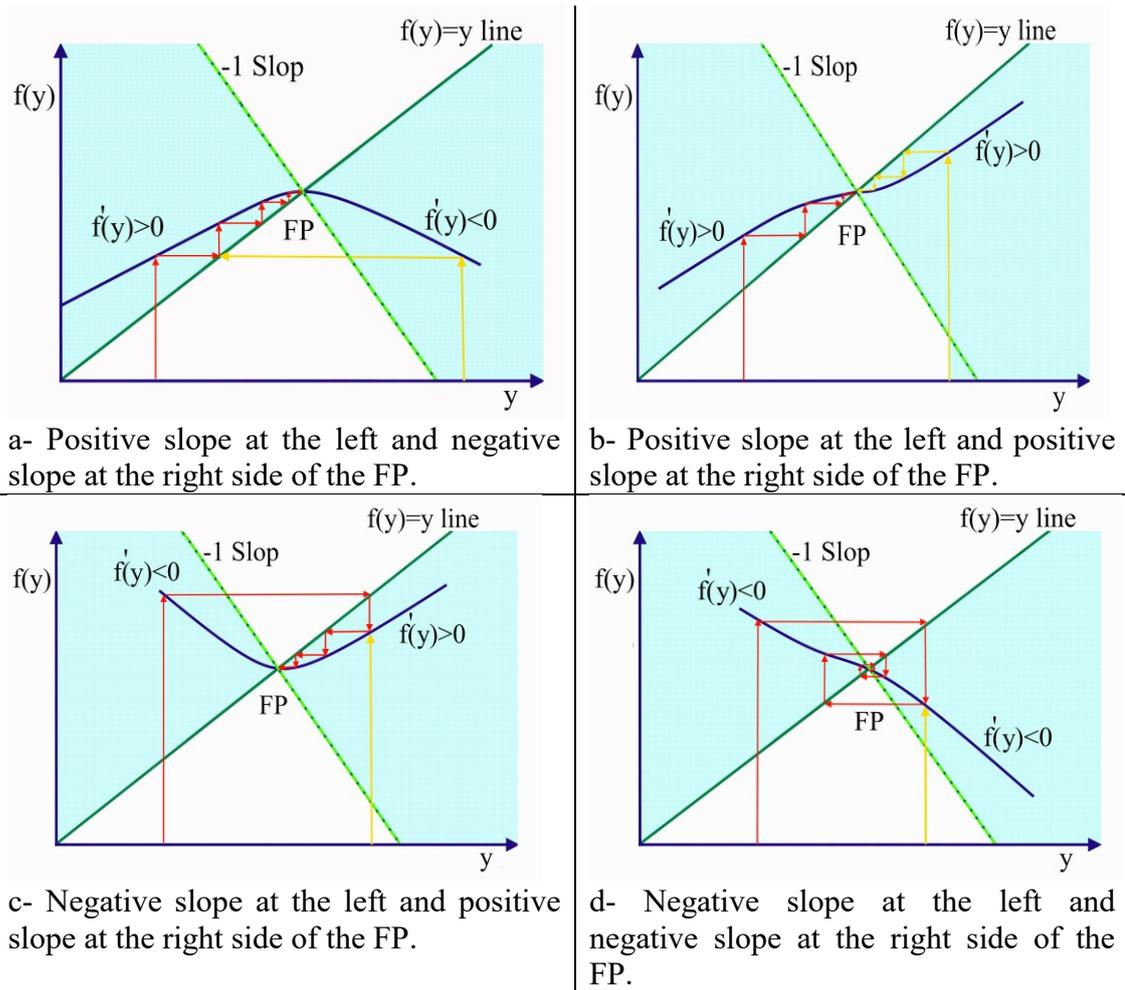


Figure 4-7 : The different progressing cases of the FP iterative procedure showing all the cases in a, b, c, and d.

But, the condition of having an absolute slope smaller than or equal to one will generate some complexities when any approximation is performed on the function $f(y)$. The complexity arises because approximation will add some error ε to the initial function points making it in the form $f(y) \pm \varepsilon$. This means that the initial smooth function is converted to what is known as a partially discontinuous function (Cromme 1997), where any two adjacent points can be within a circle of a diameter \mathbf{D} . This \mathbf{D} defines the function discontinuity level, which in our case equals to $\mathbf{D} = 2\varepsilon$ because the approximation error may be added to a point and may be subtracted from its adjacent point, thus creating a gap of double the error size. This possible discontinuity can create many small segments around the FP with slopes equal or greater than one. In such cases, the converging condition of having an absolute slope smaller than one is violated and the iterative KM type reduction procedures is going to stuck in an infinite loop on one of

these segments, as shown in Figure 4-8. Such an infinite loop can be terminated by setting a maximum iterations limit or by setting the stop condition of the iteration to be sufficiently large, as $(2\varepsilon + 1)$ (Jean-Jacques Herings et al. 2008; Cromme 1997), noting that this approximation error is measured as a multiplicand of the function discretisation level. That first solution can waste some of the system processing power and generate a high level of delay in the type reduction stage. Taking in account that control loops have to be repeated at short time slices, not longer than one tenth of the controlled plant time constant, then that solution is not preferred. The second solution can give a very short execution time, which is good, but this sacrifices accuracy, reaching to $(2\varepsilon + 1)$.

This problem can be solved by averaging of the last two points, which are generated by the last two FP iterations, if and only if a looping case is occurring. This solution smooths the approximated function and can give a result with a smaller error. In addition, this process can accelerate converging to FP even for smooth function, like case (d) in Figure 4-7 where it can easily be seen, that averaging any pair of points, in the case of looping, gives a result closer to the FP than any of the points being used in the average. Thus, before doing the average, it is essential to detect the looping case. This detecting is possible by examining the last three of the FP iterations, y_0, y_1, y_2 , where y_0 is the newest generated point, y_1 is the old generated point, and y_2 is the oldest point. From these points, two progressing steps (PS) are calculated describing the last two cases, PS_0 , and PS_1 , as follows:

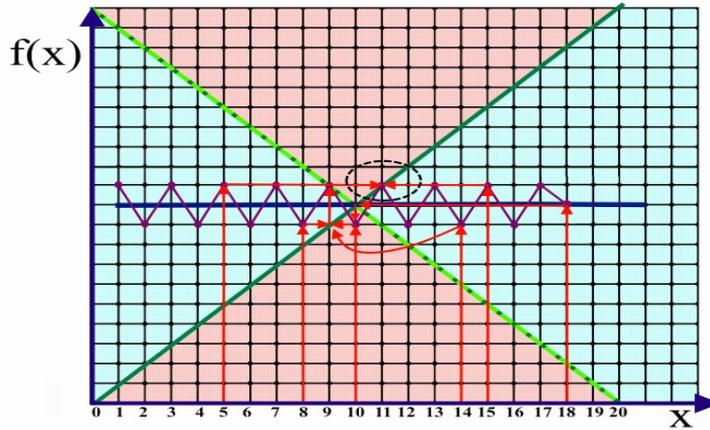
$$PS_0 = y_0 - y_1 \quad (4-67)$$

$$PS_1 = y_1 - y_2 \quad (4-68)$$

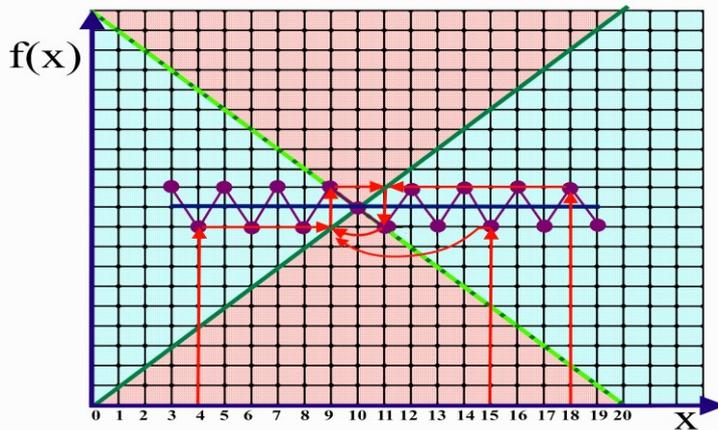
A simple inspection to all the possible cases in Figure 4-7 and Figure 4-8 of multiplying any two successive PSs would show that a negative result could only be generated if a looping case is happening. Thus, this negative can be used to indicate a looping state occurrence.

$$IF: \quad (PS_0 * PS_1) < 0 \quad Then \textit{looping flag is ON} \quad (4-69)$$

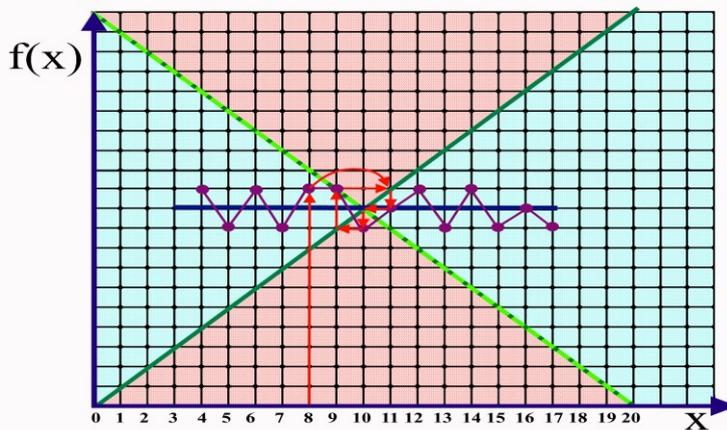
However, before starting to use this indicator, three previous points have to be generated. Another abnormal looping state can be seen in Figure 4-8- case (c), which is occurring because of the approximation noise. In such case, the PS is increasing and decreasing instead of the normal behaviour of continually decreasing till it reaches the FP. Our proposed remedy for such abnormal looping is to take an average of all the points in the loop and terminate the iterations. This can give a better result because averaging all of the points in a loop gives one central point, which is very close to the ideal FP.



a- Case of one-bit noise added to a straight line function causing a shift in the position of the FP.



b- Case of infinite loop around the FP, caused by adding one-bit error to a straight line function.



c- Infinite loop consists of two hops around the FP, caused by adding one-bit error to a straight-line function.

Figure 4-8 : Some possible effects on the progressing pattern of an iterative FP routine are shown in the sub figures a, b, and c, that are caused by adding one bit noise.

Chapter Five: Performance Evaluation

5.1. Evaluating the Adaptive Type Reduction

To evaluate the proposed adaptive trapezoidal integration using successive search technique, for fuzzy type reduction purposes, the output error of some fuzzy controller and the time gain of the new type reduction are to be evaluated for every possible fuzzy firing levels combination while the FT2 controller is running. Two Gaussian output FT2 sets, as shown in Figure 5-1, are suggested to be used during the type reduction evaluations. The Gaussian sets have a well know average features that can be considered as reference values. Thus, any result here can be treated as a reference and interpreted to any other set shape easily. In addition, Gaussian sets are off the most common set types that are being used for speed and direction control in autonomous vehicles. These controllers represent the category of fuzzy systems that will benefit the most from any developed fast type reduction. Using two fuzzy sets per output is usually enough for autonomous fuzzy controllers, as there is a limited nonlinearity in the direction and speed control.

Before commenting on to the different tests, it is required to select a better I_{Den0} value rather than this one being used in equation (4-59) - Page 83, thus achieving higher performance speed at an approximation error not exceeding the designed limits. This initial value can be higher than what is theoretically estimated before using the average area, $\frac{N_{Horizontal} \times N_{Vertical}}{2}$. Using such initial high value means starting with high calculation allowance before reducing it to the final required allowance needed to be by the end of the type reduction iterations. Total time gain is calculated by dividing the execution time of the iterative KM type reduction by the approximated type reduction

time. The execution time is taken as the average at every possible fuzzy level. The hop distance between any two adjacent fuzzy levels is considered ones as 10% and then as 20%. These two proposed transition levels are more than enough, because it is always required to design the repetition rate of any control loop to be about one tenth of the plant time constant. This rate has an analogy to fuzzy step transition size of 10%. Any smaller transition distance between the successive fuzzy levels will give better and shorter type reduction time. Also, because it is proposed, here, to use the result of the previous control cycle, as an initial guess to the current type reduction cycle, then a very close guess to the real possible result will be generated. The different initial value effects are tested using two Gaussian output IFT2 sets with fuzzy level transition size of 20%, which considered an extreme situation. The different results for using fuzzy sets interval widths of 10%, 20%, and 40% are shown in Figure 5-2-(a), (b), and (c) respectively. Those interval widths are selected because they represent most of the practically used cases (Begian 2010). The approximation allowances are in the range of 1, 2, 4 and 8. They are shown on the horizontal axis with their initial integration multipliers, which are in the range of 0.5,1,2,4, and 8. However at Figure 5-3, the same data is collected but for a fuzzy transition step size of 10%. The approximation allowances and the initial integration multipliers are selected to cover big ranges where the best performance point can be easily spotted. From these tests, the best performance time gain is determined to be close to the initial integration multiplier of four. This value is going to be used instead of the theoretical estimated value of 0.5, shown in equation (4-59). This initial value will be updated according to the required calculation allowance and the integration result of the previous type reduction iterations. This means variable calculation allowances is implemented, starting with a big value then ending to a smaller value. In the next section, tests of the time gain and the system error are going to be presented and analysed for the cases of using the I_2 test, which proposed to eliminate any error that may exceed the limits. That I_2 test is previously suggested and analysed in section 4.6.1- page 84.

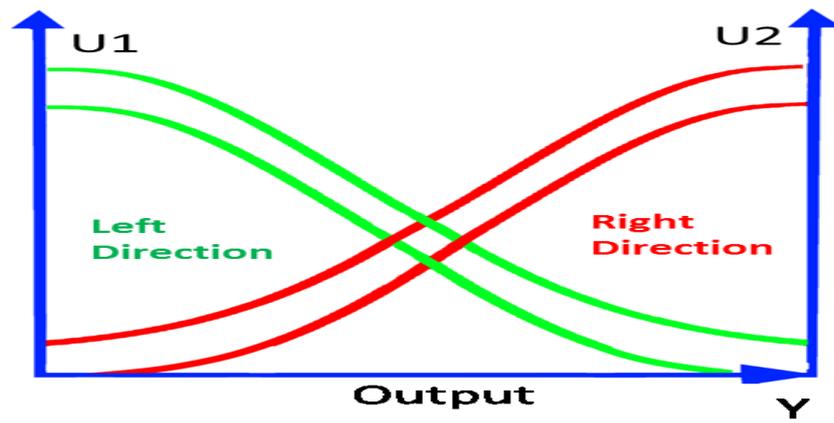
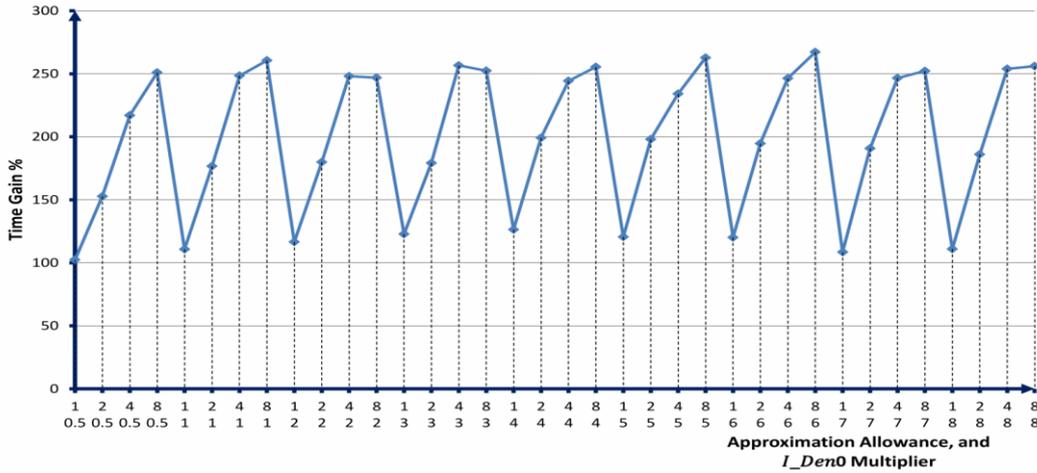
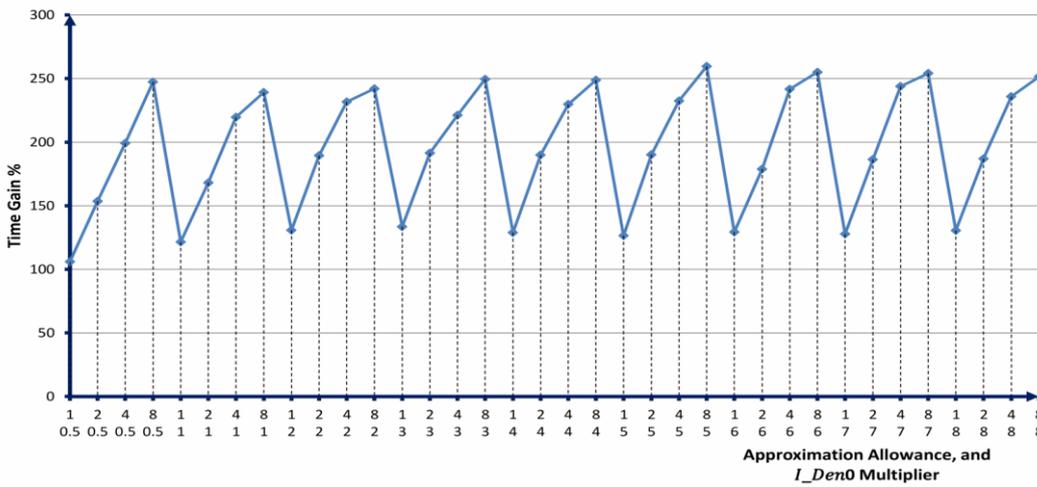


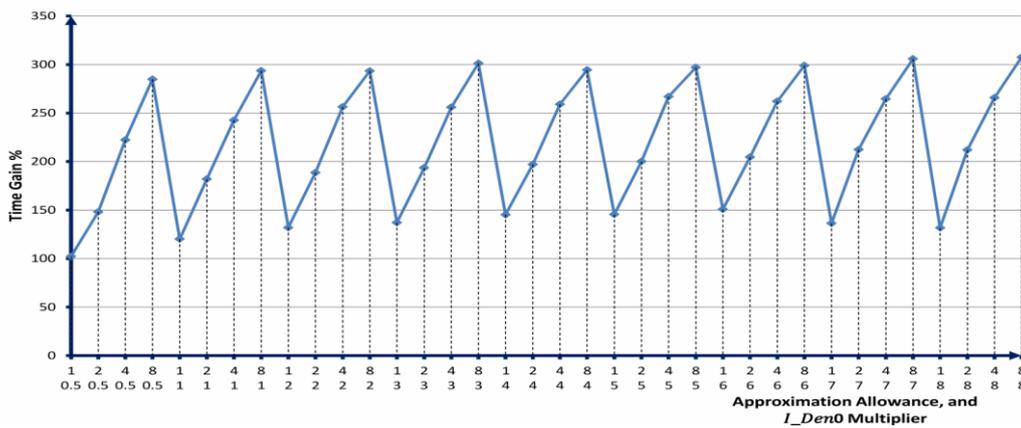
Figure 5-1: Two fuzzy type-2 sets used to evaluate the output error.



a- Average time gain for using IFT2 set width of 10%.

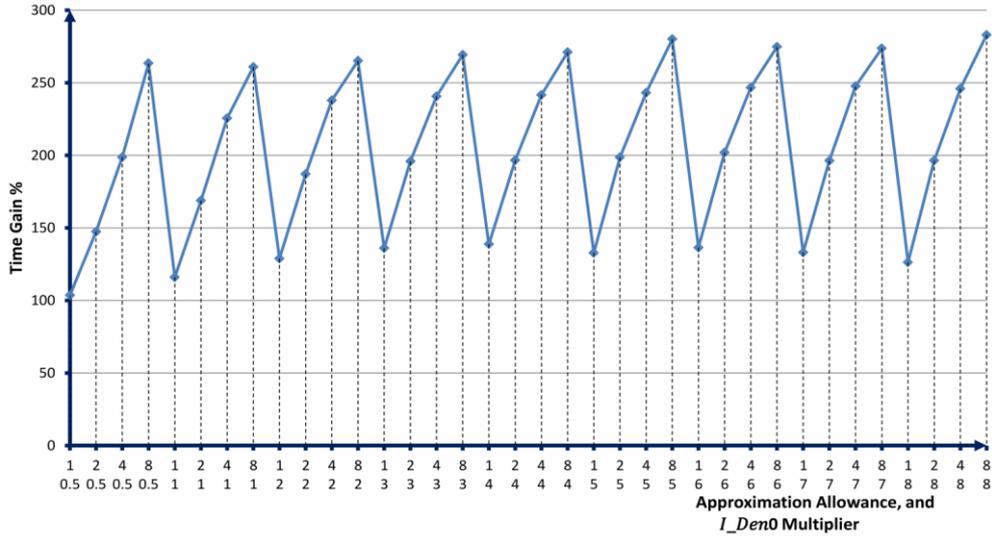


b - Average time gain for using IFT2 set width of 20%.

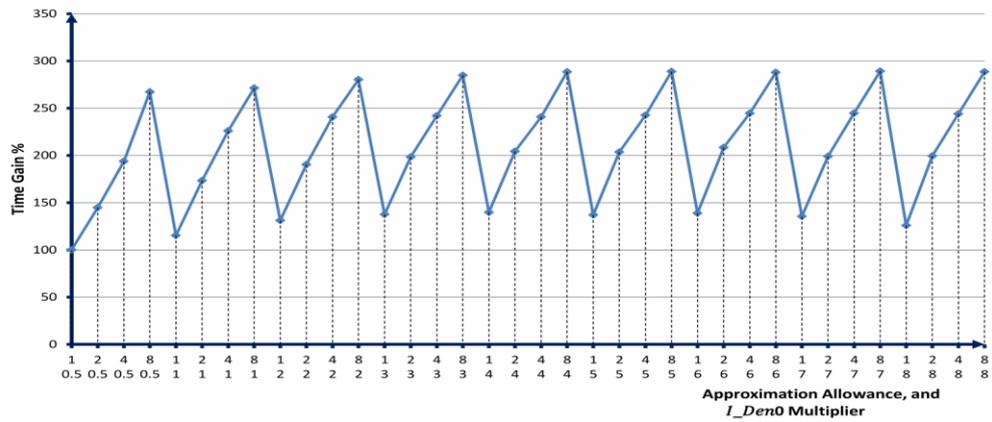


c- Average time gain for using IFT2 set width of 40%.

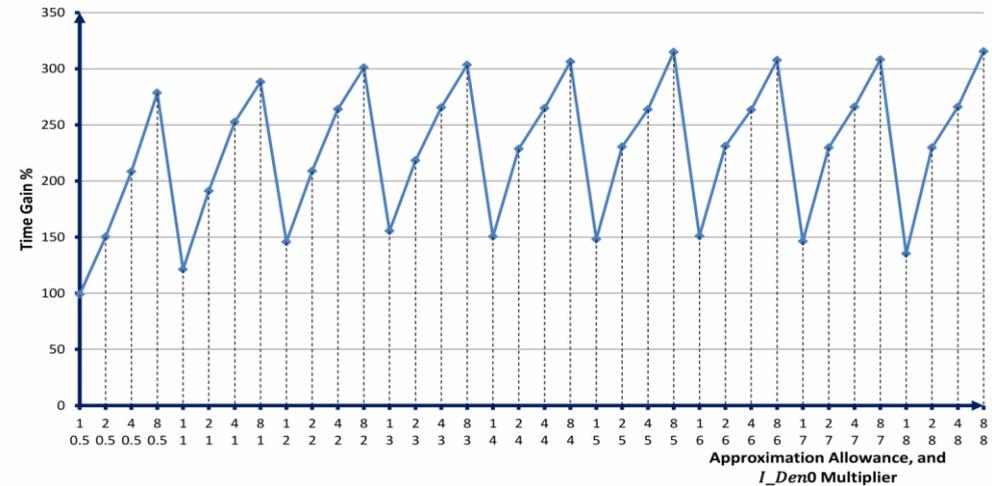
Figure 5-2: Average time gain for the approximated type reduction using fuzzy transition step size of 20% at different approximation allowances and initial integration multipliers for: (a) - IFT2 set width of 10%, (b) - IFT2 set width of 20%, (c) - IFT2 set width of 40%.



a- Average time gain for using IFT2 set width of 10%.



b- Average time gain for using IFT2 set width of 20%.



c- Average time gain for using IFT2 set width of 40%.

Figure 5-3: Average time gain for the approximated type reduction using fuzzy transition step size of 10% at different approximation allowances and initial integration multipliers for: (a) - IFT2 set width of 10%, (b) - IFT2 set width of 20%, (c) - IFT2 set width of 40%.

5.1.1. Using Variable Approximation Allowances

It can be seen from the previous section results, Figure 5-2 and Figure 5-3, that the cost of performing the successive searches is high, as the average time-gain of using approximation allowance of one has reached, in its best cases, to 150% for an initial integration multiplier value of four. What should be achieved is a time gain of 200% for an approximation allowance of one. This is because the approximation allowance of one has an accuracy similar to the accuracy that results from using normal KM at double discretisation size. A possible solution to attain better performance is increasing the calculation allowance during the second iteration of the approximated KM type reduction. This approach would generate a new non-linear approximation throughout the type reduction. This approach has to be tested for its maximum possible calculation allowance at the second iteration to keep the type reduction result at the required error. Performance time-gains are tested for fuzzy transition step size equals to 10%, of the total fuzzy range, using three different fuzzy set widths. The time gain results, shown in Figure 5-4, are for using a Gaussian IFT2 set with interval width of 10%. This time-gain is measured by dividing the execution time of the proposed approximated type reduction by the execution time of the traditional iterative KM type reduction procedure. In these charts, the horizontal line describes two variables, one of them is the required approximation error, and the second is the calculations' allowance used at the second iteration of the approximated type reduction. A wide range of error has been experimented, starting from one discrete element reaching to eight discrete elements. However, the calculation allowances are ranged from one to four. More tests are performed for interval width of 20%, and 40%. These are shown in Figure 8-2, and Figure 8-3, in appendix-A. These results showing high time-gains exceeding 200% at approximation allowance multiplier of two and greater. More tests are performed using greater fuzzy transition size of 20%, which represents an extreme operation condition should be avoided in good controller designs. The results of Figure 8-4, Figure 8-5, and Figure 8-6, in appendix-A, are for the three different interval widths of 10%, 20%, and 40%, using this extreme fuzzy transition step of 20%. Here, the average time-gain has reached to about 200% at approximation allowance multiplier equal to two. The results indicate that the time consumed by this approximated type reduction has a minor dependency on the fuzzy transition step size, but only if the set interval width is in the ranges of 10%.

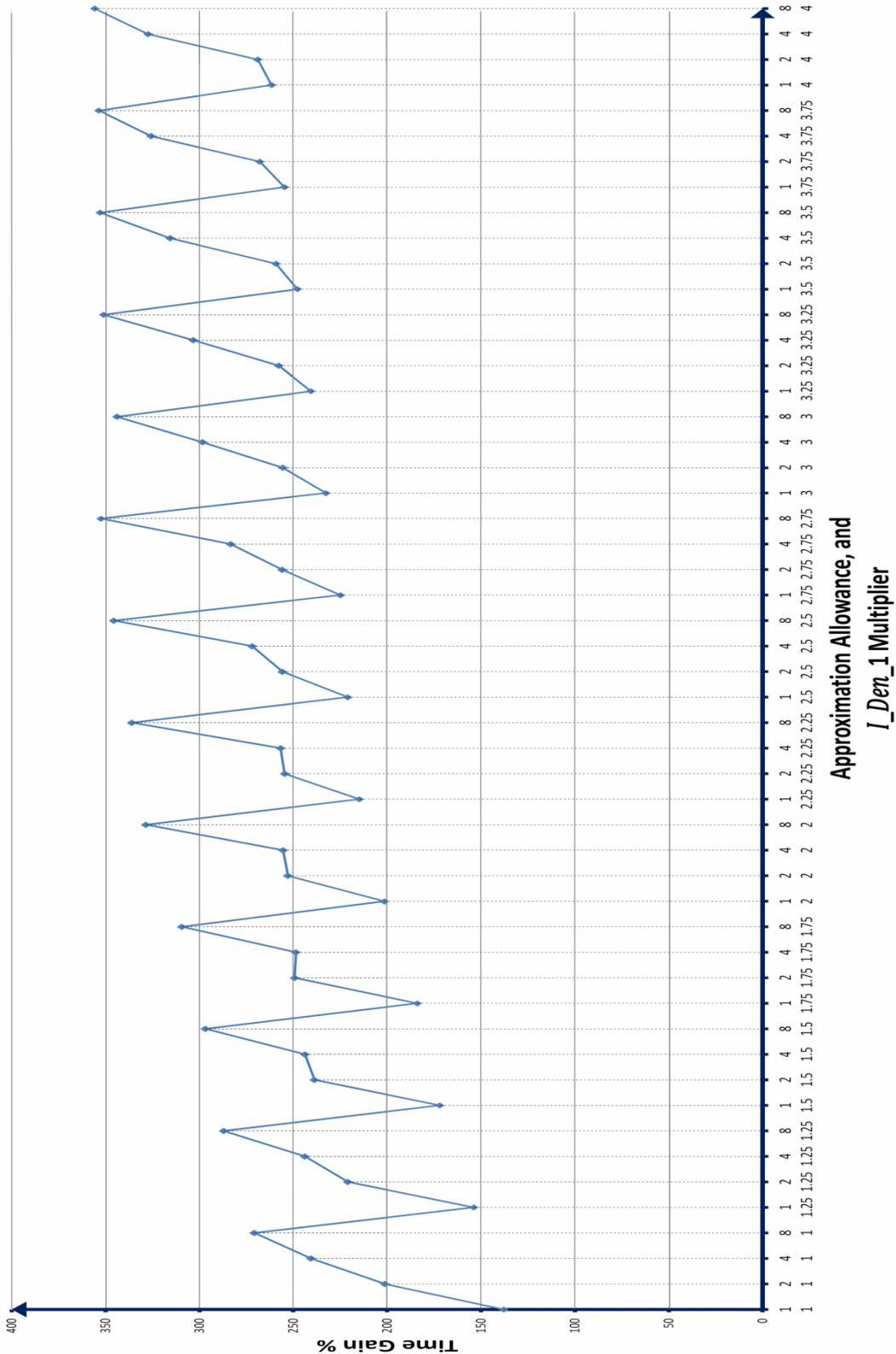


Figure 5-4: Average time-gain for the approximated type reduction using Gaussian IFT2 set having width of 10% and fuzzy transition step size of 10%, at different approximation allowances and various integration multipliers of the second iteration.

5.2. Variable Approximation Allowance Error

The maximum possible error and the error mean are another factors that have to be evaluated for the non-linear calculation allowance that being used during the approximated type reduction. This error is evaluated by calculating the difference between two type reduction units; the first unit has the traditional iterative KM type reduction. The second unit is using our proposed approximated type reduction. Identical Gaussian fuzzy sets, with similar fuzzy firing levels, are used in the type-reduction units to assure that no other factors will influence the errors. The tests are performed at all the possible combinations of two fuzzy firing levels. The maximum error results for using different approximation allowances and different scaling factors during the second approximation cycle for incremental fuzzy transition size of 10% are shown in Figure 5-5 for fuzzy sets have interval width of 10%. The maximum error results for using fuzzy interval sets with widths of 20% and 40% are shown in Figure 8-8, and Figure 8-9 , in appendix-B. Similar tests are performed by using incremental fuzzy transition step size of 20%, which it is double the fuzzy incremental size that is used during the past tests. The results are shown in Figure 5-6 for fuzzy sets have interval width of 10%. However, the results for using fuzzy interval sets widths of 20% and 40% are shown in Figure 8-11, and Figure 8-12, in appendix-B. During those last six tests, incremental fuzzy level transitions started with zero then increased gradually to reach the fuzzy level of one, is used. Practically, fuzzy firing level transitions are increasing and decreasing according to the input variable levels change. Therefore, another set of tests is performed using a decrementing fuzzy firing scheme that starts with one and ends at zero. The maximum resulting errors of these tests, using decremented fuzzy transition step size of 10%, are shown in Figure 5-7 for a fuzzy set with interval width of 10%. The maximum resulting error for using fuzzy sets have interval width of 20% and 40% are shown in Figure 8-14, and Figure 8-15, in appendix-B. Meanwhile, the maximum errors for using 20% decremented fuzzy transition step size are shown in Figure 5-8 for interval widths of 10%. The results of using fuzzy interval sets with 20% and 40% interval size are shown in Figure 8-17, and Figure 8-18, in appendix-B. It can be seen from all those cases that the final error is kept within the required limits for the second cycle allowance multipliers having values smaller or equal to 1.5. That can be seen in the

worst-case condition when the fuzzy transition size is 20% and is decrementing at fuzzy interval width of 10%. This point, which marked by a red circle in Figure 5-8, is considered a boundary value.

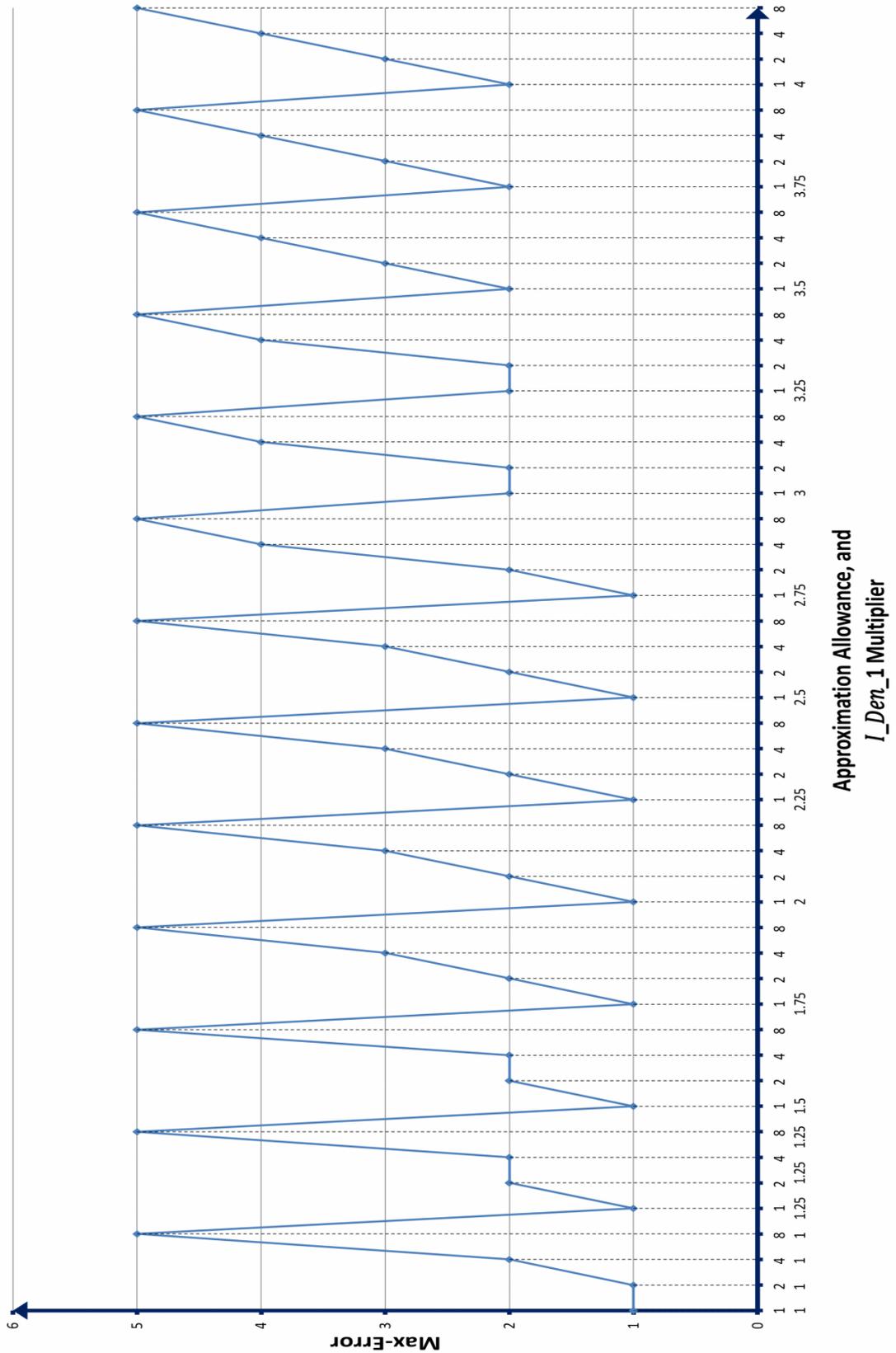


Figure 5-5: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 10% and incremental fuzzy transition step size of 10%, at different approximation allowance multipliers during the second iteration calculations.

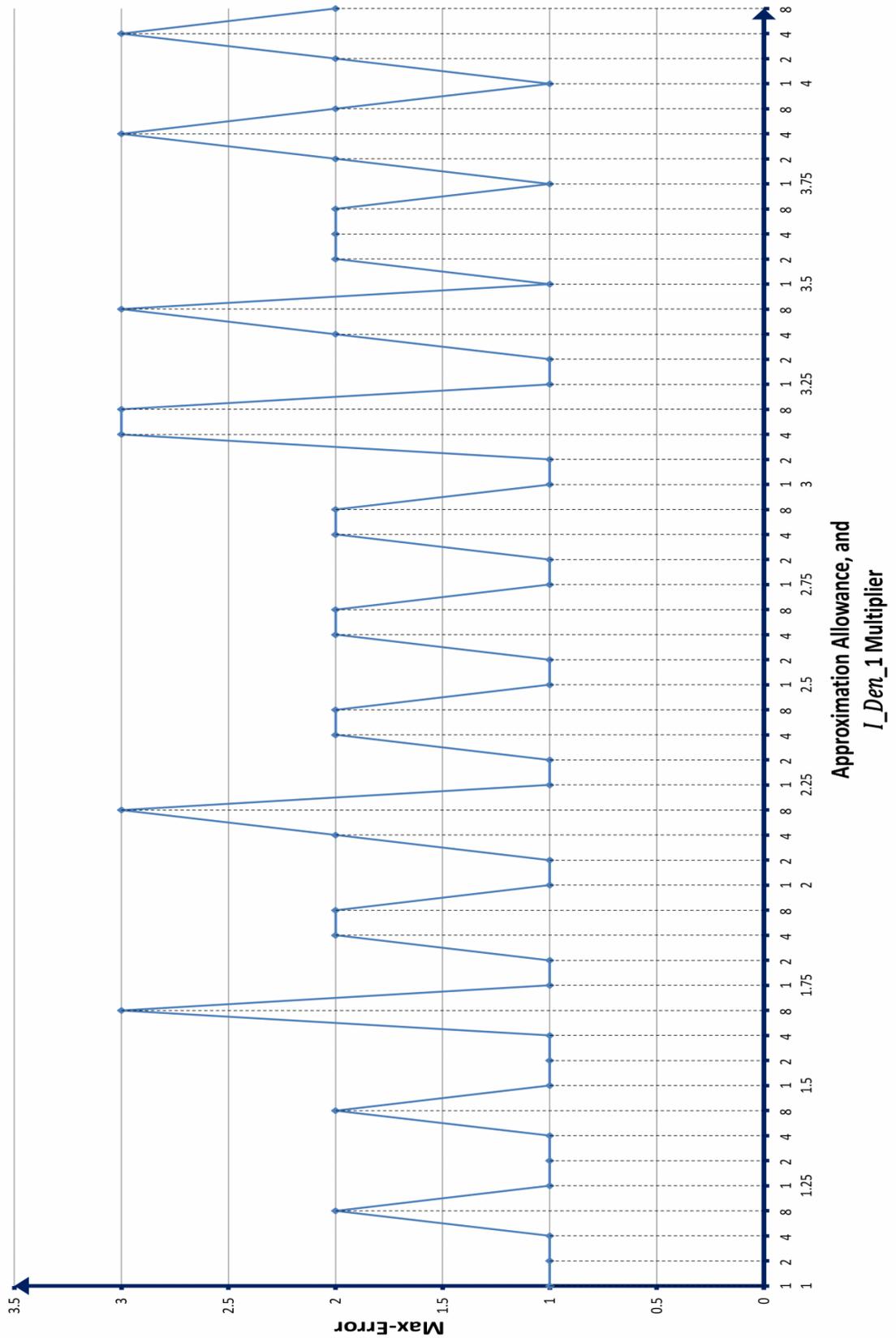


Figure 5-6: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 10% and incremental fuzzy transition step size of 20%, at different approximation allowance multipliers during the second iteration calculations.

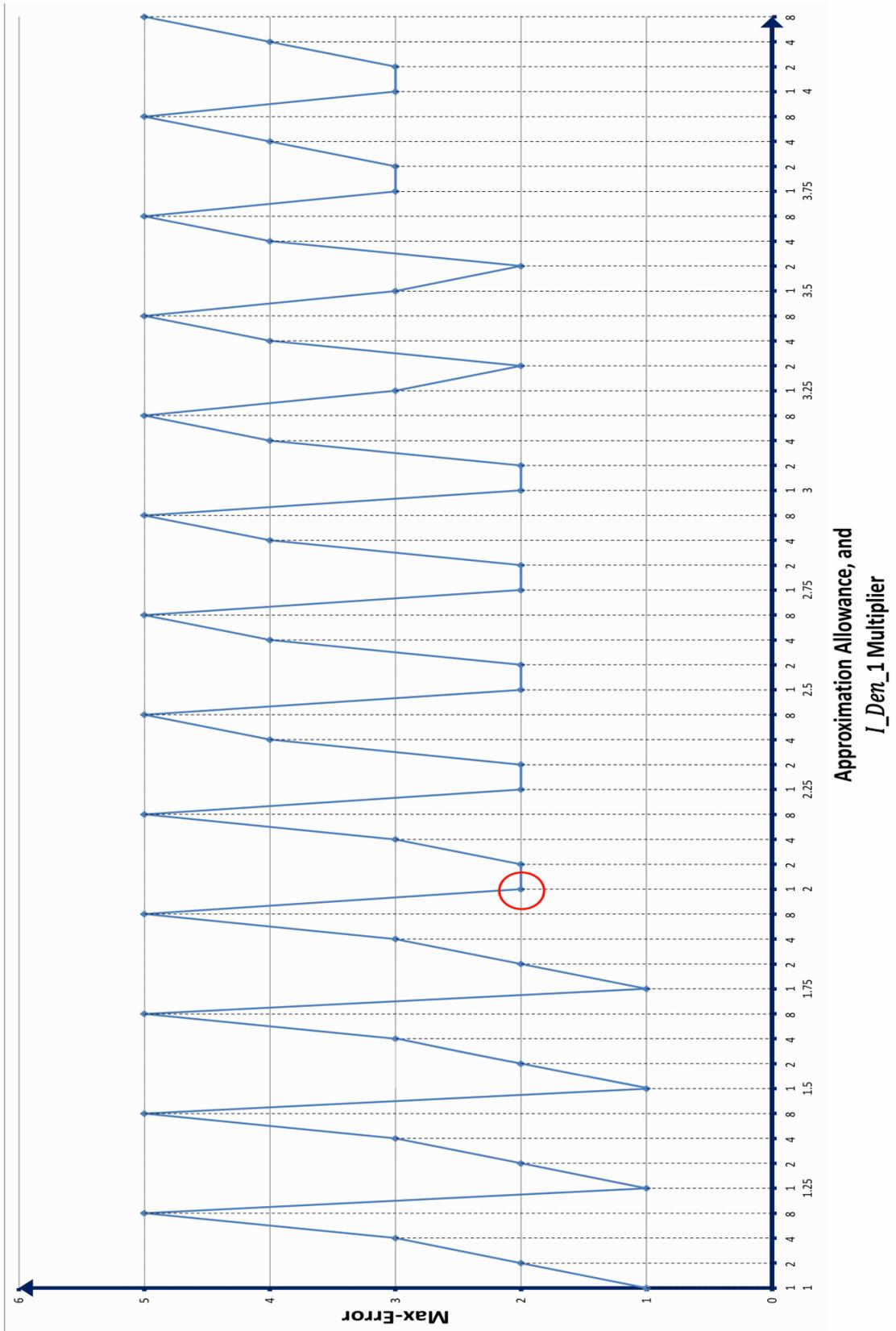


Figure 5-7: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 10% and decrementing fuzzy transition step size of 10%, at different approximation allowance multipliers during the second iteration calculations.

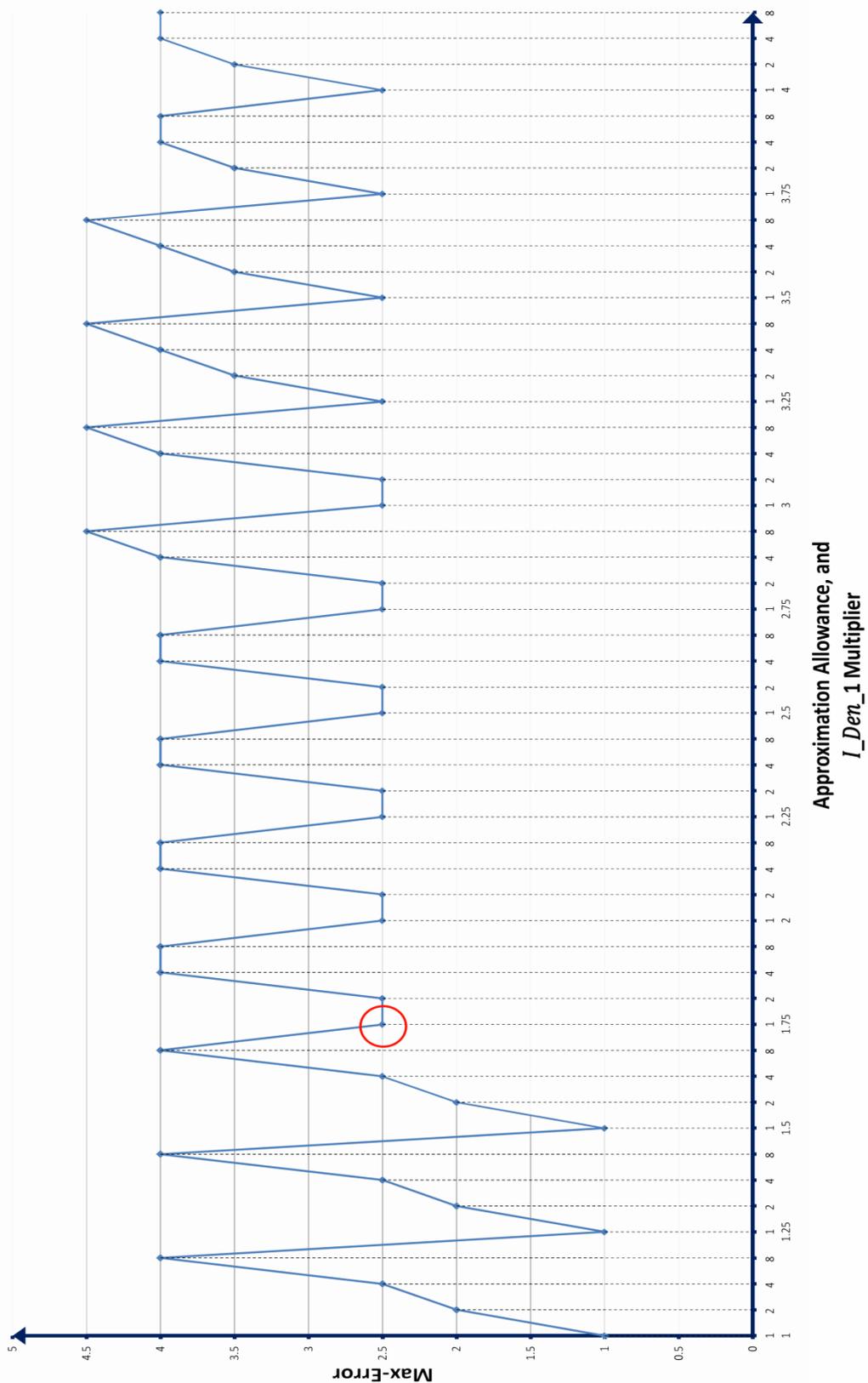


Figure 5-8: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 10% and decrementing fuzzy transition step size of 20%, at different approximation allowance multipliers during the second iteration calculations.

5.3. Performance at Optimal Parameters

Many concepts can be concluded from the two previous sections of 5.1.1 and 5.2. In these sections, all the possible calculation error and the execution time-gain, which are resulted from using high integration multiplier during the first and the second iterations of the approximated KM type reduction, are experimented. Concluding that using an initial integration guess value of four times the horizontal divisions times the vertical divisions can give the best time gain without exceeding any required error limits. Moreover, to achieve faster computations it is possible to use a higher integration multiplier, during the second iteration of the approximated type reduction. These multipliers maintain the required computations error while giving a high time-gain. The best value, that can achieve the best performance while preserving the required errors, has been deduced from sections 5.1.1 and 5.2 to be 1.5. At this value, the total cost of searching the straight segments is slightly higher than the cost-reduction resulting from using straight segments during the type reduction. This is correct for the total cost only. The total cost is evaluated by averaging the reduction time at every possible fuzzy firing combination. A closer look at the different time-gain values during individual fuzzy firing levels using different interval widths with incremental, and decremented fuzzy levels shows that time gain is reaching very high values when the fuzzy firing levels are low. This is reasonable because the fuzzy sets will have longer straight segments that are going to be used during the approximation to achieve that high time-gain. The individual time-gains of using these two ideal values can be seen throughout Figure 5-9 to Figure 5-11 at all the possible fuzzy firing levels, U1 and U2, using a decrementing step size of 10% and interval FT2 sets widths of 10%, 20%, and 40%. The results of using incremental firing level in steps of 10% on IFT2 sets having interval widths of 10%, 20%, and 40%, are shown in Figure 5-12 to Figure 5-14.

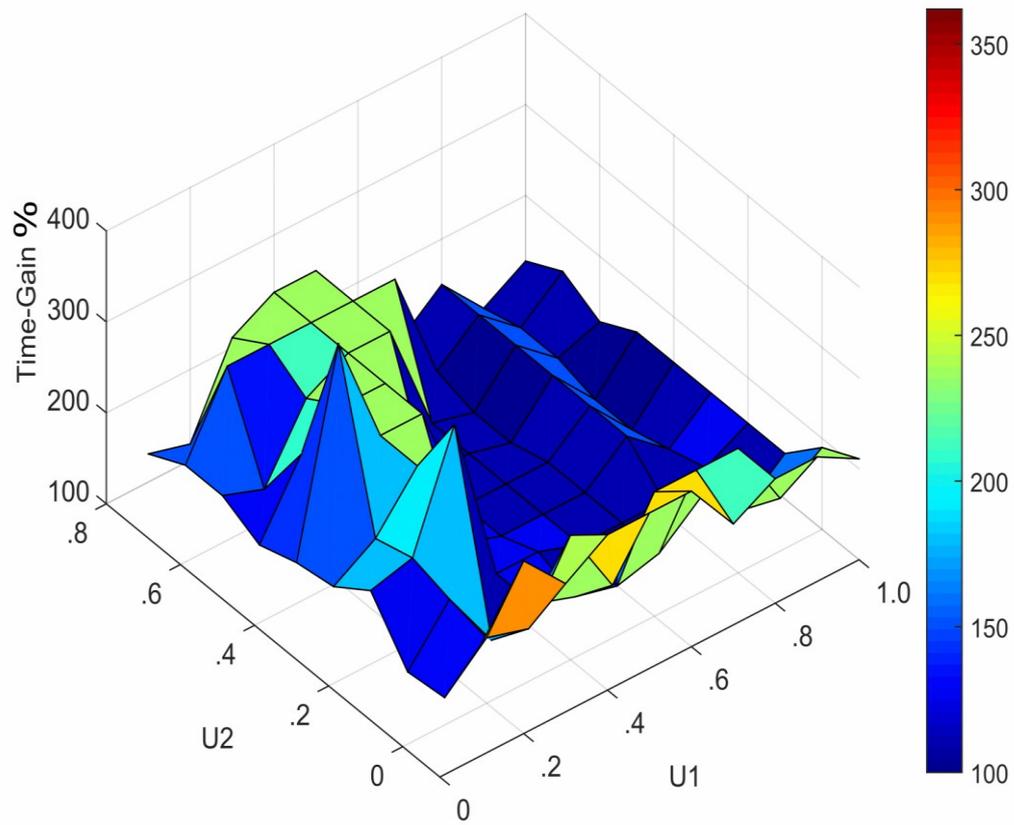


Figure 5-9: Time-Gain for all the possible combinations of two fuzzy firing levels, U1 and U2, transitioning from one to zero in step size of 10%, and acting on two Gaussian's IFT2 sets having 10% interval width.

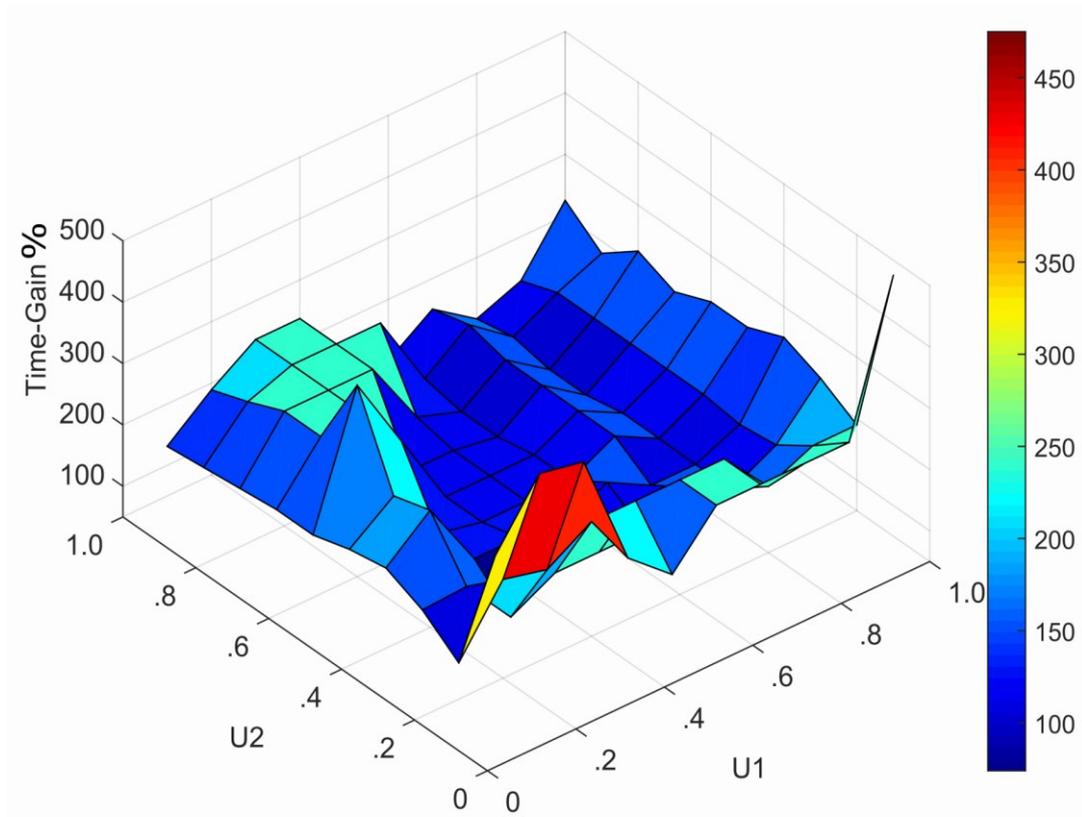


Figure 5-10: Time-Gain for all the possible combinations of two fuzzy firing levels, U1 and U2, transitioning from one to zero in step size of 10%, and acting on two Gaussian's IFT2 sets having 20% interval width.

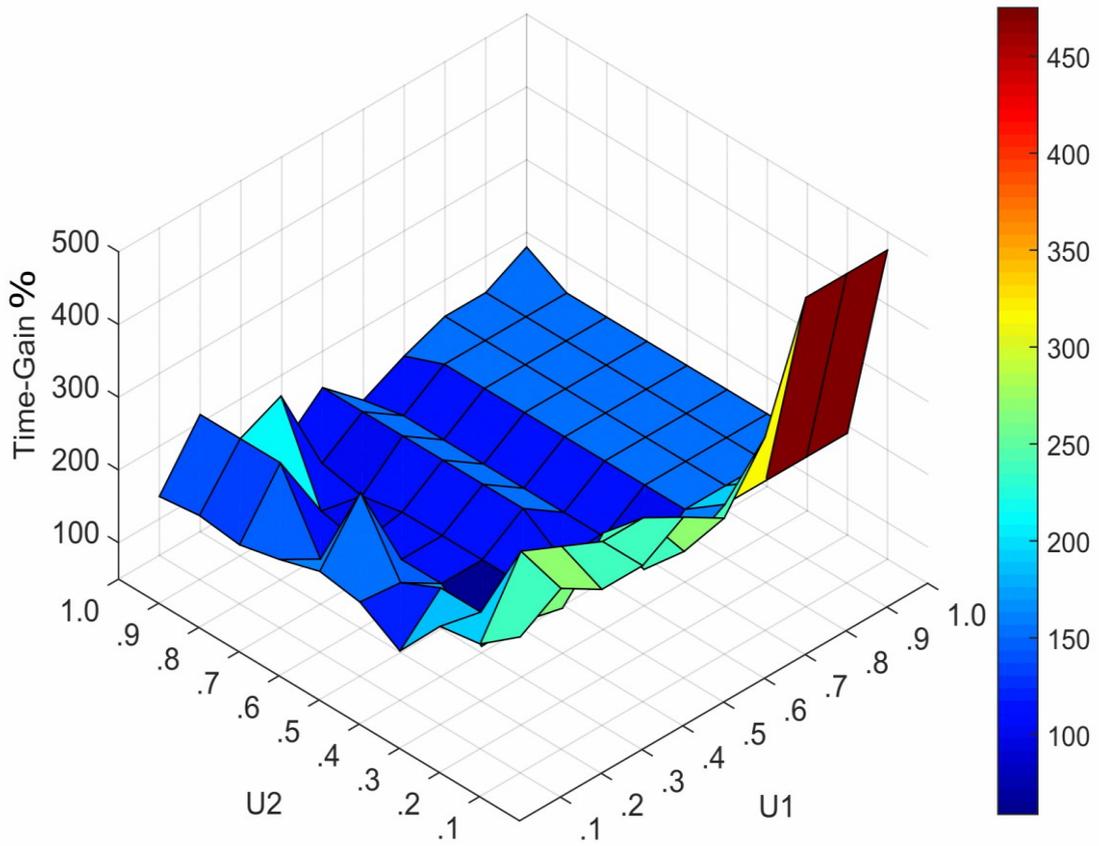


Figure 5-11: Time-Gain for all the possible combinations of two fuzzy firing levels, U1 and U2, transitioning from one to zero in step size of 10%, and acting on two Gaussian's IFT2 sets having 40% interval width.

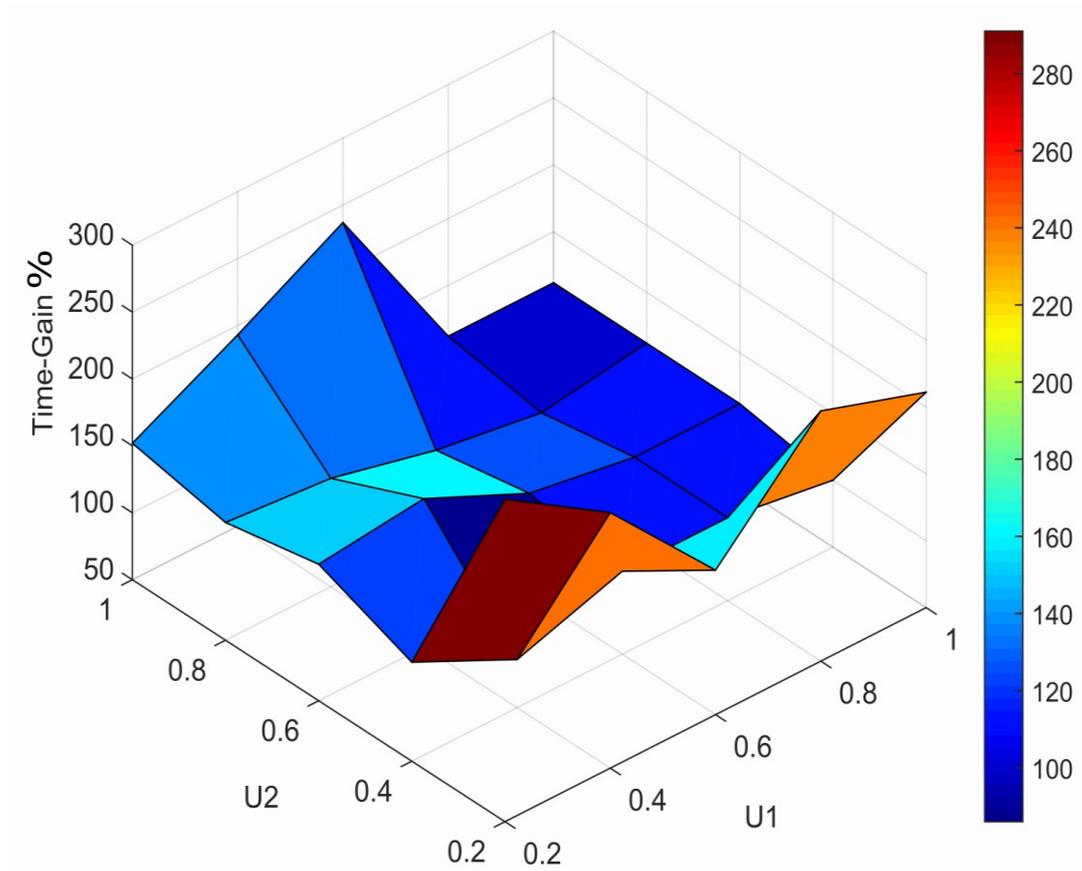


Figure 5-12: Time-Gain for all the possible combinations of two fuzzy firing levels, U1 and U2, transitioning from one to zero in step size of 20%, and acting on two Gaussian's IFT2 sets having 10% interval width.

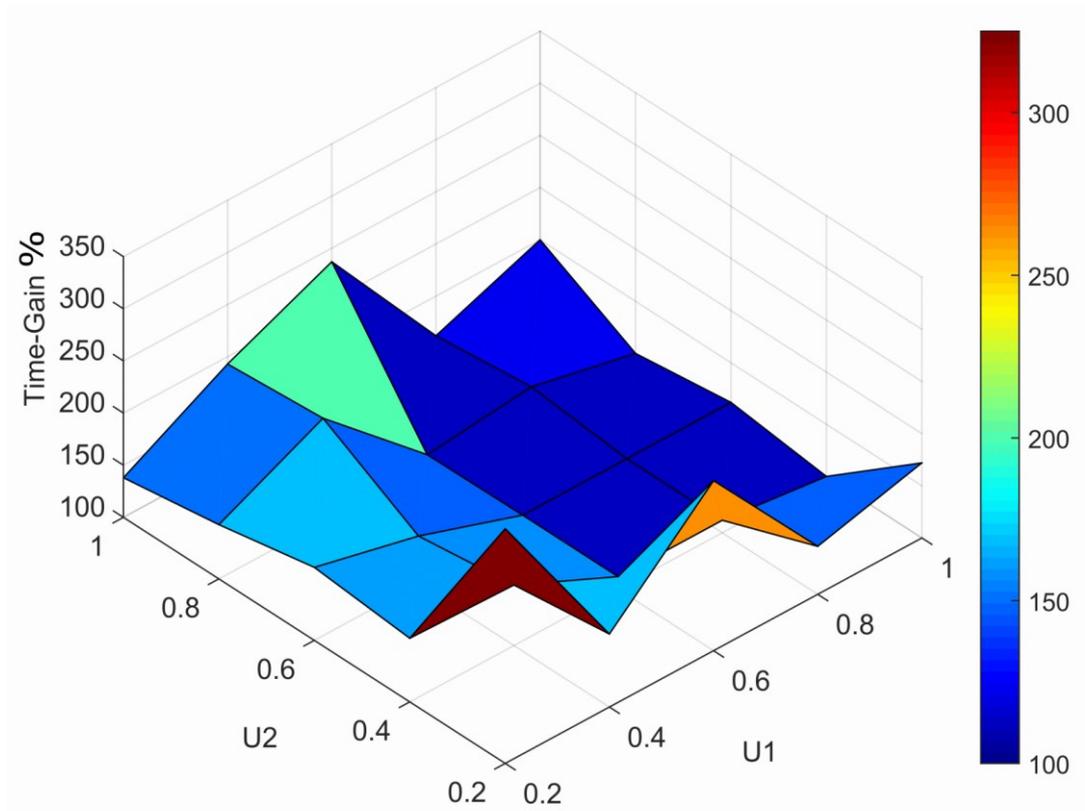


Figure 5-13: Time-Gain for all the possible combinations of two fuzzy firing levels, U1 and U2, transitioning from one to zero in step size of 20%, and acting on two Gaussian's IFT2 sets having 20% interval width.

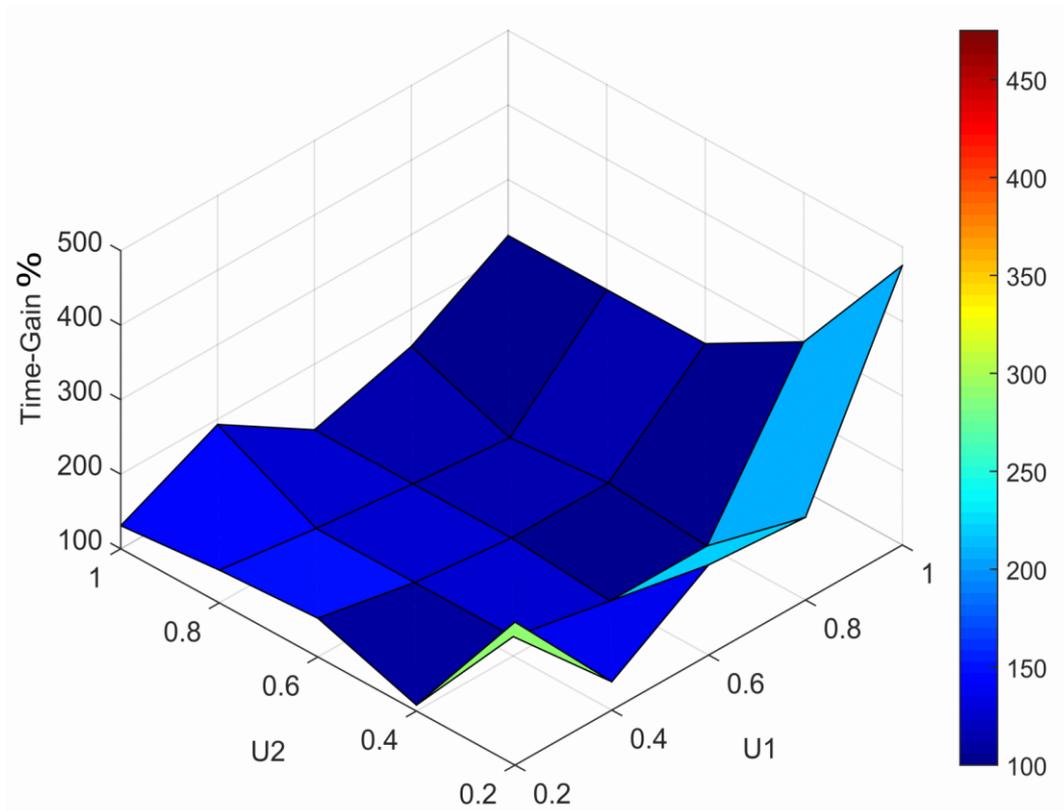


Figure 5-14: Time-Gain for all the possible combinations of two fuzzy firing levels, U1 and U2, transitioning from one to zero in step size of 20%, and acting on two Gaussian's IFT2 sets having 40% interval width.

5.4. Fuzzy Type-2 Autonomous Controller using Adaptive COG

Different behaviours like wall-following, track-following, obstacle-avoidance and goal seeking can be used in autonomous vehicles. The aggregation of all these sub-actions generates one final output. In order to evaluate performance error and execution time-gain of the proposed type reduction, the wall-following behaviour is selected because different wall shapes can be constructed to impose different working conditions throughout the tests. The direction error of the autonomous wall-following controller is measured for the wall structure shown in Figure 5-15, which contains acute, right, obtuse, and reflex-angled corners. Two fuzzy type-2 wall-following controllers have been built, the first uses traditional KM-type reduction with Gaussian output sets, each is discretised to 1000 elements, while the second controller uses our adaptive type reduction, as shown in Figure 5-16. Here the first controller, with a traditional KM-type reduction, is considered as a reference controller. The second controller uses the adaptive COG during the type reduction iterations. Thus, any difference, or error, between the outputs of these two controllers is definitely caused at the adaptive type-reduction stage. In addition, any execution time difference between those two identical controllers is due to the execution difference between their type reduction stages. The computation time-gain for these allowances at interval widths of 10%, 20% 40% are shown in Figure 5-17, as a bar chart form, for a fuzzy direction controller part of an autonomous real and simulated vehicle. It can be seen that time-gain is high, reaching to 500%, in most of the cases, if small interval widths in the range of 10% are used. However, time-gain is small, in the range of 150% when large interval widths, in ranges of 40%, are used. This happens because the uncertainty will be high at these values, thus more iterations will be required to get the left and right switching points. The cost of these extra iterations seems to be higher than what is being gained out of the adaptation. The associated mean absolute errors with these different tests, evaluated using 10000 samples for each, are shown Figure 5-18, where the mean absolute errors are, in all these cases, smaller than one in a thousand. This is considered very good for our fuzzy controller which is using a discretisation level of 1/1000 for its output fuzzy sets. Other tests have been conducted to compare the average time-gain for different sets discretisation levels using an interval set width of 10%, shown in Figure 5-19. It shows that best time-gain is happening at low allowances, before reaching saturation where no real acceleration can be achieved.

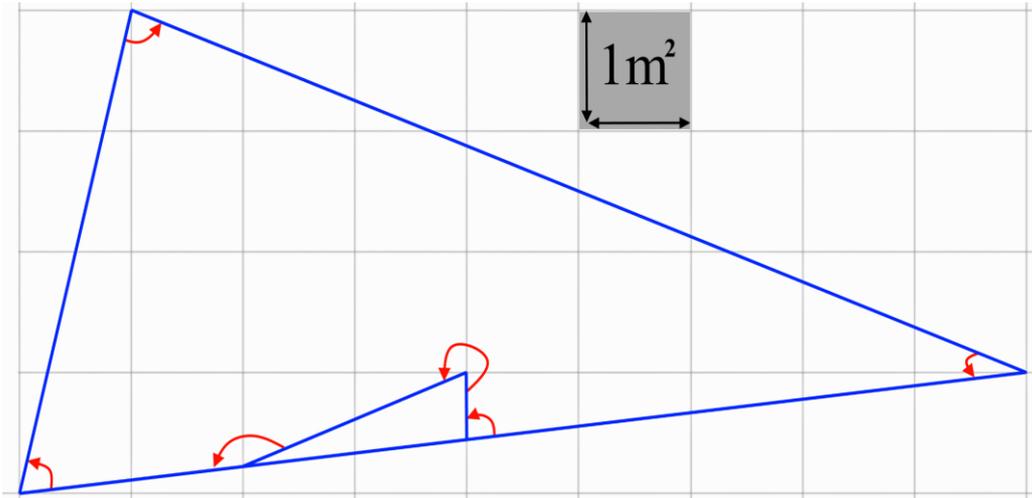


Figure 5-15: Wall following test field arena.

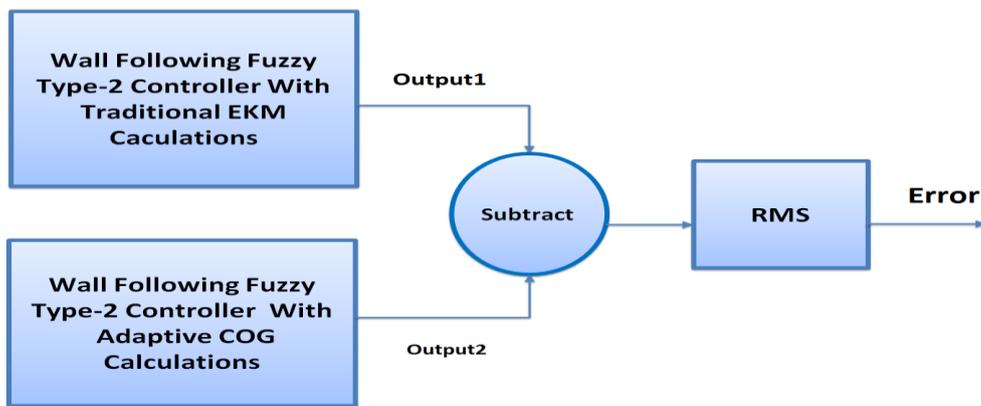


Figure 5-16 : Comparing two fuzzy autonomous wall following controllers.

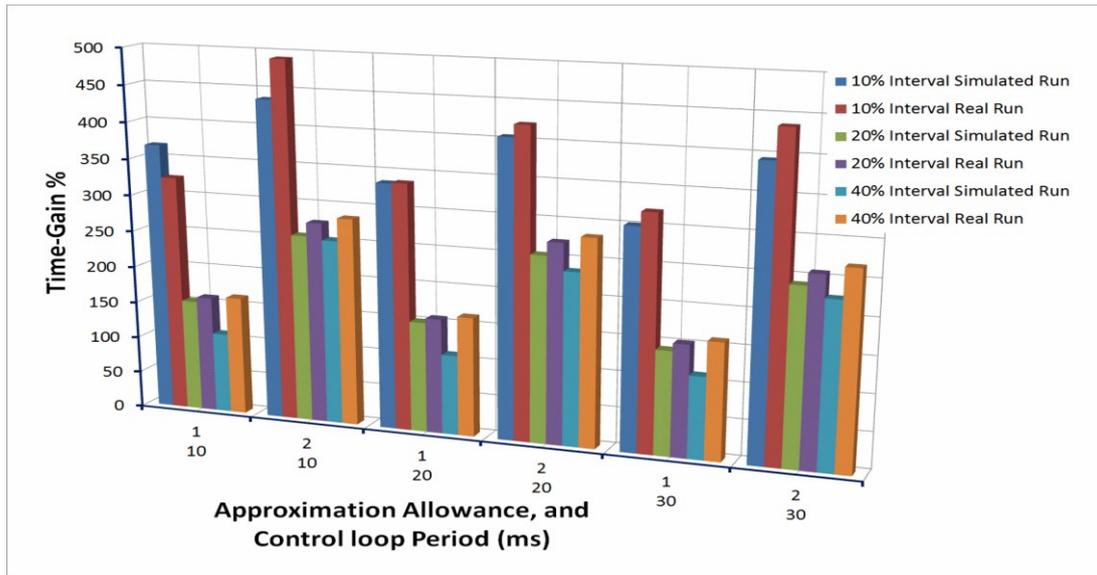


Figure 5-17: Execution time-gain at calculation allowances of one and two using control loop times of 10ms, 20ms, and 30ms for 10%, 20%, and 40% interval widths.

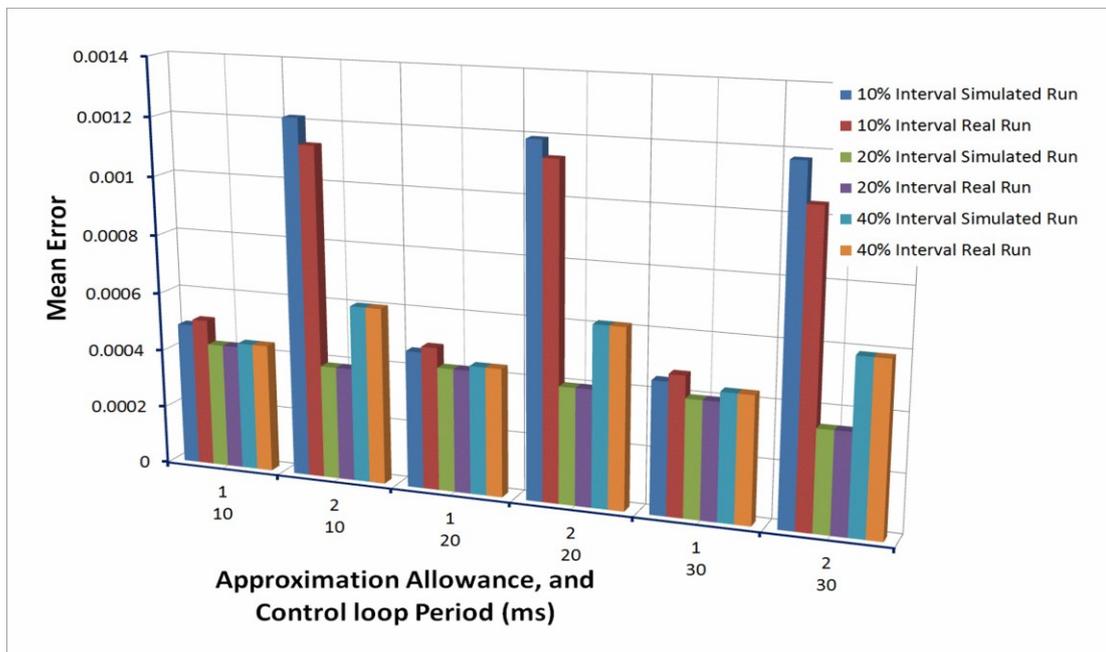


Figure 5-18: Mean absolute error at calculation allowances of one and two using control loop times of 10ms, 20ms, and 30ms for 10%, 20%, and 40% interval widths.

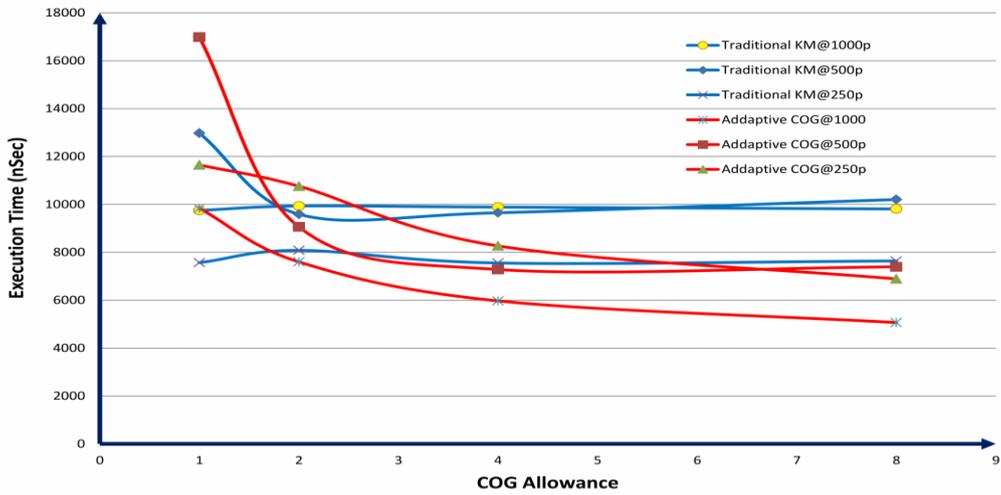


Figure 5-19: The traditional KM and adaptive type reduction time-gain for output discretisation levels of 250, 500, and 1000 points using interval set width of 10%.

Chapter 6: Redundancy Elimination

What has been done in Chapter 4, and 5 is mainly an elimination of the redundant operations out of the iterative KM type reduction. This is performed by using the successive binary search to locate just the important sections that can be used to calculate the type reduction at adequate system error. Similarly, in this chapter, the redundant calculations in the enhanced KM type reduction procedure are going to be spotted and eliminated to achieve faster type reduction. Then, this will be taken further to eliminate the redundancy out of the process of reducing many sets in one go.

6.1. Accelerating the EKM

The improvements of the incremental-EKM proposed by (Duran et al. 2008) can be taken further by calculating the numerator and denominator initials using only the lower membership function, instead of the initial suggestion, which calculates these initials using lower and upper membership functions for left and right uncertainty points, respectively. This will cut, basically, half of the initial method's computation cost and accelerate reaching to the uncertainty points. The left and right uncertainty points can be calculated using this accelerated version using equations (6-1) to (6-4), instead of the old method, which uses equations (2-23) and (2-24), on page 23. A new indexing scheme is suggested to reduce the type reduction cost of this type reduction. This assures faster access to the left and the right uncertainty points by starting closer to them. The proposed indexing uses an incremental scheme starts from zero for calculating the left uncertainty

point and using decremented indexing scheme starts from N while calculating the right uncertainty point. This helps to travel smaller distance, getting to the uncertainty points faster.

$$\underline{Rnum}_{n-1} = \underline{Rnum}_n + y_{n-1} \cdot (\overline{u_{yn-1}} - \underline{u_{yn-1}}) ;$$

$$\text{Using descending index } n = n - 1, \quad N \geq n > 1 \quad (6-1)$$

$$\underline{Rden}_{n-1} = \underline{Rden}_n + (\overline{u_{yn-1}} - \underline{u_{yn-1}}) ;$$

$$\text{Using descending index } n = n - 1, \quad N \geq n > 1 \quad (6-2)$$

For the left point as:

$$\underline{Lnum}_{n+1} = \underline{Lnum}_n + y_n \cdot (\overline{u_{yn}} - \underline{u_{yn}}) ;$$

$$\text{Using ascending indexing } n = n + 1, \quad 0 < n < N - 1 \quad (6-3)$$

$$\underline{Lden}_{n+1} = \underline{Lden}_n + (\overline{u_{yn}} - \underline{u_{yn}}) ;$$

$$\text{Using ascending index } n = n + 1, \quad 0 < n < N - 1 \quad (6-4)$$

Here, initialization of left and right numerators and denominators are equal, as follows: $\underline{Rnum} = \underline{Lnum}$ and $\underline{Rden} = \underline{Lden}$. These initials are evaluated by calculating the COG of the lower membership function that for the interval FT2 set, whom to be reduced. The sequential search of the right uncertainty point is to be terminated when the new calculated rightmost point, y_{Rn} , has become smaller than its preceding result, y_{Rn-1} . The sequential search of the left uncertainty point is to be terminated when the new calculated left most point, y_{Ln} , has become bigger than its previous value, y_{Ln+1} , as illustrated in Figure 6-1.

$$y_{Ln} = \frac{\underline{Lnum}_n}{\underline{Lden}_n}, \quad y_{Rn} = \frac{\underline{Rnum}_n}{\underline{Rden}_n} \quad (6-5)$$

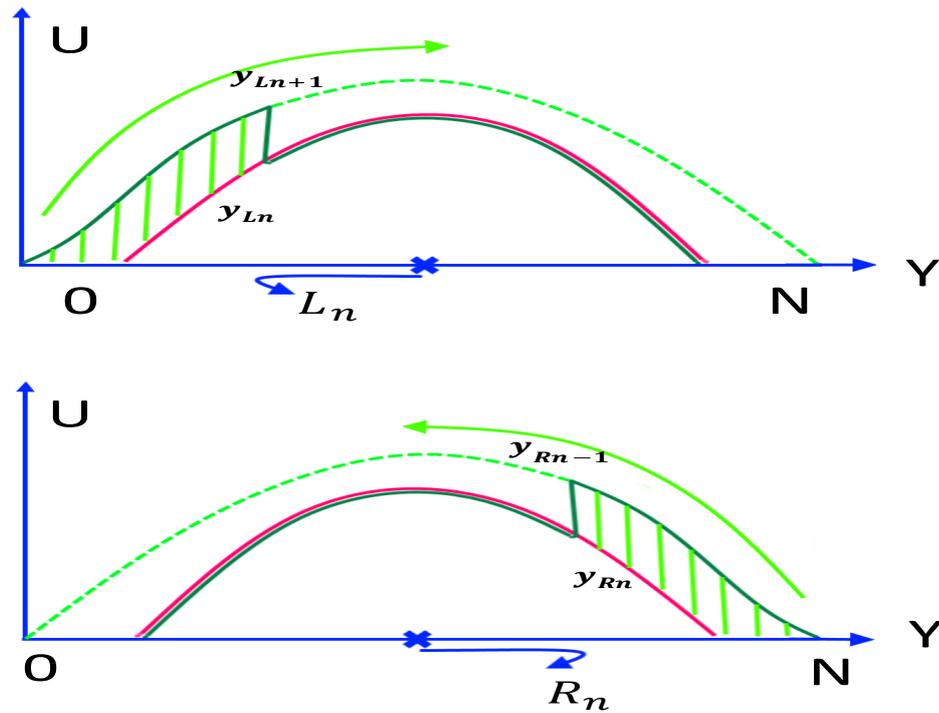


Figure 6-1: Modified incremental EKM.

The computation cut can be expected because of: (a)- Any lower membership function usually has smaller universe extension and fewer discrete elements, (b)- Same initial values are used for the left and right uncertainty calculations, (c)- Two different indexing schemes are used to assure shorter transition distance while getting to the uncertainty points. The resulting execution time-gains from using the above modifications are shown in Figure 6-2 in respect to the traditional incremented EKM-IF procedure, which is described in section 2.4.8, page 23. Different fuzzy interval widths are tested throughout the autonomous fuzzy type-2 controller using simulated in real vehicle. The time-gain resulting from the sets with high interval widths is high because the uncertainty point will fall closer to the ends and the sequential search will locate them faster.

Here, it is impossible to use the proposed adaptation during the initial evaluations because high levels of error will result. This occurs because it is only possible to use the integration result, as described by equation (4-64) on page 83, of the lower membership

function to control the initial values error. This integration result will probably be bigger than the integration result of the interval set that is shaped by this LMF, thereby generating unexpected and uncontrolled computation error while evaluating the uncertainty points.

This accelerated EKM procedure is implemented during our tests using java programming language. The exact coding is shown in Appendix-C in two parts. The first is for the left uncertainty point calculations, while the second is for the right point.

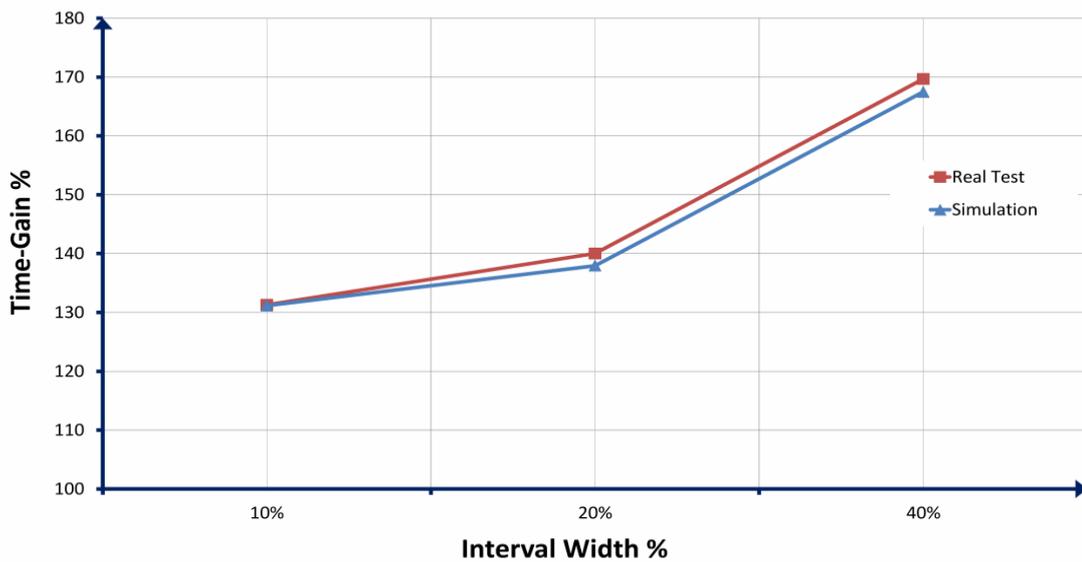


Figure 6-2: Execution time gain of the modified incremental EKM execution time in comparison to the initial incremental formula using different interval set widths.

6.2. A Global Type Reducer

Throughout most of the commonly implemented type reduction procedures, it is essential to use regular fuzzy type-2 set (Chen et al. 2013), this is, to have convex upper and lower membership functions. Otherwise, most of the proposed type reduction procedures will only provide an approximated result, depending on their irregularity.

However, a similar issue will appear if many fuzzy type-2 sets are reduced in one go. Achieving this requires merging their individual universes into one global universe, which will definitely have less computation cost, compared to the normal case where each fuzzy set has to be reduced individually. This expected computation cost cut comes from the fact that all of the overlapped sections between the sets will be processed into the type reduction ones instead of twice or more. This can be seen more clearly in Figure 6-3, where reducing the six interval sets individually, as what is usually done, will require performing the reduction six times using $6 \times N$ elements. Here, every set is assumed to have N discrete divisions. However, performing the type reduction in one go, while all the sets are placed on one global universe, will require using only $3 \times N + 0.5 \times N$ elements. This is resulting due to the excluding of the redundant calculations associated with the overlapped sections. In general, if it is required to reduce K fuzzy sets, each having an overlap ratio of 50% and each one has N elements, using normal techniques individually, then it is required to process $K \times N$ elements. Reducing them in one go will cut about half of these calculations reaching to process only the points amount shown below.

$$\text{The One-Go points} = N \times \left(\frac{K}{2} + 0.5 \right) \quad (6-6)$$

From this, it is possible to sketch the time-gain graph of this new type reduction with respect to the accelerated EKM. In this sketch, it is assumed that different identical systems do exist but they have different fuzzy set counts, ranging from one to 12 sets. The resulting time-gain for each case is shown in Figure 6-4. The ultimate, ideal, time-gain situation will be 200% if too many fuzzy sets are reduced all together using the one-go procedure.

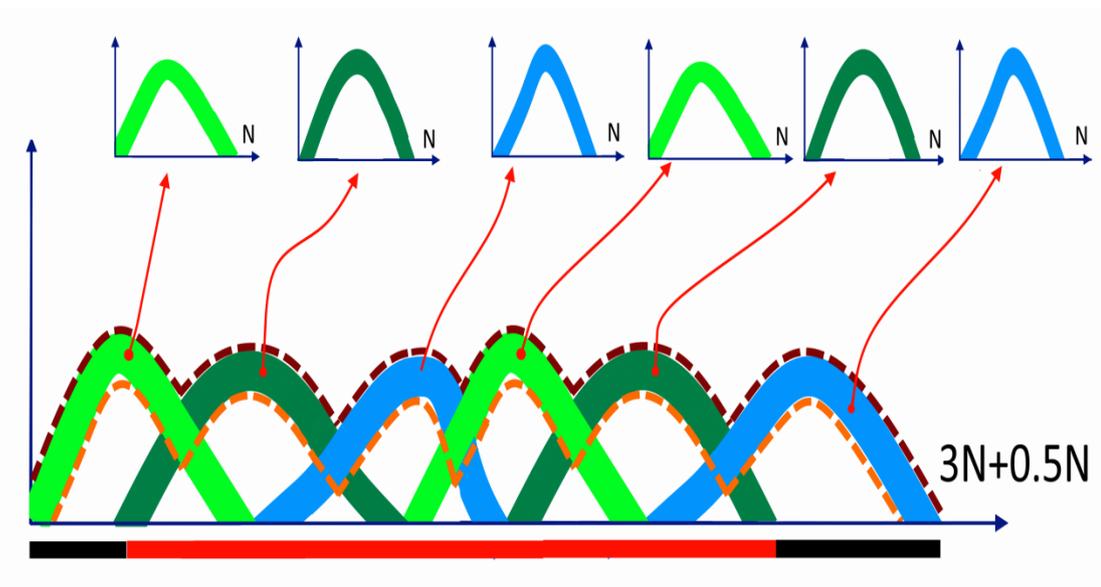


Figure 6-3: Possible computation cut due to using a one-go type reduction

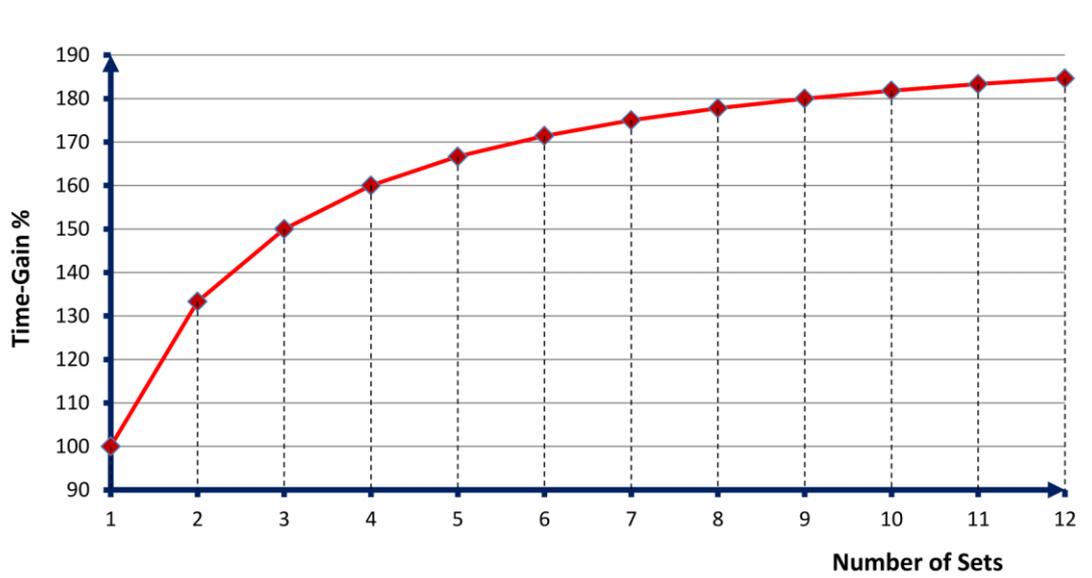


Figure 6-4: Time gain due to reducing multiple sets in one go

6.3. The Proposed One-Go Type Reduction Procedure

Our proposal for this one-go type reduction routine is based on principles very similar to the global optima search process used to locate maximums and minimums. However, this can be implemented in a cost-effective manner by extending the accelerated EKM-IF type reduction procedure to the following form:

- 1- Construct the global LMF by taking the maximum of the lower membership values which share one universe point.
- 2- Construct the global UMF by taking the maximum of the upper membership values which share one universe point.
- 3- Calculate the initial numerator and the initial denominator using the global LMF, in the same way, it is been calculated for the accelerated EKM (presented in section 6.1).
- 4- Calculate the global stop (GS) point by dividing the initial numerator by the initial denominator.
- 5- For the right uncertainty point, use the global universe elements, search sequentially using a decremented indexing, start from the highest index, stop at the GS point, and use the global UMF.
- 6- During this search, keep the maximum result only:

IF(Current Right > Global Right) THEN (Global Right = Current Right)
- 7- For the left uncertainty point, use the global universe elements, search sequentially; use incremental indexing, start from index zero, stop at the GS point, and use the global UMF.
- 8- During this search, to get the left uncertainty point, keep the minimum result only:

IF(Current Left < Global Left) THEN (Global Left = Current Left)

6.3.1. Benefits of the Proposed One-Go Procedure

In addition to the cost cut resulting from reducing multiple sets in one go, this procedure opens the door to using non-regular fuzzy type-2 sets. These sets are more flexible, which can produce more accurate system modelling and better controllers' tuning. Also, using such non-regular fuzzy type-2 sets can reduce the total fuzzy sets in a system, leading to a dramatic reduction in a system's complexity and rule base sizes.

In spite of the benefits of the proposed One-Go type reduction, it is not implemented in our autonomous vehicle because it currently has only two Gaussian output fuzzy sets. Such small controller is very good for practical evaluations of the common type reduction procedures because the rule base will be small. However, the practical evaluation of the One-Go type reduction procedure would require a bigger fuzzy controller with many output fuzzy sets. Building such controller and testing the time-gain practically is in our plan for the future work.

Chapter 7: Implementation Aspects

7.1. The Autonomous Vehicle

In this research, the tests of the autonomous fuzzy type-2 controllers are performed using the simulator MobileSim from ActivMedia[®] robots. The Pioneer P3-DX mobile robot, shown in Figure 7-1, is selected to be the platform for the practical tests. It is fully modelled by the above simulator and has sixteen ultra-sonic sensors, eight of them are for the front motion and the rest are for the back. This series of ground robotic vehicles are of those types, whom are common, flexible, and can be upgraded easily with wide span of accessories. They have been used a lot by many researchers to develop autonomous fuzzy controllers, as an example: (Yaonan et al. 2011; Wagner and Hagra 2009) ,

The software development toolkit for this robot is supported by a library named Advanced Robot Interface for Applications (ARIA). This library can be used with Java and C++ languages. The Java language is chosen here to write the fuzzy control and the type reduction algorithms because Java is an object-oriented programming language, robust, reliable, and portable (Cingolani and Alcalá-Fdez 2012; Collier and Meyer 2000). From those features, robustness and portability are the most important features for robotic systems, which dominate this selection.

You can find in appendix-D the implementation of the successive binary search routine that being used throughout the autonomous vehicle controller and type reduction evaluations



Figure 7-1: The Pioneer P3-DX mobile robot.

7.2. Using Schmitt-Trigger

In fuzzy controllers, a specific fuzzy rule-base has to describe the system model and any required decisions using linguistic IF-THEN commands. However, in some cases, the available data can be limited and cannot be used to make a decision even if a human is using it. A simple example of such case can be seen for a typical autonomous vehicle equipped with front, right and left distance sensors and which uses the rule base shown below. A decision ambiguity is happening in this case that can be seen in rule number three: Here, one of two possible actions should be taken when there is no other data that can give a clue to which one is preferred.

1-	If left is far & front is far & right is far then	Go forward
2-	If left is far & front is far & right is close then	Go slight left
3-	If left is far & front is close & right is far then	Ambiguity
4-	If left is far & front is close & right is close then	Go hard left
5-	If left is close & front is far & right is far then	Go slight right
6-	If left is close & front is far & right is close then	Go forward
7-	If left is close & front is close & right is far then	Go hard right
8-	If left is close & front is close & right is close then	Go back

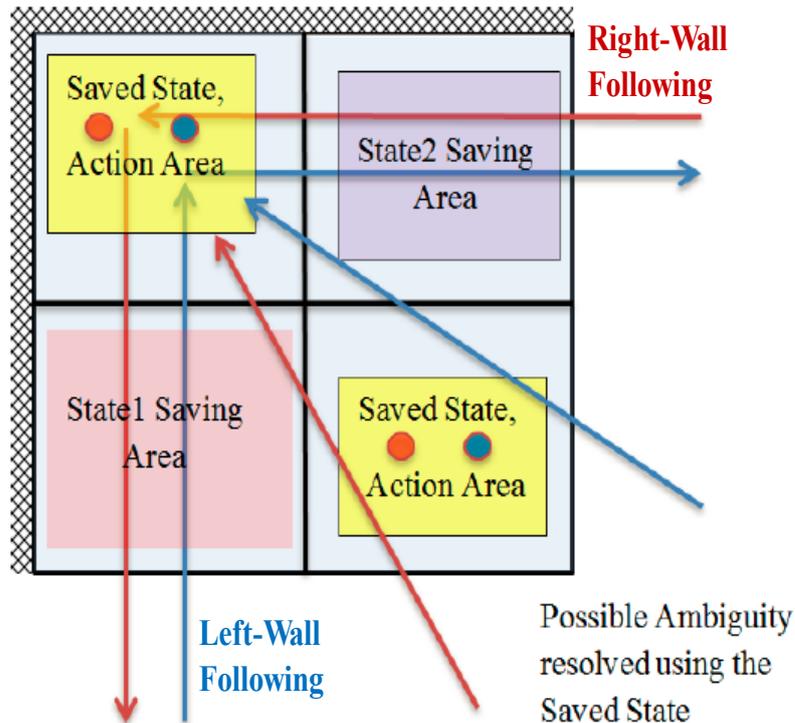


Figure 7-2: Areas where the left or right wall following state can be captured and where they can act.

One of the solutions, for such cases, can be seen in (Figueroa et al. 2005) (Chang et al. 2013), where random directions are used. This randomness is acceptable in limited and simple cases only. In (Melinguì and Chettibi 2013), the potential field evaluation has been suggested to decide the best autonomous vehicle direction but this still cannot suggest correct direction if symmetrical objects exist around the vehicle. The work in (Nurmaini and Tutuko 2011) had used the fuzzy type-2 autonomous controller with neural network controller to identify the environment but they chose one preferred direction in such ambiguity cases, which is still a limited solution. Here the absolute correct decision cannot be taken if the fuzzy controller is using the current state data only. The environment map can be used to get correct decisions, but using maps can generate excess over-heading delay and can have a negative effect on evaluating the type reduction performance. In addition, while constructing a new environment map, it will require finding some simple solution capable of resolving such ambiguity. The human, in

such cases, makes his decision by using some simple old data in addition to the current details. In a similar way, any old fuzzy states can be used to resolve such ambiguity conflict. Such conflict can appear in cases where acute wall corners and symmetrical environments exist. Such conflict has been addressed in (Mucientes et al. 2007), attempting to resolve it throughout accurate fuzzy rules tuning, but they still do not deliver a real solution, only considering such corners as dangerous areas that have to be avoided early. The decision-making in the implemented wall following controller is based on using some saved old states with the current input readings to generate correct actions that can overcome any conflict situations. The states, that have to be recognized and saved to define the future actions, are identified using some extra rules designed specifically for this purpose. As shown in Figure 7-2, for a situation where the autonomous vehicle is passing close to a right wall, the state "right wall" is saved. Similarly, if the autonomous vehicle is passing close to a left wall, the state "left wall" is saved. These states are used to resolve any possible ambiguity conditions like those whom are resulting when getting close to acute wall corners or running in a long narrow corridor. The update process of the saved state requires some care, where the symmetrical environment can trigger successive fast state switching. This fast state change would give an average action between the two actions, where they should not be averaged. These two behaviours outputs, at worst, can cancel each other, causing a vehicle collision. Here, the old right-wall state, for example, has to be kept active till the vehicle goes relatively far away from the right wall and close enough to the left wall. This principle has been implemented in our fuzzy wall-following controller by using an external Schmitt-trigger to correctly discriminate one behaviour (either left or right wall-following). In our fuzzy controller, one output ("state") has been generated using two output fuzzy sets, left and right. Each one of the two sets has one rule which controls its fuzzy membership level, as follows:

1-	IF left distance is close THEN	The state is left wall-following
2-	IF right distance is close THEN	The state is right wall-following

Normally, these state outputs will go low if the left and the right distances are symmetrical; this leads to fully conflicting rules action. However, if the state output is being fed to a Schmitt-trigger then any mid-valued levels will be shifted to either minimum or maximum, depending on the previous Schmitt-output level. The Schmitt

output will feedback to the fuzzy “L/R-select” input; any ambiguity in approaching wall structures having an angle of 90 degrees or less, or entering a long corridor, thus can be easily resolved.

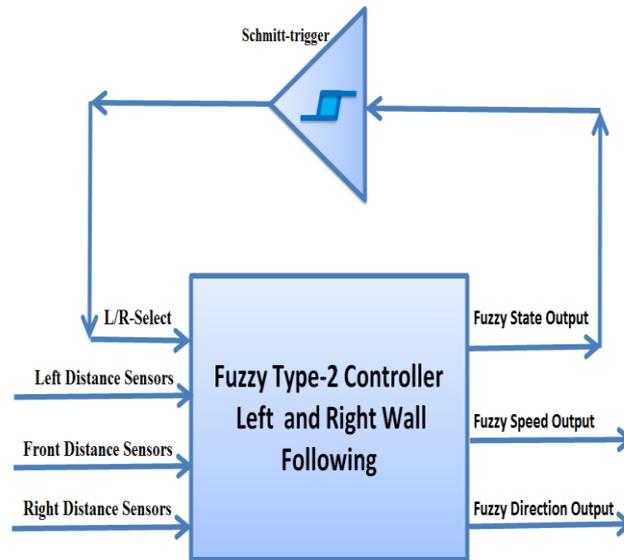


Figure 7-3: Left and right wall-following discrimination using Schmitt-trigger

7.3. Using Schmitt-Trigger

Based on the required accuracy and complexity, different sensing devices can be used in autonomous vehicles, such as a stereo vision camera (Achtelik et al. 2009), a 3D laser (Zhuang et al. 2013), and the differential GPS systems (Rodríguez-Castaño et al. 2000), and ultrasonic distance sensors. However, the sensors readings will be processed as numbers inside digital autonomous vehicle controllers with no indication how accurate any reading is. This means the cheaper the sensors are, the harder to use. Therefore, only those basic ultrasonic sensors of the Pioneer P3-DX are used during the fuzzy type-2 autonomous vehicle controller’s evaluations in order to expose it to high uncertainty levels. These ultrasonic sensors are mounted around the robotic vehicle to form a circular detection pattern. However, processing the different directions’ measurements depends on the required job nature. For a dynamic environment and multi-robotic fields, the circular sense pattern is preferred, where a collision is expected from

all directions (Chang et al. 2013). However, for a wall-following operation, two different actions have to be constructed, for front and side detectors. The obstacles, which fall at a specific distance from the vehicle side, would require less attention than a similar obstacle occurring at a similar distance but in front of the vehicle. This can be considered through the fuzzy rule base. However, another approach is suggested here. It works by reshaping the sensing pattern into a slim vertical rectangle, as shown in Figure 7-4, to get simpler rule-base that can generate different action strengths for those objects falling at similar distances but at different angles with respect to the moving direction. This reshaping process creates a long detection range ahead and a short detection range on the sides (the side detection range has to be short to offer the vehicle more flexibility in getting inside narrow sections in various environments). In addition, this rectangular detection pattern simplifies the alignment of the vehicle alongside walls using less fuzzy rules. This reduces the total fuzzy sets computation time and system complexity, enabling better evaluation of the performance at the type reduction stage. In our wall-following controller, the sonar readings are pre-scaled as: $(side\ distance)/\cos(\theta)$, where θ is the detector angle with respect to the side direction. The *side distance* is the closest apart required to the left or the right direction. This pre-scaling process can offer a simple and real time adaptation based on the required acting and the current moving speed of the vehicle.

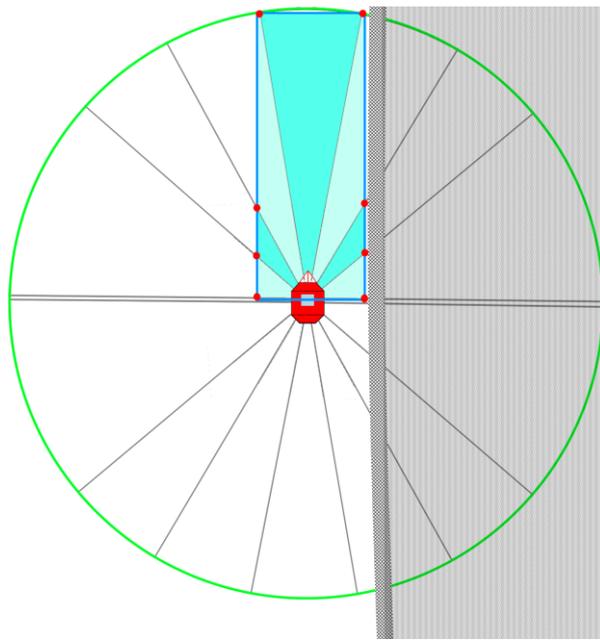


Figure 7-4: Scaling the distance sensors for the wall-following behaviour

Chapter 8: Conclusions and Future Work

8.1. Conclusions

The literature review conducted throughout this work covers all type reductions known up to date. It shows the importance of the Karnik–Mendel type reduction family. The features of these type reductions are recommended for control systems, such as their smooth output, uncertainty measures, accuracy, and generality. Their impact on autonomous fuzzy controllers and mobile robot controllers are revealed by means of another survey conducted here. It is therefore important to continue the research in order to reduce the computation cost of these type reduction procedures and to then utilise that in an autonomous vehicle controller.

By the end of this research, the following conclusions were reached:

The effect of the main bottleneck in the fuzzy type-2 controllers (the type reduction calculations) has successfully reduced in different methods. An adaptation is suggested to cut some of the computations associated with the type-reduction core computation (the centre of gravity, or COG). The adaptation is based on locating the sections of the fuzzy sets containing straight lines and slight curves. These sections are considered to be straight lines, which are used to perform faster COG calculations within type reduction procedures. Such segments are located using a binary successive search approach. This technique has never (according to the extent of our knowledge) been used before in fuzzy systems. The required output error of the fuzzy type-2 controller is used to decide to what extent the binary search is aggressive. A fixed accuracy is suggested, first throughout the iterative KM type reduction to reduce its computation cost. The resulting time-gain was not very high. A variable accuracy level is proposed as an alternative

during the iterations of the Karnik–Mendel type reduction. A higher time-gain was thus achieved while at the same time preserving the final error within required limits.

During the approximation process, a negative impact was spotted due to the resulting discontinuity, which generates extra unwanted iterations degrading the progress rate of the iterative KM type reduction. A suitable state detection flag was used to trigger an averaging process; this managed to resolve the issue.

Extra enhancements were proposed to the enhanced Karnik–Mendel type reduction in its incremental form (EKM-IF). These are related to using better initial calculations and a better indexing scheme, which assures accelerated reach to the left and the right uncertainty points. Both the simulated and the real tests show a resulting time gain approaching 170%. The original accuracy is intact by the proposed acceleration.

A new type reduction technique is proposed to reduce multiple fuzzy sets altogether in one go. This one-go procedure has the potential to exclude the computation redundancy which exists due to fuzzy sets overlapping. The efficiency of this procedure is analysed for common overlapping cases, showing a time-gain approaching 200%.

The computation cost comparisons (of the proposed adaptive type reductions with respect to the iterative KM) are showing a time-gain ranging from 150% to 500%, depending on the fuzzy interval width and the control loop delay. The final fuzzy type-2 controller performance error was in the range of 10^{-3} (in comparison to the iterative KM) when a calculation allowance of one discrete element, out of 1000 elements, is used.

The evaluations are performed using an autonomous fuzzy type-2 controller accomplishing a left and right wall following, with obstacle avoidance tasks. These tasks are considered the most crucial behaviour for autonomous vehicles, especially in terms of navigation in new environments. The choice of this behaviour for the evaluations was successful because of the ease in creating different operation conditions, which can impose a wide variety of fuzzy firing levels.

A cheap solution based on current state latching principles using an external Schmitt trigger is suggested for fuzzy controller systems to resolve rule-base ambiguity, thus to arrive at a correct decision. This approach managed to remove the partial and full rules

conflict, which appears close to acute corners and narrow corridors. Implementation was simple where just one extra output and two rules were added.

A dynamic sensing pattern is proposed and implemented in our autonomous vehicle. This approach controls the sensitivity of the sensors, according to the currently running task, and is very successful, helping to reduce the fuzzy rule base and thus promoting better evaluation at the type reduction stage.

8.2. Contributions

The final outputs of this work are listed below:

- I. Design and analysis of a binary successive search technique to locate straight and slightly curved segments in fuzzy sets based on a required output error limit.
- II. Design of a method to detect and process unwanted iterations happening within the iterative KM type reduction due to discontinuity caused by approximations.
- III. Design of an accelerated version of the enhanced KM-IF type reduction procedure. Better initialisations are proposed, which can be used for left and right uncertainty calculations. Shorter searching distances are used, individually, for the left and for the right uncertainty calculations.
- IV. Design of a procedure to reduce non-regular fuzzy sets. Analysis and implementation of this procedure to reduce multiple fuzzy sets in one go. This one-go procedure excludes redundancy of the overlapped sections to achieve performance time-gain approaching 200%, in comparison to the enhanced KM-IF type reduction procedure.
- V. Design of a simple ambiguity resolving method and a dynamic sensing scheme, for autonomous wall-following behaviours, to minimise the fuzzy rule-base complexity. That is useful to get accurate performance measurement at the type reduction and defuzzification stages.

8.3. Future Work

Although this work has successfully managed to cut some of the high cost associated with the accurate type reduction procedures, it is possible to do more work for the sake of type-reductions and autonomous vehicle controllers. Those are outlined in the following items as a future work, thus someday to be done.

- **Parallel Processing.**

Although, it is suggested by this research to use the binary successive search, but it is being implemented using sequential procedure running in a single microprocessor. However, building parallel type reduction procedures is a promising research field for this searching technique and for the adaptive integrations, in general.

- **Using Field-Programmable Gate Array chips.**

In spite of the high speed and low cost of the microprocessors, it can be seen how many control systems are migrating to the field-programmable gate array chips (FPGA). This is a very promising working field for the type reduction calculations because the FPGA chips can offer a simple multiply and add logic gate arrays. This is an essential aspect to achieve excellent throughput rate of the type reduction calculations.

- **Negative Fuzzy Rules.**

It is known that fuzzy rule bases are trying to mimic the human knowledge and decision-making, but human negative knowledge like "Don't do this----- again, IF you are-----" is very important way to define new fuzzy rules, which could be very small and efficient. By this way, dozens of rules that are using the form "IF this----- THEN do this-----" can be omitted. This approach is very suitable for fuzzy re-enforcement learning systems. Such systems are very important for autonomous vehicles approaching and discovering new environments.

- **Circular Universe.**

Throughout most fuzzy systems, linear data universes are used. This is very much like the rectangular coordination system. However, many physical quantities around us can be presented much better using polar coordinating system. For example, the obstacle positions around an autonomous vehicle. In such/similar cases, it will be more efficient to use circular fuzzy type-2 sets. The idea of using such circular fuzzy set is new, and it has only been implanted in some systems using FT1 sets. Thus, there is a big research field to be covered relating to implementing the circular fuzzy controllers, building the required rule bases, performing the type reduction etc...

Appendix-A

The performance time-gain tests result for the proposed nonlinear approximated type reduction at different allowance levels and calculation clearance of the second iteration.

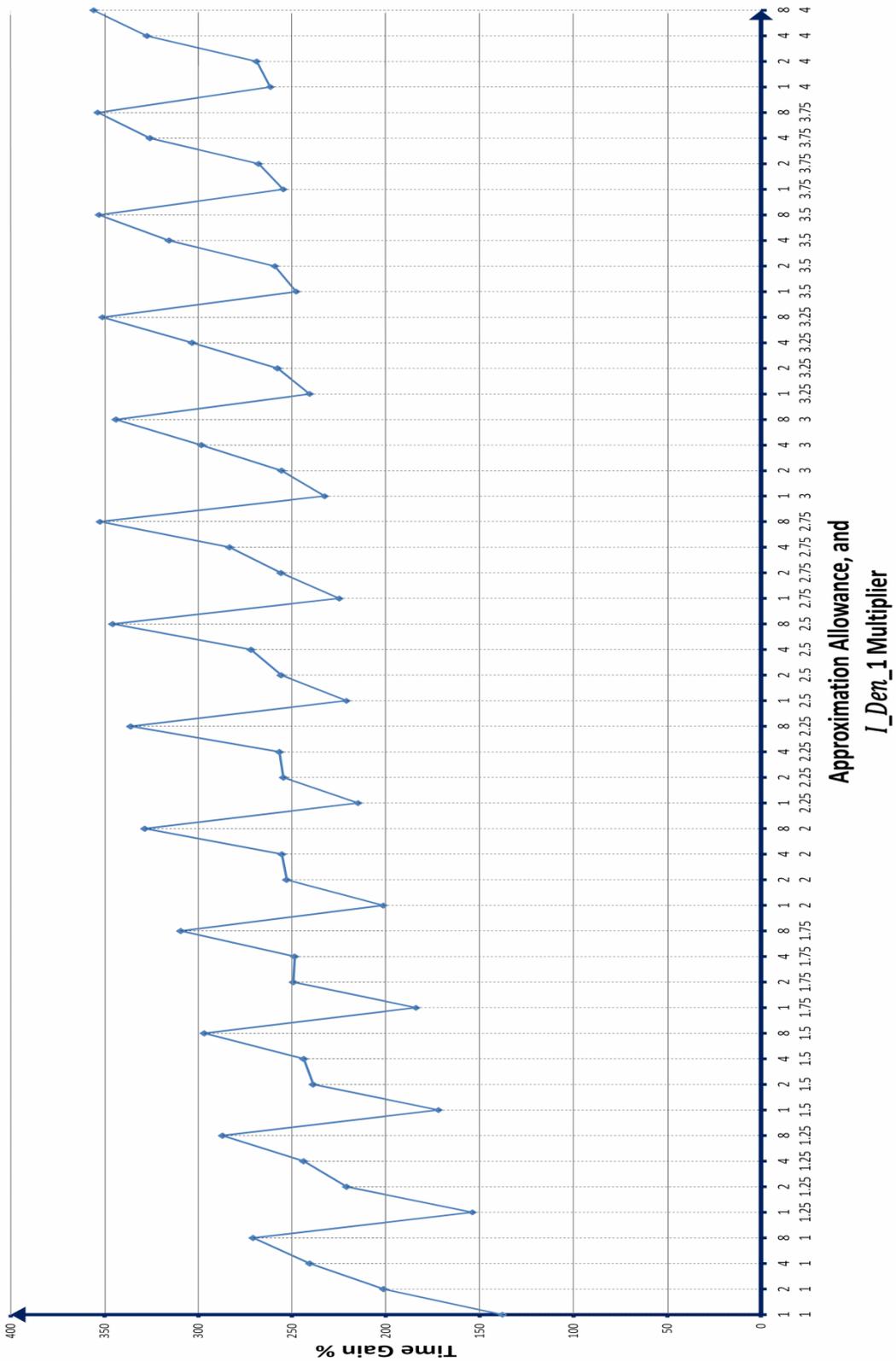


Figure 8-1: Average time-gain for the approximated type reduction using Gaussian IFT2 set having width of 10% and fuzzy transition step size of 10%, at different approximation allowances and various integration multipliers of the second iteration.

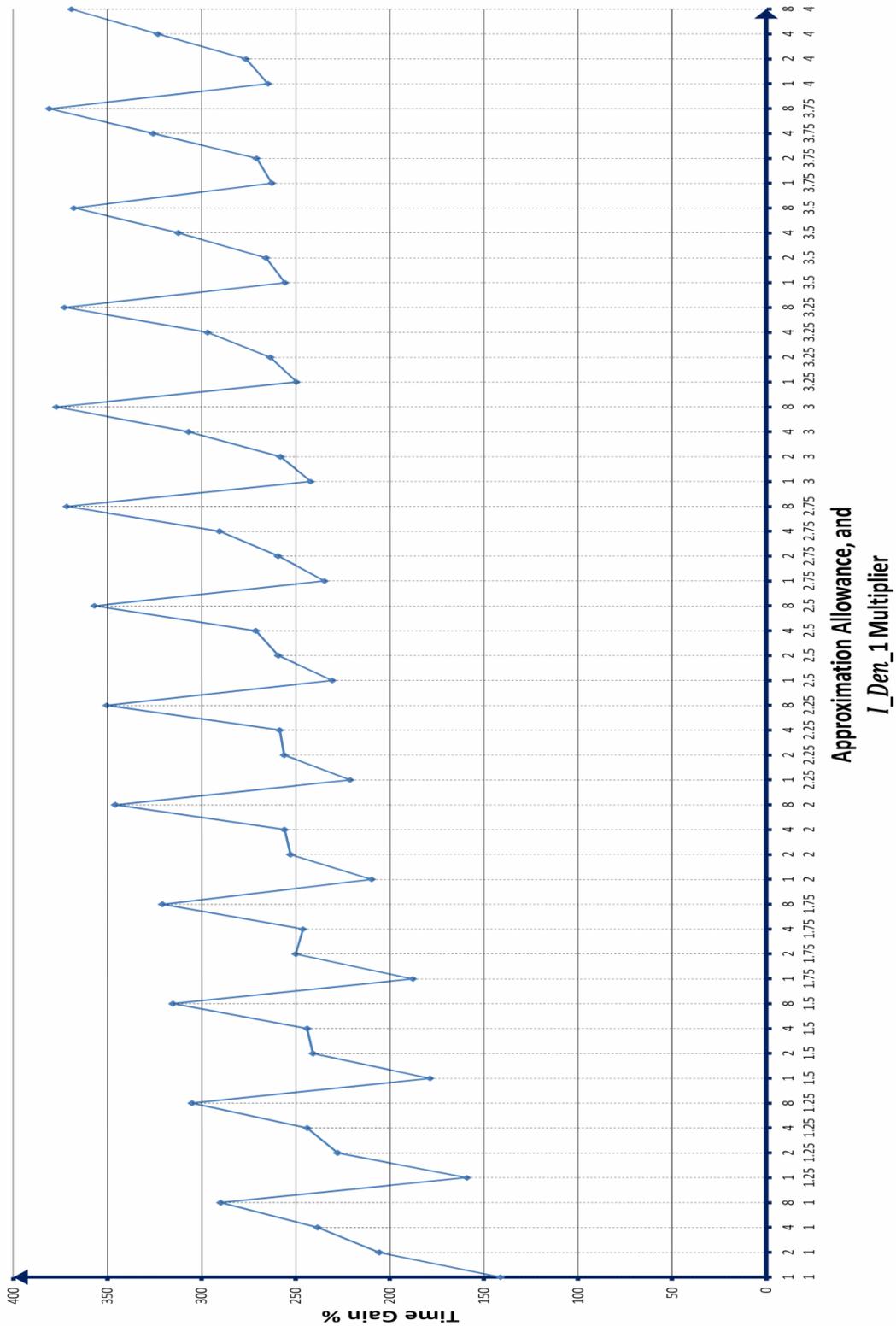


Figure 8-2: Average time-gain for the approximated type reduction using Gaussian IFT2 set having width of 20% and fuzzy transition step size of 10%, at different approximation allowances and various integration multipliers of the second iteration.

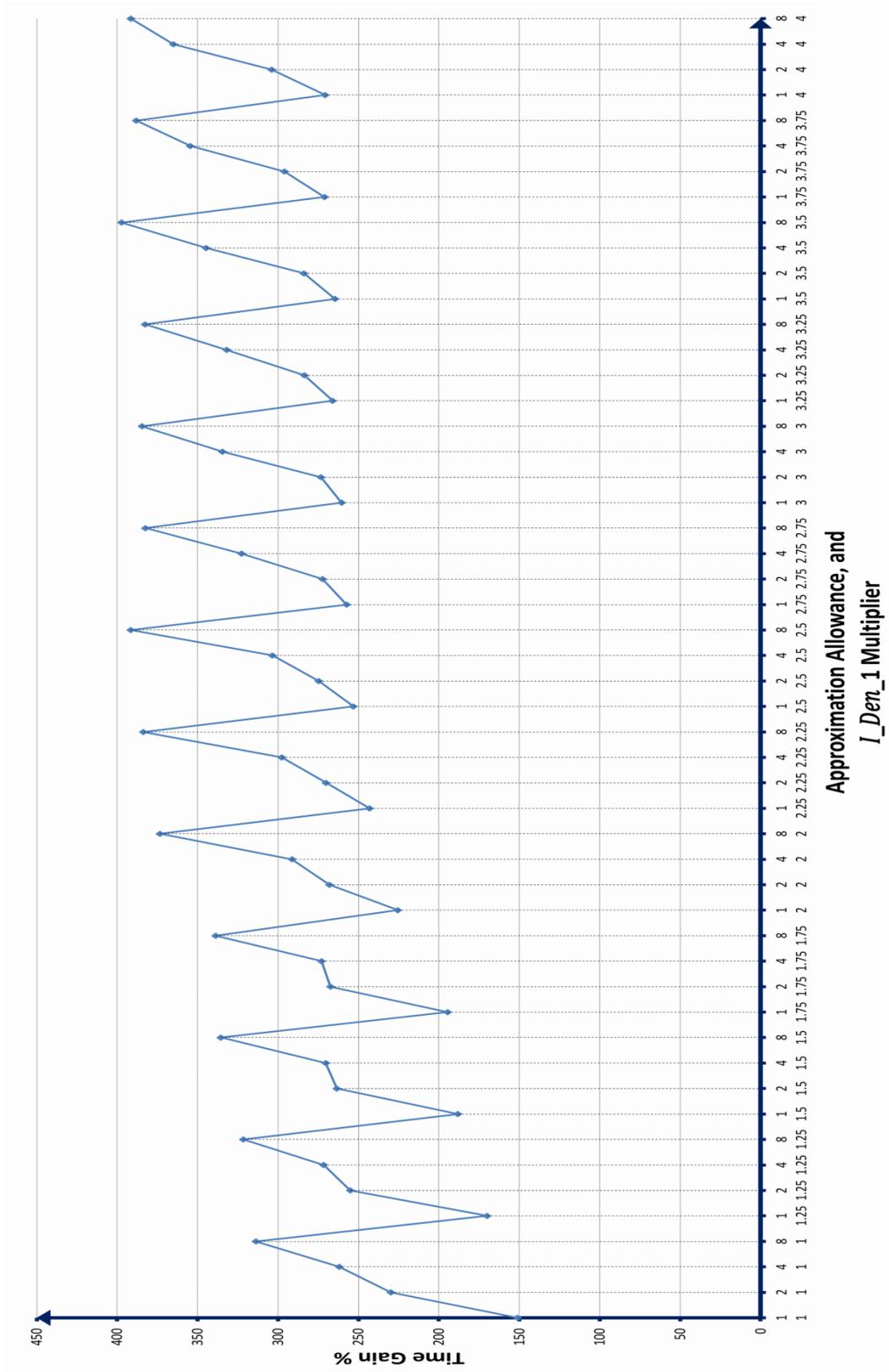


Figure 8-3: Average time-gain for the approximated type reduction using Gaussian IFT2 set having width of 40% and fuzzy transition step size of 10%, at different approximation allowances and various integration multipliers of the second iteration.

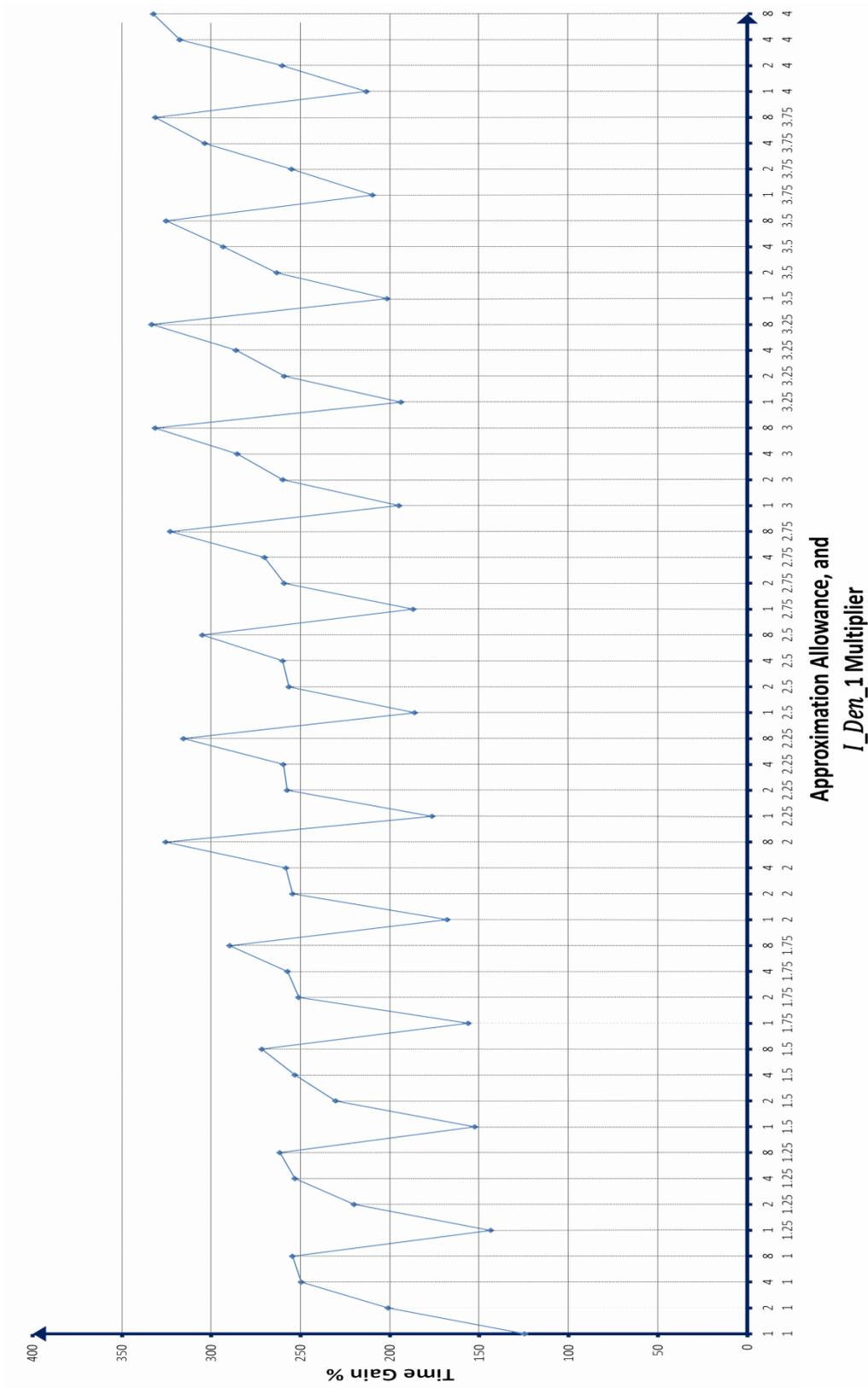


Figure 8-4: Average time-gain for the approximated type reduction using Gaussian IFT2 set having width of 10% and fuzzy transition step size of 20%, at different approximation allowances and various integration multipliers of the second iteration.

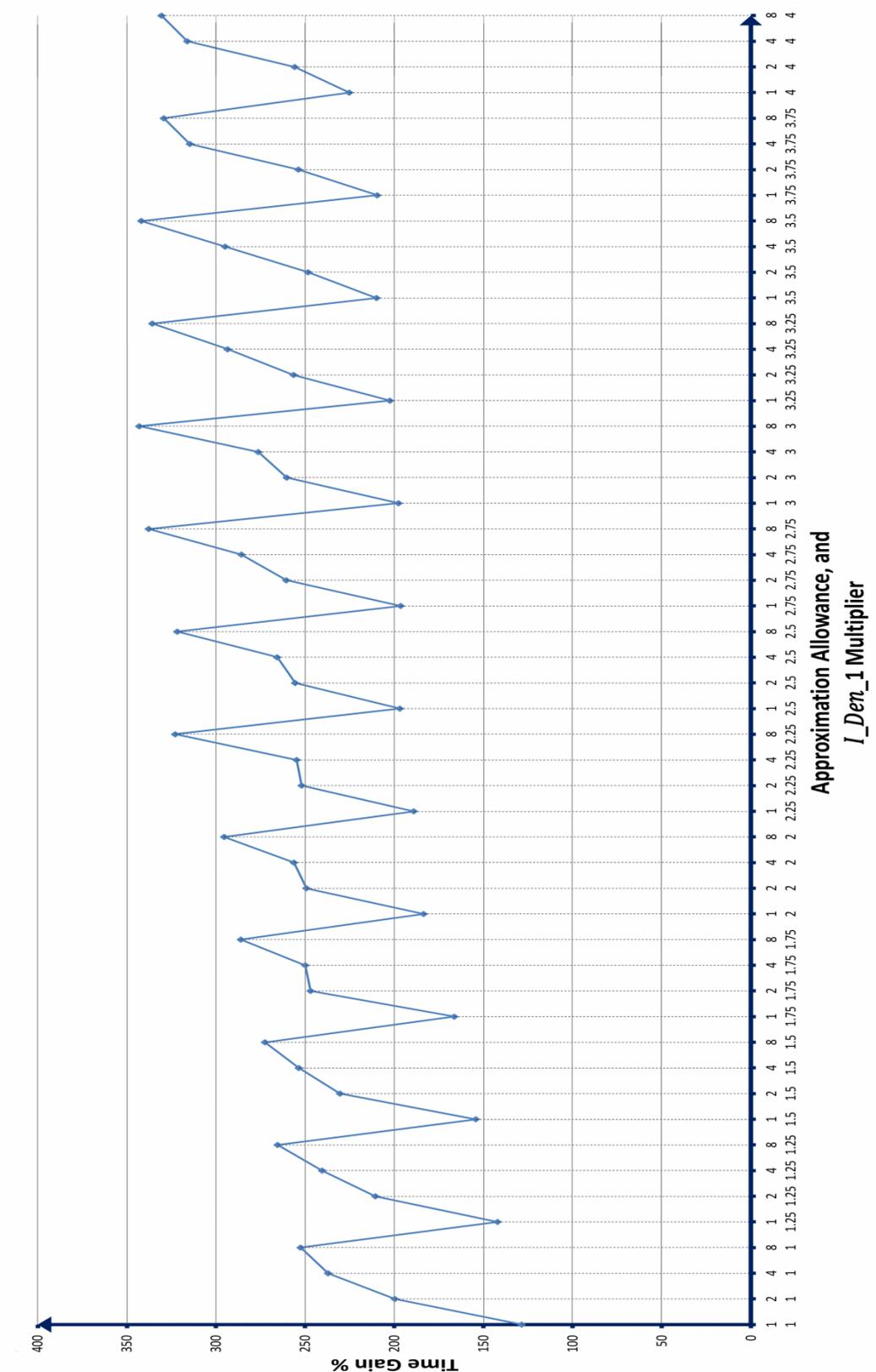


Figure 8-5: Average time-gain for the approximated type reduction using Gaussian IFT2 set having width of 20% and fuzzy transition step size of 20%, at different approximation allowances and various integration multipliers of the

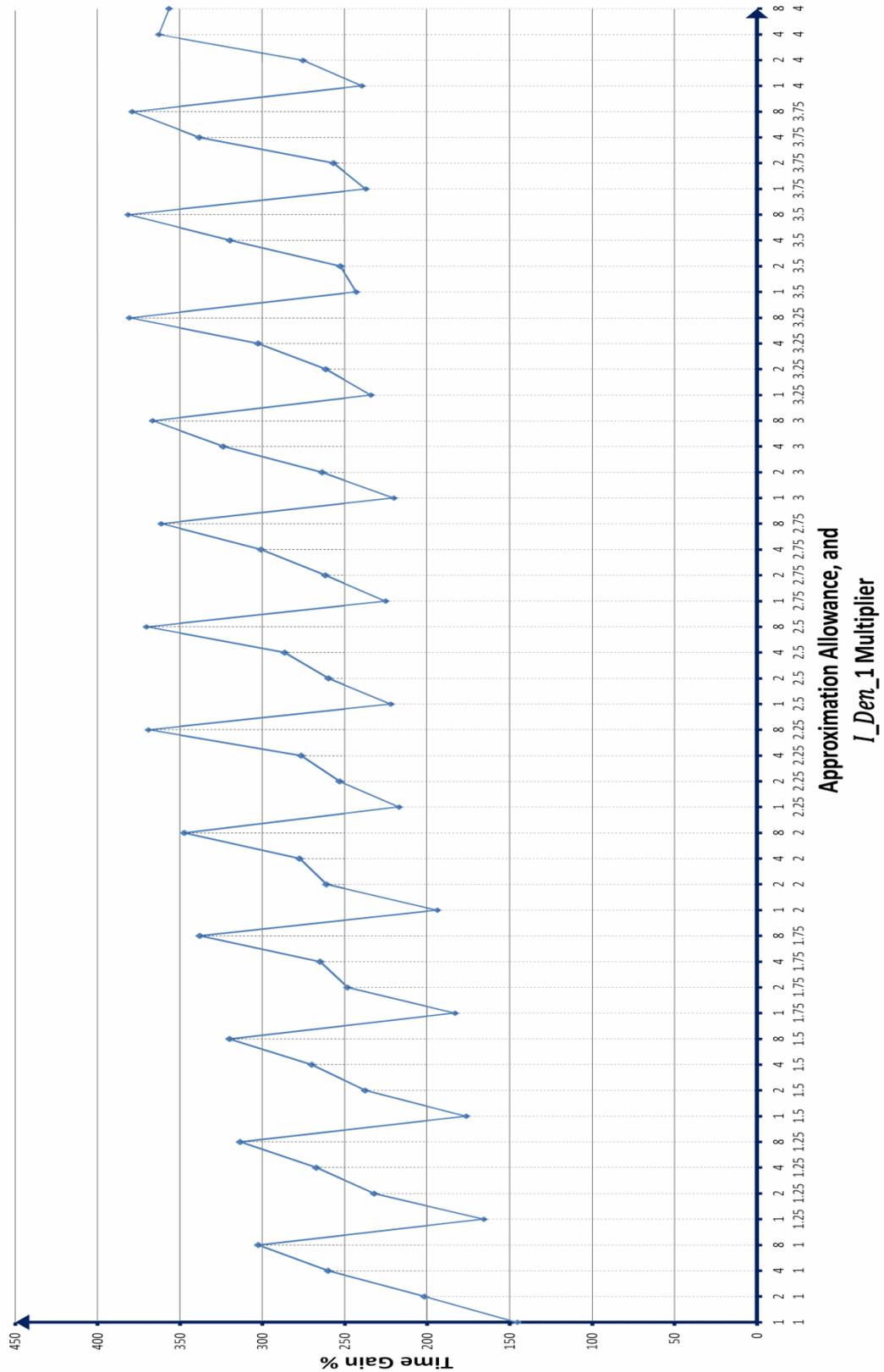


Figure 8-6: Average time-gain for the approximated type reduction using Gaussian IFT2 set having width of 40% and fuzzy transition step size of 20%, at different approximation allowances and various integration multipliers of the second iteration.

Appendix-B

The results of the maximum error tests that are performed on the proposed nonlinear approximated type reduction using all the possible firing levels are presented in this appendix. The used firing levels are transitioned using two different steps, 10%, and 20%. A wide span of allowance levels and calculation clearance, at the second iteration, are used with different fuzzy interval widths to cover all the possible practical cases.

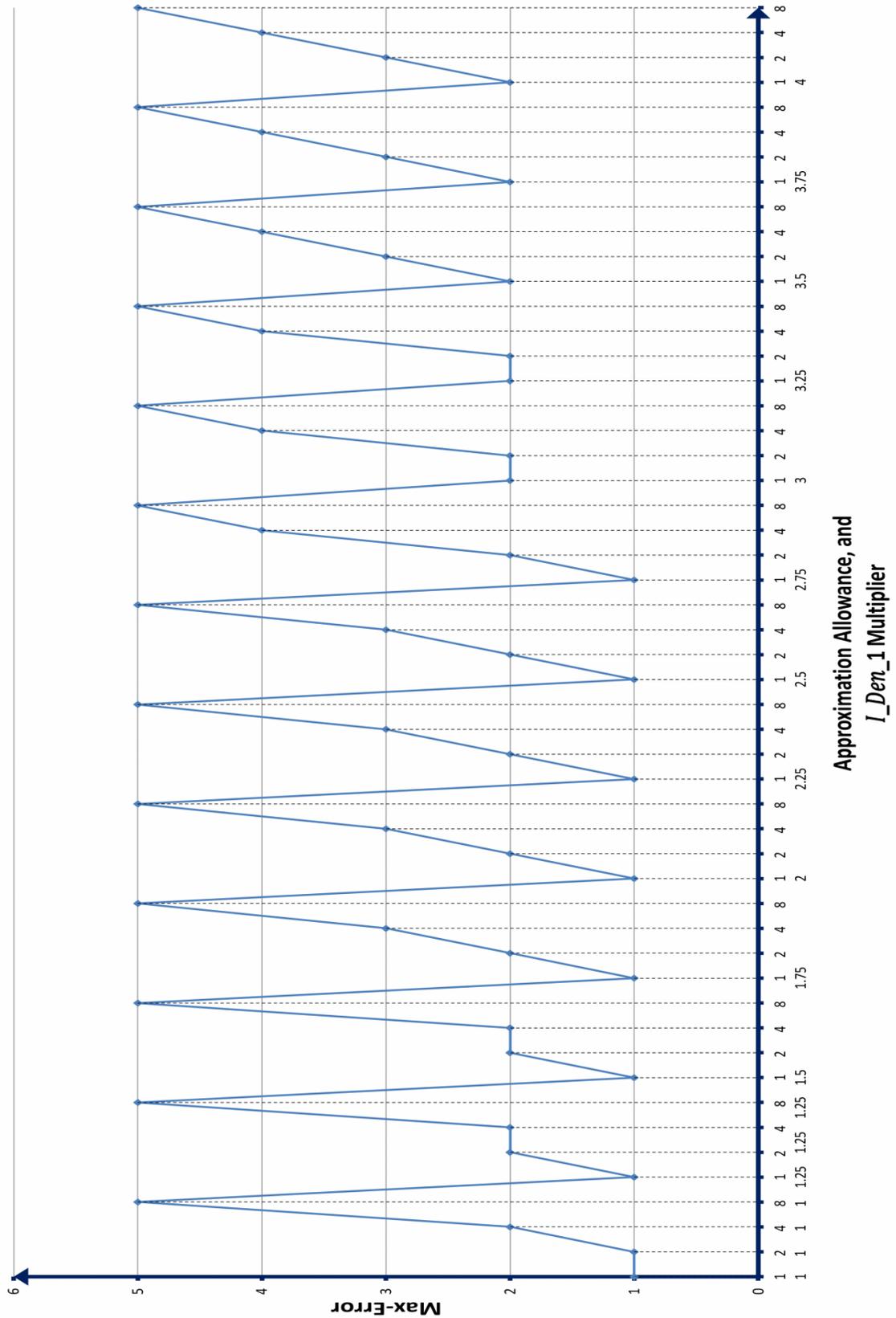


Figure 8-7: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 10% and incremental fuzzy transition step size of 10%, at different approximation allowance multipliers during the second iteration calculations.

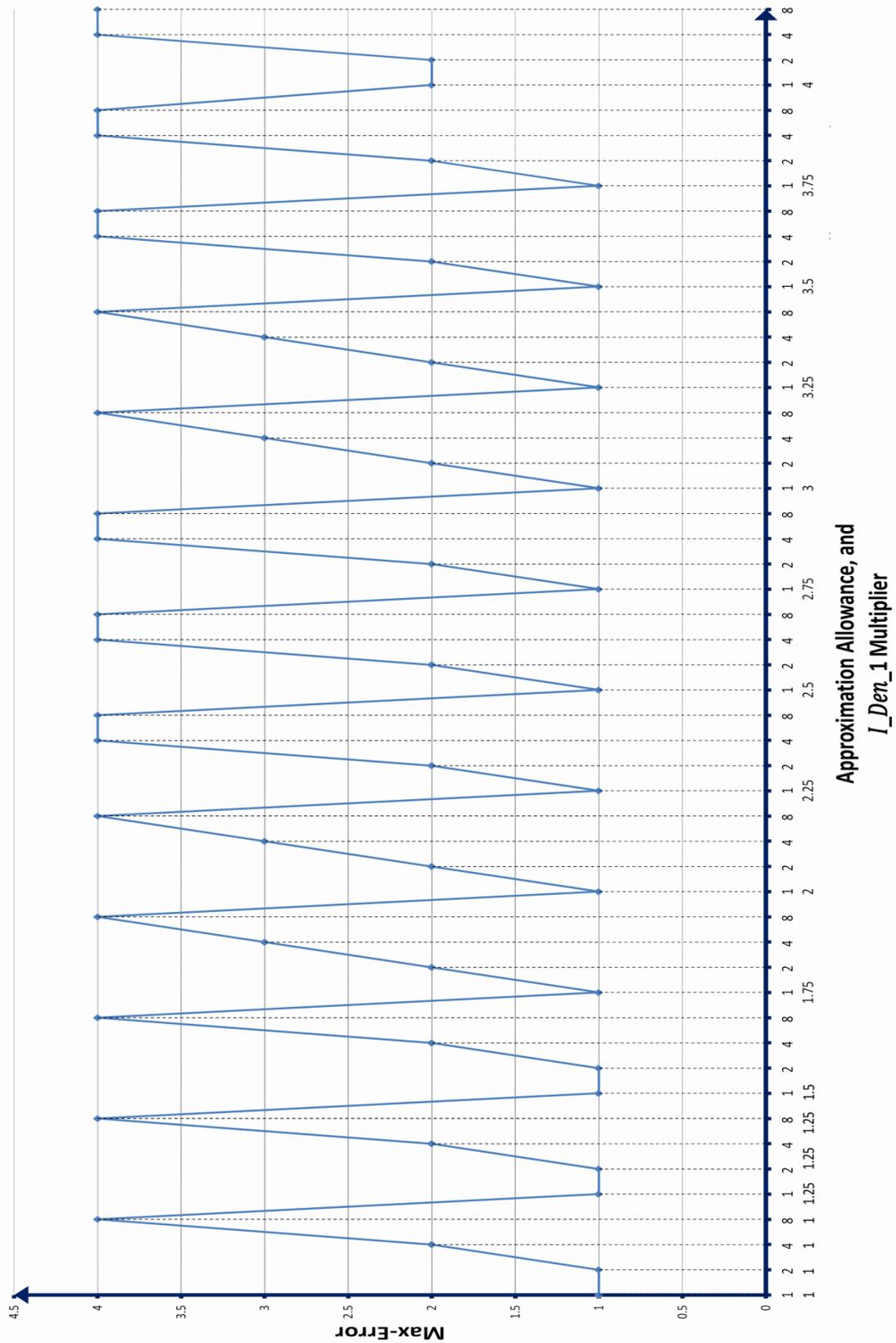


Figure 8-8: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 20% and incremental fuzzy transition step size of 10%, at different approximation allowance multipliers during the second iteration calculations.

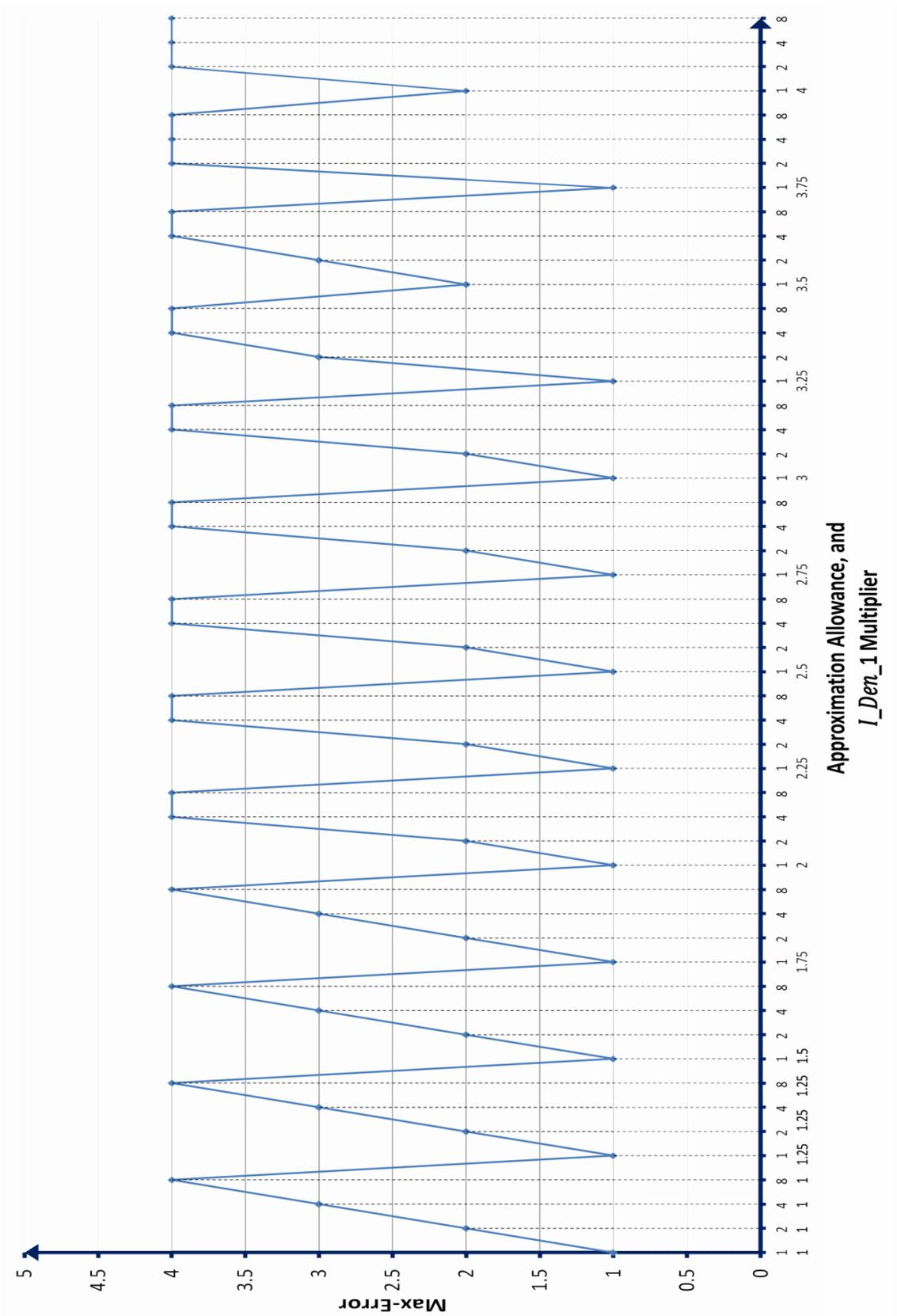


Figure 8-9: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 40% and incremental fuzzy transition step size of 10%, at different approximation allowance multipliers during the second iteration calculations.

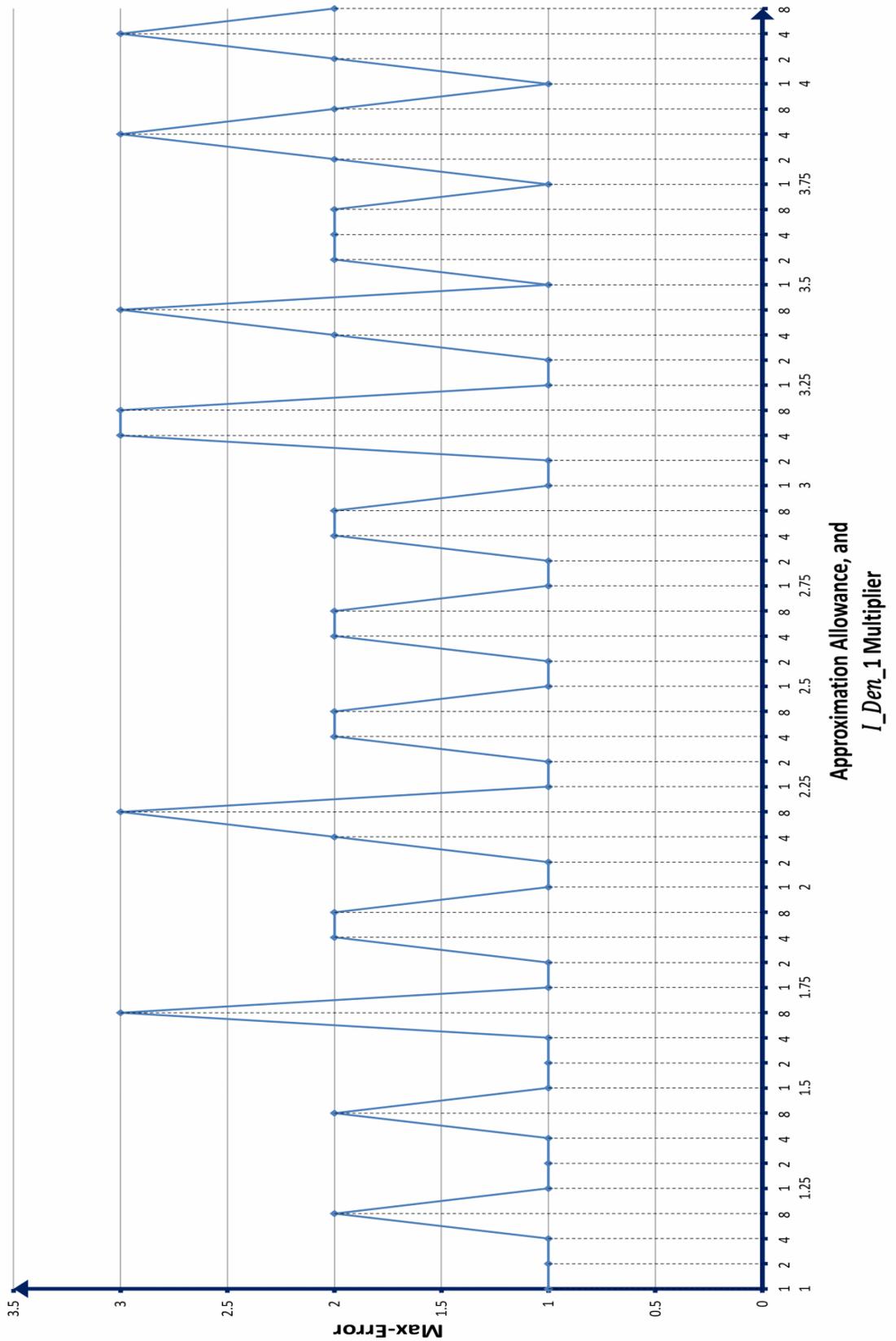


Figure 8-10: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 10% and incremental fuzzy transition step size of 20%, at different approximation allowance multipliers during the second iteration calculations.

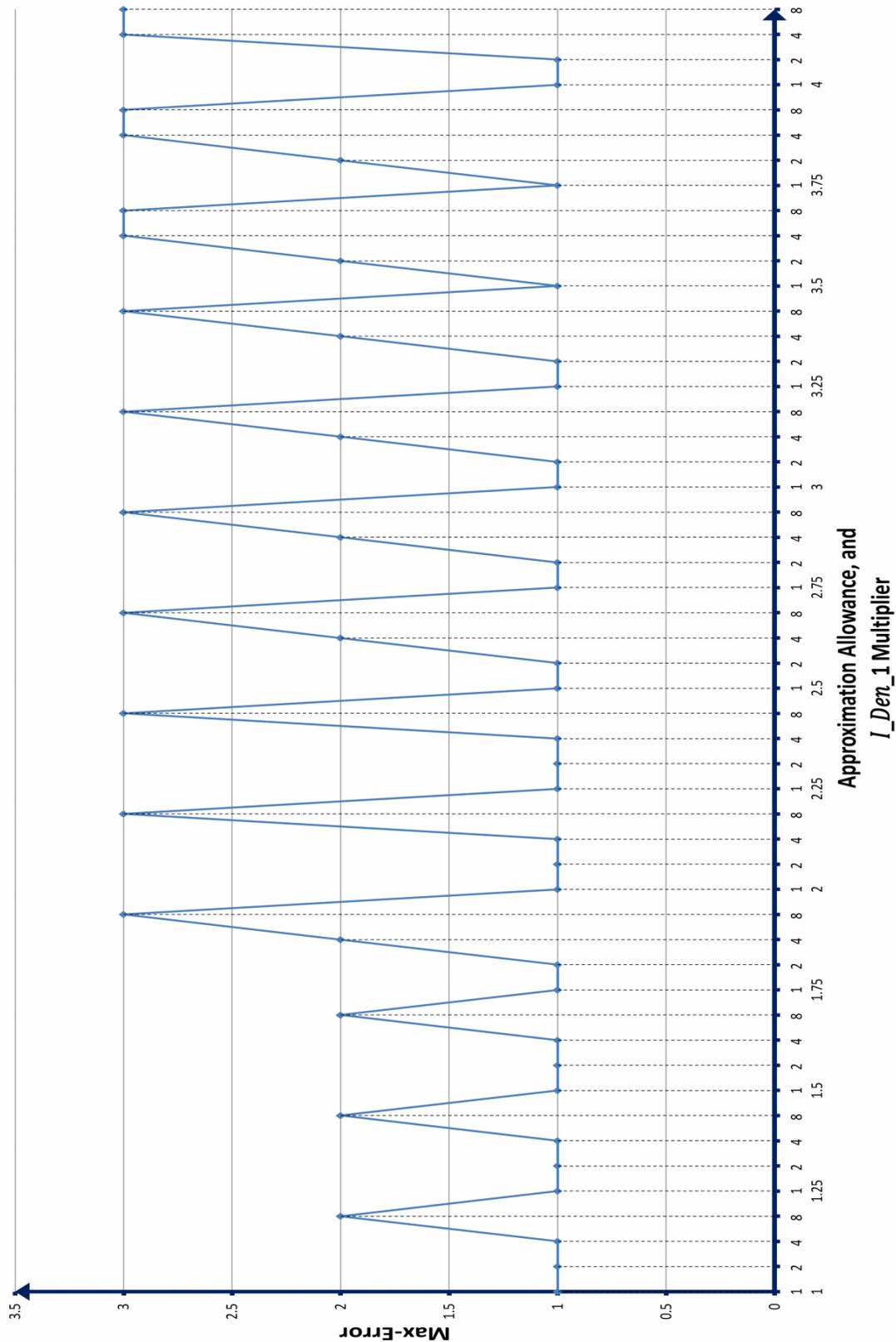


Figure 8-11: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 20% and incremental fuzzy transition step size of 20%, at different approximation allowance multipliers during the second iteration calculations.

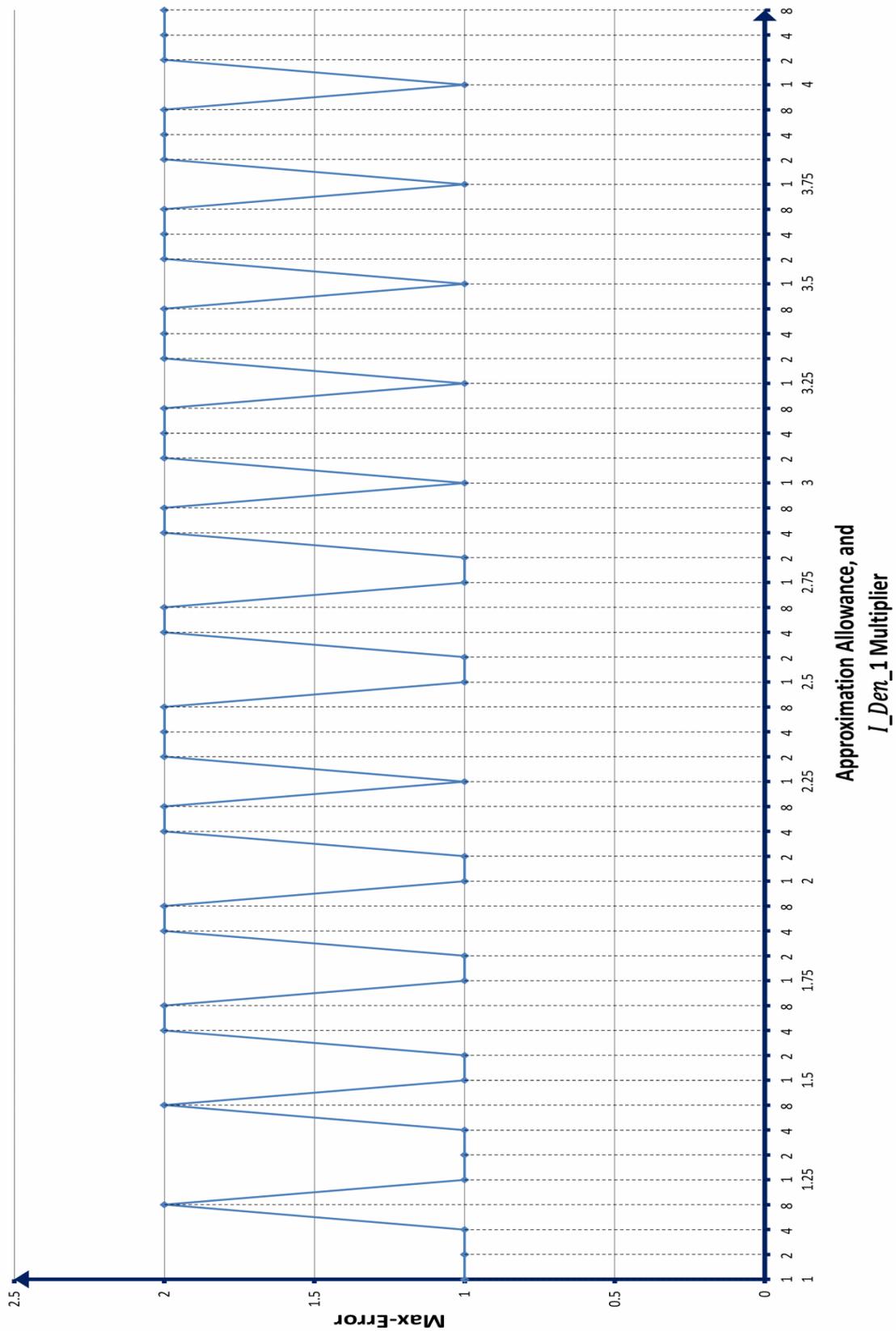


Figure 8-12: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 40% and incremental fuzzy transition step size of 20%, at different approximation allowance multipliers during the second iteration calculations.

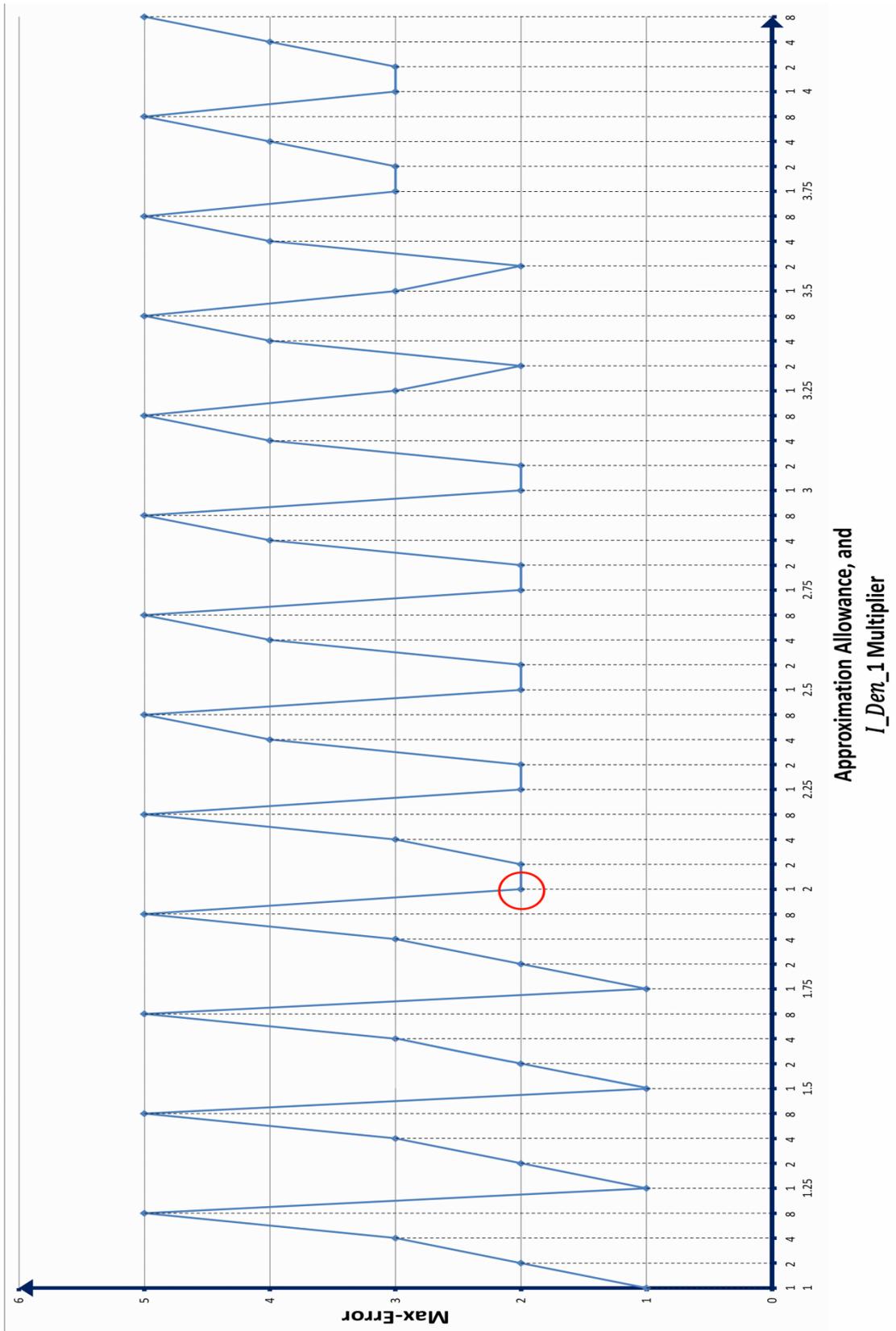


Figure 8-13: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 10% and decrementing fuzzy transition step size of 10%, at different approximation allowance multipliers during the second iteration calculations.

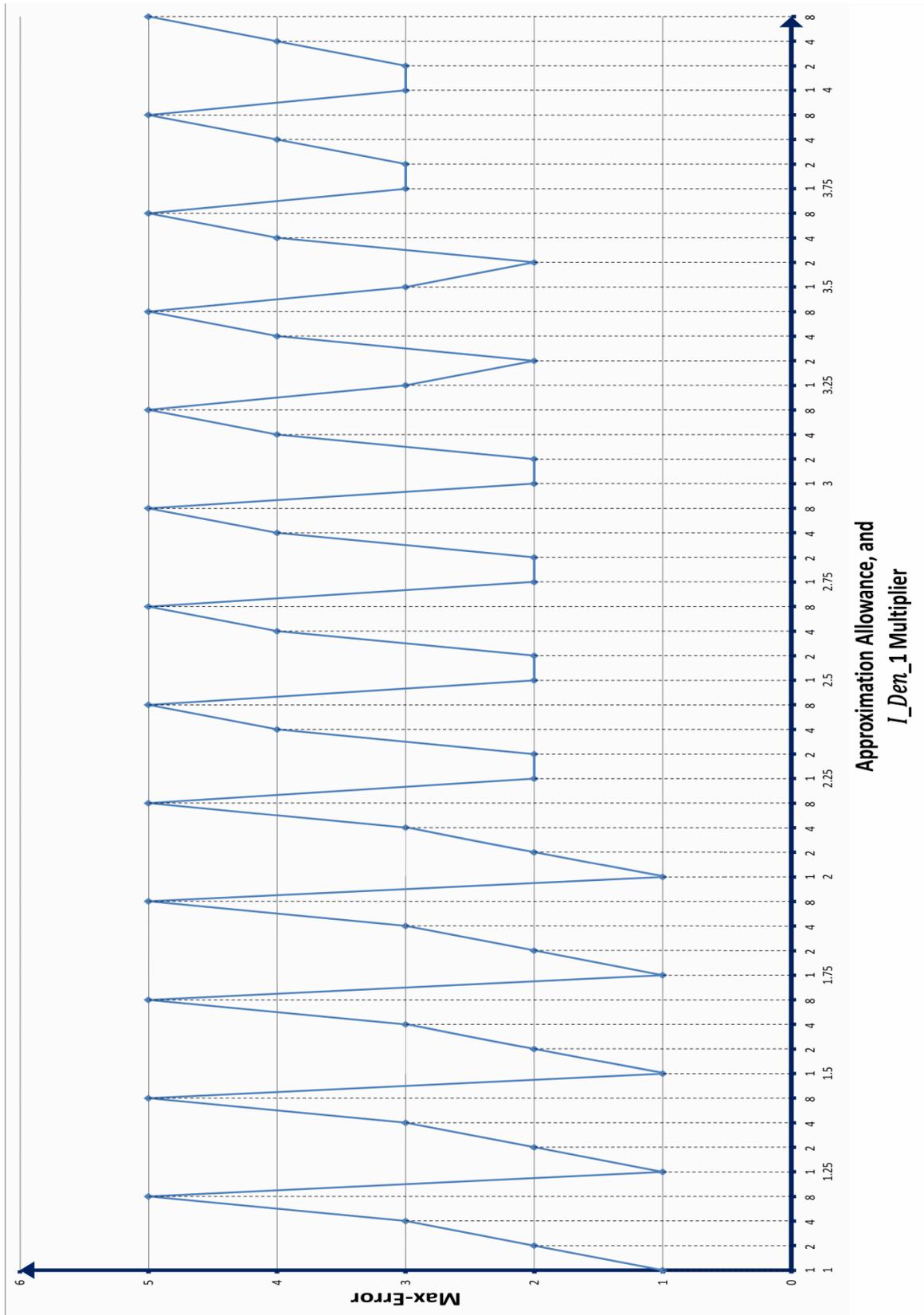


Figure 8-14: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 20% and decrementing fuzzy transition step size of 10%, at different approximation allowance multipliers during the second iteration calculations.

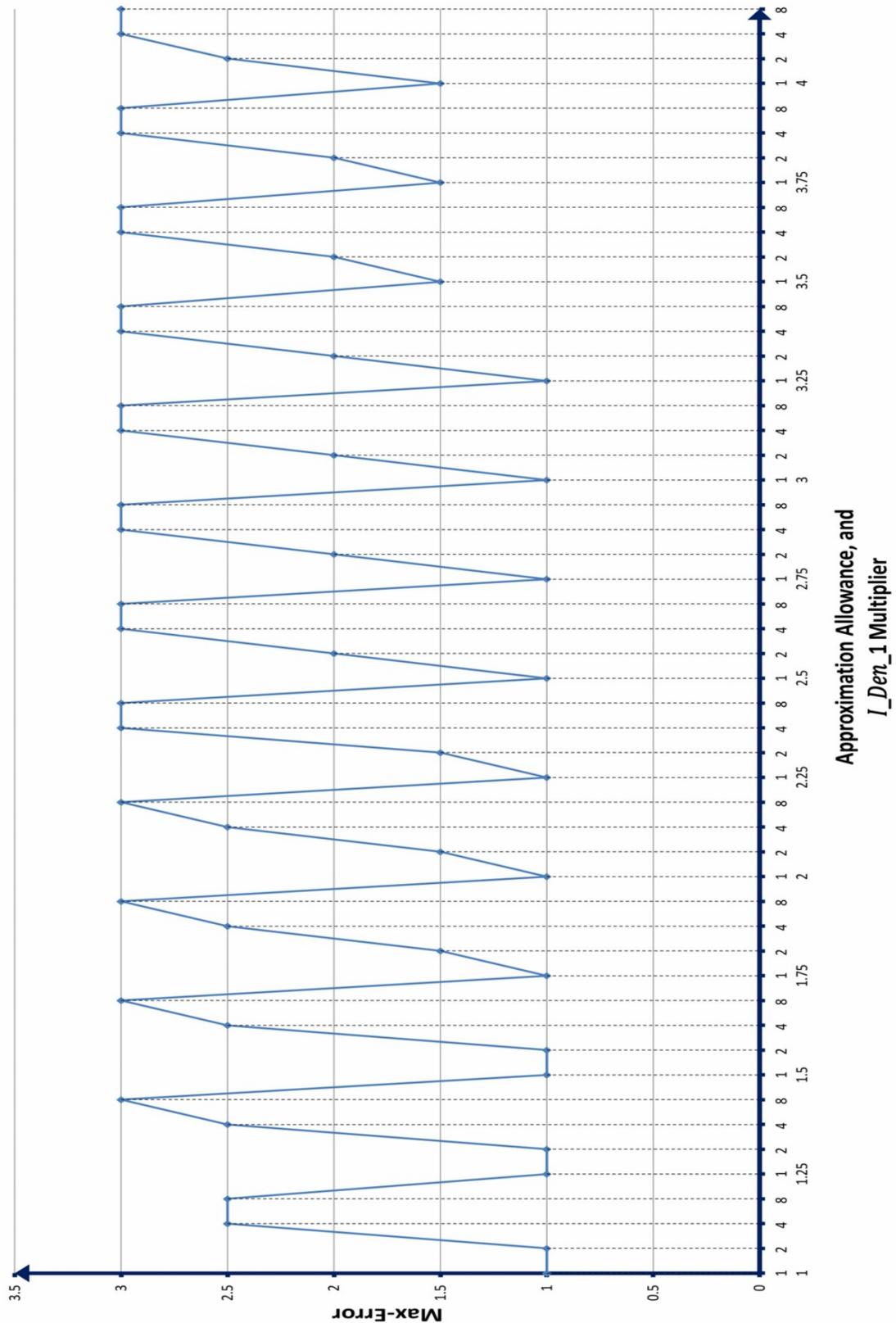


Figure 8-15: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 40% and decrementing fuzzy transition step size of 10%, at different approximation allowance multipliers during the second iteration calculations.

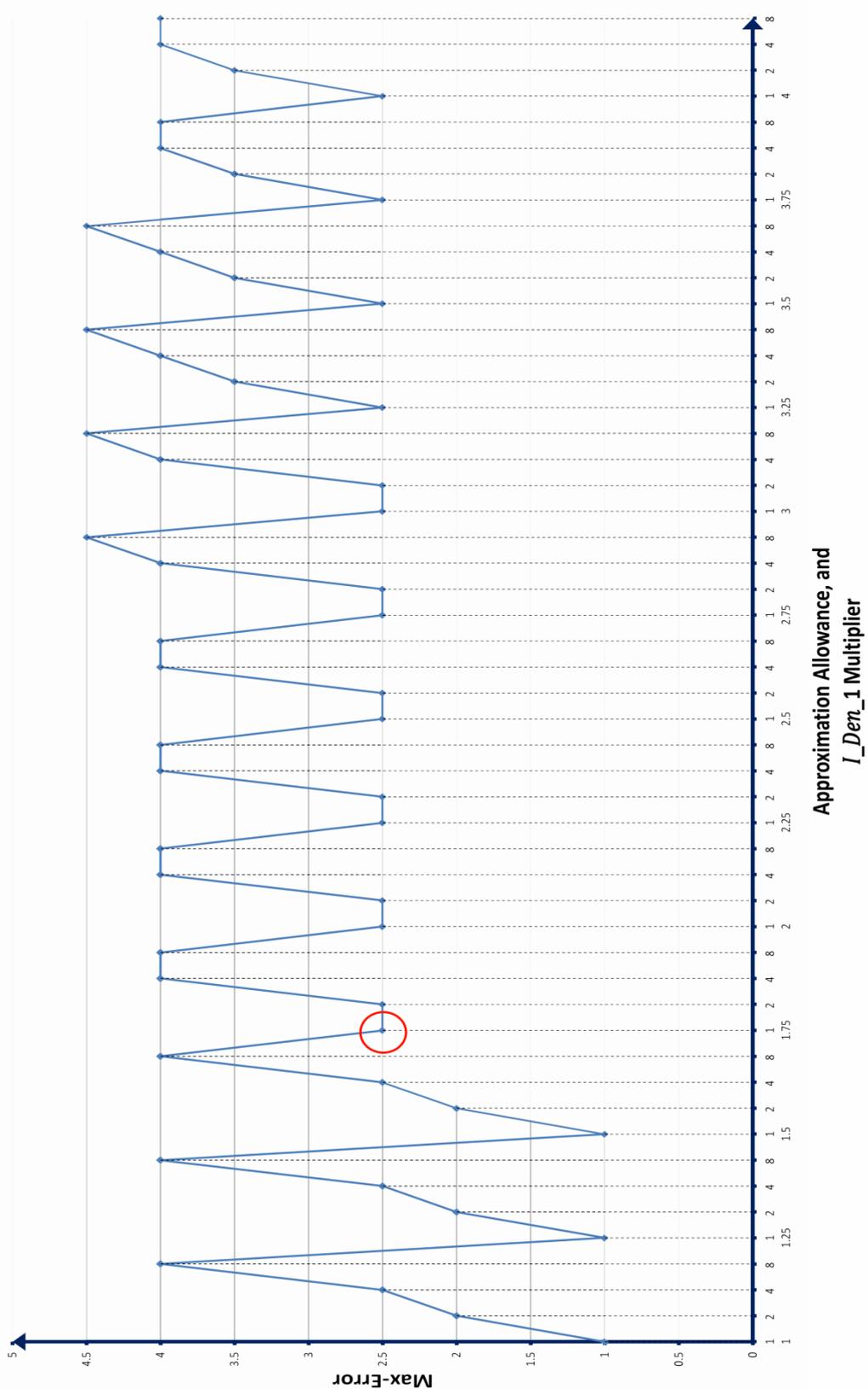


Figure 8-16: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 10% and decrementing fuzzy transition step size of 20%, at different approximation allowance multipliers during the second iteration calculations.

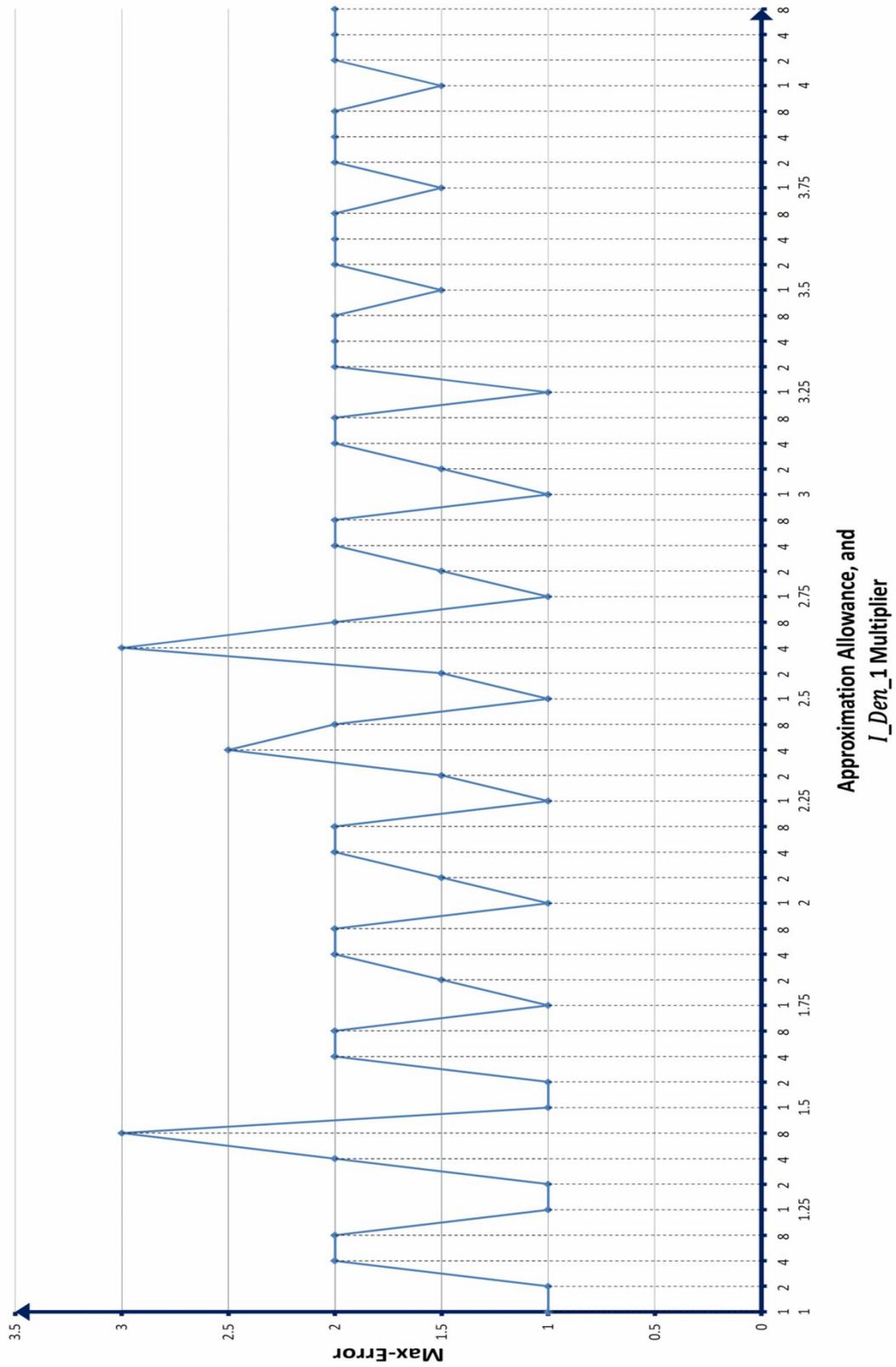


Figure 8-17: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 20% and decrementing fuzzy transition step size of 20%, at different approximation allowance multipliers during the second iteration calculations.

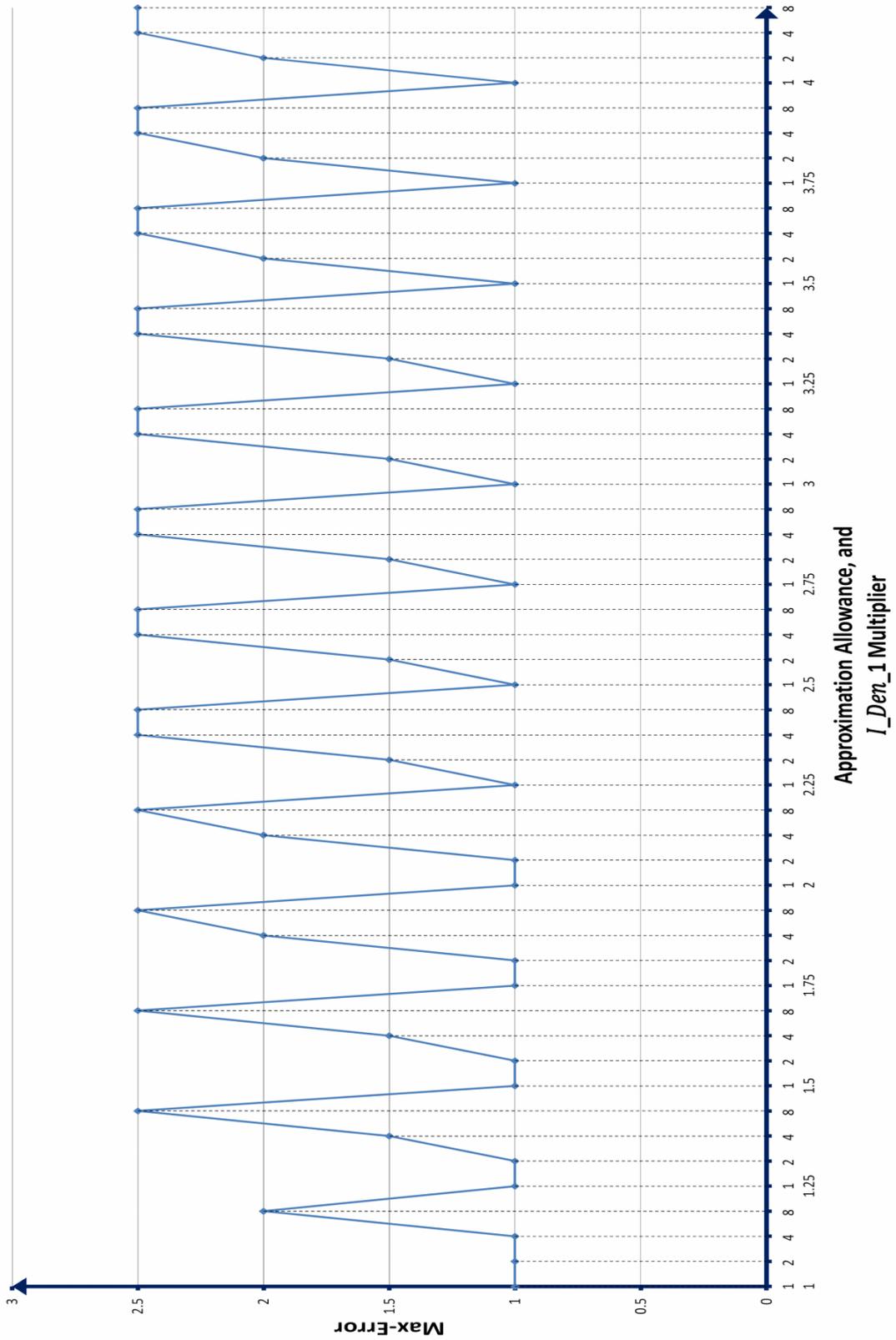


Figure 8-18: Maximum error of the approximated type reduction for using Gaussian IFT2 set having interval width of 40% and decrementing fuzzy transition step size of 20%, at different approximation allowance multipliers during the second iteration calculations.

Appendix-C

The accelerated EKM type reduction routine to locate the left and right uncertainty points, implemented using Java language.

```
int      initialNumeratorL = 0;
int      initialDeNominatorL = 0;
initialDeNominatorL=initialDeNominatorL + LMF[outNo][0]/2; // Using
the first point in the set
for(int i=1;i< MaxHorizontalIndex -1;i++){
    initialNumeratorL=initialNumeratorL+LMF[outNo][i]*i; // For the Left
Point - the LMF is used
    initialDeNominatorL=initialDeNominatorL+ LMF[outNo][i];
}
initialNumeratorL=initialNumeratorL+LMF[outNo][MaxHorizontalIndex -
1]*(MaxHorizontalIndex -1)/2;
initialDeNominatorL=initialDeNominatorL+
LMF[outNo][MaxHorizontalIndex -1]/2;

initialNumeratorR= initialNumeratorL;
initialDeNominatorR=initialDeNominatorL;
//-----calculate the left point using the lower-MF -----
--

newLeftPoint= MaxHorizontalIndex; // Start from the rightmost point
int setIndex=0; // Incremental indexing
do{
    dU=(UMF[outNo][setIndex]-LMF[outNo][setIndex]);
    initialNumeratorL=initialNumeratorL+setIndex * dU;
    initialDeNominatorL=( initialDeNominatorL + dU);
    oldLeftPoint=newLeftPoint;//Save
    newLeftPoint=initialNumeratorL/initialDeNominatorL; // Update
    setIndex++; // Incremental indexing
}while (setIndex< MaxHorizontalIndex && ((newLeftPoint <=
oldLeftPoint)));

// ----- Calculate the right point -----

newRPoint=0;
setIndex= MaxHorizontalIndex;
do{ setIndex--; // Start from the rightmost point
    dU=(UMF[outNo][setIndex]-LMF[outNo][setIndex]);
    initialNumeratorR=initialNumeratorR + setIndex*dU;
```

```
initialDeNominatorR=( initialDeNominatorR+dU);
oldRightPoint=newRPoint;// Save old point
newRPoint= initialNominatorR/initialDeNominatorR; // Update

} while (setIndex>0 &&((newRPoint >= oldRightPoint )));
//-----
int LeftResult= oldLeftPoint;
int rightResult= oldRightPoint;
```

Appendix-D

The trapezoidal rule integration using successive binary search implemented using Java language.

```
/**
 *
 * @param startP Trapezoid rule start point
 * @param endPoint Trapezoid rule end point
 * @param set Trapezoid rule acting set
 * @param outNo Trapezoid rule output set number
 * @return Array contains Numerator@0, denominator@1 using int precision
 */
static public void trapezo(int startP, int endPoint, int set [][], int outNo, int
tolerance){
    int i =startP;
    int i0;
    int dy1;
    int step;
    int offset;
    while (i< endPoint ) { // The (i) Roles over the universe of discourse.
        i0=i; // Save Stat point
        i++;
        dy1=(set[outNo][i] - set[outNo][i0]); // Using Up set for the left side
        step =(endPoint-i); // Math.min(1, (endPoint-i)); // Start step = remaining space
div2
        while(step>1){
            step=step>>1; // Divide by 2
            i=i+step;
            if(((Math.abs(((i-i0)*dy1)-(set[outNo][i]-set[outNo][i0]))>tolerance))) {
                i=i-step;
            }
        }
        int temp1;
        int temp2;
        offset=i-i0;
        num_deNum[1]=num_deNum[1]+
        ((temp1=set[outNo][i0])+(temp2=set[outNo][i]))*offset; // Save the denominator
        num_deNum[0]=num_deNum[0]+(i0*temp1 + i*temp2)*offset; // Save the
        numerator
    }
}
```

Bibliography:

Abdelmajid, B., El Moudden, A. and O.P., C. (2011). Study of the Characteristics and Computation Analysis Results of Electromechanical Systems Models. *International Journal of Information Technology and Computer Science*, 3(5), pp.1–11.

Achtelik, M. et al. (2009). Stereo Vision and Laser Odometry for Autonomous Helicopters in GPS-denied Indoor Environments. In G. R. Gerhart, D. W. Gage, & C. M. Shoemaker, eds. *SPIE Defense, Security, and Sensing*. p. 733219. [online]. Available from: <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=778573> [Accessed May 19, 2014].

ActivMedia Robotics. (2006). *Pioneer 3 Operations Manual*. Version 3. MobileRobots Inc.

Ahmed, M. (2011). *Compliance Control of Robot Manipulator for Safe Physical Human Robot Interaction*. PhD thesis. Örebro University, Sweden. [online]. Available from: <http://oru.diva-portal.org/smash/record.jsf?pid=diva2:388183> [Accessed December 24, 2013].

Aisbett, J. and Rickard, J. (2014). Centroids of Type-1 and Type-2 Fuzzy Sets When Membership Functions Have Spikes. *IEEE Transactions on Fuzzy Systems*, 6706(c), pp.1–10.

Aleksander, I., Gregorio, M. De and França, F. (2009). A brief introduction to Weightless Neural Systems. In *proceedings, European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning*. Bruges (Belgium), pp. 22–24. [online]. Available from: http://pdf.aminer.org/000/271/480/a_modified_spreading_algorithm_for_autoassociation_in_weightless_neural_networks.pdf.

Analog Devices Inc., E. (2005). *The Data Conversion Handbook*. W. Kester, ed. Elsevier-Newnes, Boston.

Astudillo, L., Castillo, O. and Aguilar, L. (2007). Intelligent Control for a Perturbed Autonomous Wheeled Mobile Robot: Type-2 Fuzzy Logic Approach. *Nonlinear Studies*, 14(i), pp.37–48.

Aurenhammer, F. and Klein, R. (1991). Voronoi Diagrams. *ACM Computing Surveys*, 23(3), p.94. [online]. Available from: <http://portal.acm.org/citation.cfm?doid=116873.116880>.

Baklouti, N. and Alimi, A. (2007). Motion Planning in Dynamic and Unknown Environment Using an Interval Type-2 TSK Fuzzy Logic Controller. In *IEEE International Fuzzy Systems Conference*. [online]. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4295647 [Accessed June 3, 2014].

Baklouti, N., John, R. and Alimi, A. (2012). Interval Type-2 Fuzzy Logic Control of Mobile Robots. *Journal of Intelligent Learning Systems and Applications*, 4(November), pp.291–302. [online]. Available from: http://www.scirp.org/fileOperation/download.aspx?path=JILSA20120400007_33802704.pdf&type=journal [Accessed October 14, 2013].

- Barden, J. (2013). *A Modified Clenshaw-Curtis Quadrature Algorithm*. Thesis. WORCESTER POLYTECHNIC INSTITUTE, USA. [online]. Available from: https://www.wpi.edu/Pubs/ETD/Available/etd-042413-133119/unrestricted/A_Modified_Clenshaw-Curtis_Quadrature_Algorithm.pdf.
- Begian, M. (2010). *Systematic Design of Type-2 Fuzzy Logic Systems for Modeling and Control with Applications to Modular and Reconfigurable Robots*. University of Waterloo. [online]. Available from: <https://uwspace.uwaterloo.ca/handle/10012/5301> [Accessed October 14, 2013].
- Braden, B. (1986). The Surveyor's Area Formula. *The College Mathematics Journal*, 17(4), pp.326–337. [online]. Available from: http://www.academia.edu/8176010/The_Surveyors_Area_Formula [Accessed March 19, 2014].
- Burden, R.L. and Faires, J.D. (2011). *Numerical Analysis*. Ninth Edit. Brooks/Cole, Cengage Learning, Boston, USA.
- Carluccio, G. and Albani, M. (2011). Adaptive Numerical Integration Algorithms for the Evaluation of Surface Radiation Integrals. In *Proceedings of the 5th European Conference on Antennas and Propagation (EUCAP)*. IEEE, pp. 2275–2278. [online]. Available from: <http://ieeexplore.ieee.org/abstract/document/5782029/>.
- Castillo, O. (2011). *Type-2 Fuzzy Logic in Intelligent Control Applications*. Verlag Berlin Heidelberg: Springer. [online]. Available from: http://books.google.com/books?hl=en&lr=&id=p4TxcJhBgloC&oi=fnd&pg=PR1&dq=Type-2+Fuzzy+Logic+in+Intelligent+Control+Applications&ots=qOzY7HR00Q&sig=Zvw-glBMG0XKOkBwf6Eq0Xors_Q [Accessed March 23, 2014].
- Castrup, M.S. and Castrup, D.H.T. (2010). *NASA Measurement Quality Assurance Handbook*. Washington DC.
- Chang, Y.H. et al. (2013). Type-2 Fuzzy Formation Control for Collision-Free Multi-Robot Systems. *International Journal of Fuzzy Systems*, 15(4), pp.435–451. [online]. Available from: http://www.ijfs.org.tw/ePublication/2013_paper_4/ijfs13-4-se-7-Chun-Lin-IJFS-27-r1_v3.pdf [Accessed May 28, 2014].
- Chen, C., Chen, S. and Kuo, Y. (2013). Reduction of Interval Type-2 LR Fuzzy Sets. *ieeexplore.ieee.org*. [online]. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6576860 [Accessed March 30, 2014].
- Chen, Y.S. and Yao, L. (2009). Robust Type-2 Fuzzy Control of an Automatic Guided Vehicle for Wall-Following. In *International Conference of Soft Computing and Pattern Recognition*. IEEE, pp. 172–177. [online]. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5370094 [Accessed January 13, 2014].
- Chen, Y.S. and Yao, L. (2011). Type-2 Fuzzy Control of an Automatic Guided Vehicle for Wall-Following. In D. L. Grigorie, ed. *Fuzzy Controllers, Theory and Applications*. INTECH, pp. 172–177. [online]. Available from: <http://www.intechopen.com/books/fuzzy-controllers-theory-and-applications/type-2->
- Chiclana, F. and Zhou, S. (2011). The Type-1 OWA Operator and the Centroid of Type-

- 2 Fuzzy Sets. *European Society for Fuzzy Logic and Technology (EUSFLAT) Conference*, (July). [online]. Available from: http://www.atlantispress.com/php/download_paper.php?id=2323 [Accessed February 14, 2014].
- Chiclana, F. and Zhou, S. (2013). Type-Reduction of General Type-2 Fuzzy Sets: The Type-1 OWA Approach. *International Journal of Intelligent Systems*, 28, pp.505–522. [online]. Available from: <http://onlinelibrary.wiley.com/doi/10.1002/int.21588/full> [Accessed October 14, 2013].
- Cingolani, P. and Alcalá-Fdez, J. (2012). jFuzzyLogic: A Robust and Flexible Fuzzy-Logic Inference System Language Implementation. In *IEEE International Conference on Fuzzy Systems*. Ieee, pp. 1–8. [online]. Available from: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6251215>.
- Collier, D. a. and Meyer, S.M. (2000). An empirical comparison of C, C++, Java, Perl, Python, REXX, and Tcl. *International Journal of Operations & Production Management*, 20(6), pp.705–729.
- Coupland, S. (2007). Type-2 Fuzzy Sets: Geometric Defuzzification and Type-Reduction. *IEEE Symposium on Foundations of Computational Intelligence*, pp.622–629. [online]. Available from: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4233971> [Accessed March 11, 2014].
- Cromme, L.J. (1997). Fixed Point Theorems for Discontinuous Functions and Applications. *Nonlinear Analysis, Theory, Methods & Applications*, 30(3), pp.1527–1534.
- Cruz-Uribe, D. and Neugebauer, C.J. (2003). An Elementary Estimates Proof of Error for the Trapezoidal Rule. *Mathematics Magazine*, 76(4), pp.303–306.
- Cruz-Uribe, D., Neugebauer, C.J. and Lafayette, W. (2002). SHARP ERROR BOUNDS FOR THE TRAPEZOIDAL RULE AND SIMPSON’S RULE. *Journal of Inequalities in Pure and Applied Mathematics*, 3(4), pp.1–22.
- Doncker, E. De, Gupta, A. and Greenwood, G. (1996). Adaptive Integration Using Evolutionary Strategies. *Proceedings of 3rd International Conference on High Performance Computing (HiPC)*, pp.94–99.
- Duran, K., Bernal, H. and Melgarejo, M. (2008). Improved Iterative Algorithm for Computing the Generalized Centroid of an Interval Type-2 Fuzzy Set. *Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS)*, pp.1–5. [online]. Available from: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4531244>.
- Erol, O.K. and Eksin, I. (2006). A new optimization method: Big Bang–Big Crunch. *Advances in Engineering Software*, 37(2), pp.106–111. [online]. Available from: <http://linkinghub.elsevier.com/retrieve/pii/S0965997805000827> [Accessed June 4, 2014].
- Fang, H. et al. (2011). Robust anti-sliding control of autonomous vehicles in presence of lateral disturbances. *Control Engineering Practice*, 19(5), pp.468–478. [online]. Available from: <http://linkinghub.elsevier.com/retrieve/pii/S0967066111000189> [Accessed September 19, 2013].
- Figueroa, J. et al. (2005). A Type-2 Fuzzy Controller for Tracking Mobile Objects in the

- Context of Robotic Soccer Games. *The 14th IEEE International Conference on Fuzzy Systems*, 2(1), pp.359–364. [online]. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1452420 [Accessed May 28, 2014].
- Fitzgibbon, A., Pilu, M. and Fisher, R.B. (1999). Direct Least Square Fitting of Ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5), pp.476–480.
- Fog, A. (2004). *How to optimize for the Pentium family of microprocessors*. [online]. Available from: <http://flsdemo.nrcfoss.au-kbc.org.in/content/docs/Qmail/qmail/bib/2004/fog.pdf> [Accessed June 27, 2014].
- Gander, W. and Gautschi, W. (2000). Adaptive quadrature—revisited. *Bit Numerical Mathematics*, 40(1), pp.84–101.
- Garcia, J.F. (2012). An approximation method for Type Reduction of an Interval Type-2 fuzzy set based on α -cuts. *Proceedings of the Federated Conference on Computer Science and Information Systems*, pp.49–54. [online]. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6354390 [Accessed March 29, 2014].
- George J. Klir and Bo Yuan. (1995). *FUZZY SETS AND FUZZY LOGIC: Theory and Applications*. Prentice Hall PTR, USA. [online]. Available from: <http://digilib.uin-suka.ac.id/7049/> [Accessed March 23, 2014].
- Gomez, J.F. and Jamshidi, M. (2010). Fuzzy Adaptive Control for a UAV. *Journal of Intelligent & Robotic Systems*, 62(2), pp.271–293. [online]. Available from: <http://link.springer.com/10.1007/s10846-010-9445-4> [Accessed September 2, 2013].
- Gonnet, P. (2012). A Review of Error Estimation in Adaptive Quadrature. *ACM Comput. Surv.*, 44(4), p.22:1–22:36. [online]. Available from: <http://arxiv.org/abs/1003.4629>.
- Gonnet, P. (2009). *Adaptive Quadrature Re-Revisited*. PhD thesis. SWISS FEDERAL INSTITUTE OF TECHNOLOGY, Zurich.
- Greenfield, S., Chiclana, F., Coupland, S., et al. (2009). The Collapsing Method of Defuzzification for Discretised Interval Type-2 Fuzzy Sets. *Information Sciences*. [online]. Available from: <http://www.sciencedirect.com/science/article/pii/S0020025508002727> [Accessed October 14, 2013].
- Greenfield, S. et al. (2011). The Sampling Method of Defuzzification for Type-2 Fuzzy Sets : Experimental Evaluation. *Information Sciences*. [online]. Available from: <http://www.sciencedirect.com/science/article/pii/S002002551100627X> [Accessed March 11, 2014].
- Greenfield, S. (2012). *Type-2 Fuzzy Logic: Circumventing the Defuzzification Bottleneck*. PhD thesis. De Montfort University, Leicester, UK. [online]. Available from: <http://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.571317> [Accessed November 8, 2013].
- Greenfield, S., Chiclana, F. and John, R. (2009). The Collapsing Method: Does the Direction of Collapse Affect Accuracy?. *IFSA/EUSFLAT Conf.*, 1, pp.980–985. [online]. Available from: http://www.researchgate.net/publication/221398844_The_Collapsing_Method_Does_the_Direction_of_Collapse_Affect_Accuracy/file/3deec5232e6b41ca22.pdf [Accessed March 12, 2014].

- Greenfield, S., John, R. and Coupland, S. (2005). A Novel Sampling Method for Type-2 Defuzzification. *In Proc. UK Workshop on Computational Intelligence (UKCI)*, pp.120–127. [online]. Available from: <https://dora.dmu.ac.uk/handle/2086/980> [Accessed April 14, 2014].
- Hagras, H. (2004a). A Hierarchical Type-2 Fuzzy Logic Control Architecture for Autonomous Mobile Robots. *IEEE Transactions on Fuzzy Systems*, 12(4), pp.524–539. [online]. Available from: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1321080>.
- Hagras, H. (2004b). A Type-2 Fuzzy Logic Controller For Autonomous Mobile Robots. *IEEE International Conference on Fuzzy Systems (IEEE Cat. No.04CH37542)*, 2, pp.965–970. [online]. Available from: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1375538>.
- Halgamuge, S.K., Runkler, T.A. and Glesner, M. (1996). On the Neural Defuzzification Methods. *Proceedings of IEEE 5th International Fuzzy Systems*, pp.463–469. [online]. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=551785 [Accessed February 26, 2014].
- Hamrawi, H. (2011). *Type-2 Fuzzy Alpha-cuts*. PhD thesis. De Montfort University, Leicester, UK. [online]. Available from: <https://dora.dmu.ac.uk/handle/2086/5137> [Accessed March 21, 2014].
- Hsiao, M.-Y. et al. (2009). Combined Interval Type-2 Fuzzy Kinematic and Dynamic Controls of the Wheeled Mobile Robot with Adaptive Sliding-Mode Technique. *IEEE International Conference on Fuzzy Systems*, pp.706–711. [online]. Available from: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5277375> [Accessed June 3, 2014].
- Hsu, C.-H. and Juang, C.-F. (2012). Continuous Ant Optimized Type-2 Fuzzy Controller for Accurate Mobile Robot Wall-Following Control. *International conference on Fuzzy Theory and Its Applications (iFUZZY2012)*, pp.187–191. [online]. Available from: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6409698>.
- Hu, J. and Li, T. (2014). Cascaded navigation control for agricultural vehicles tracking straight paths. *International Journal of Agricultural and Biological Engineering*, 7(1), pp.36–44. [online]. Available from: <http://ir.sia.cn/handle/173321/14671> [Accessed May 23, 2014].
- Jean-Jacques E. Slotine and Weiping, L. (1991). Fundamentals of Lyapunov Theory. In *Applied Nonlinear Control*. Prentice Hall, p. 46.
- Jean-Jacques Herings, P. et al. (2008). A fixed point theorem for discontinuous functions. *Operations Research Letters*, 36(1), pp.89–93.
- Kang, T., Zhang, H. and Park, G. (2009). Stereo-Vision Based Motion Estimation of a Humanoid Robot for the Ego-Motion Compensation by Type-2 Fuzzy Sets. *IEEE International Symposium on Industrial Electronics, (ISIE)*, pp.1785–1790. [online]. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5213137 [Accessed June 3, 2014].
- Karnik, N. and Mendel, J. (1998). Type-2 Fuzzy Logic Systems: Type-Reduction. *IEEE International Conference on Systems, Man, and Cybernetics*. [online]. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=728199 [Accessed January 31,

2014].

Karnik, N., Mendel, J. and Liang, Q. (1999). Type-2 Fuzzy Logic Systems. *IEEE Transactions on Fuzzy Systems*, 7(6), pp.643–658. [online]. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=811231 [Accessed January 20, 2014].

Karnik, N.N. and Mendel, J.M. (2001). Centroid of a type-2 fuzzy set. *Information Sciences*, 132(1–4), pp.195–220. [online]. Available from: <http://linkinghub.elsevier.com/retrieve/pii/S002002550100069X>.

Kayacan, E. et al. (2013). Towards Agrobots: Trajectory Control of an Autonomous Tractor Using Type-2 Fuzzy Logic Controllers. *IEEE/ASME Transactions on Mechatronics*, pp.1–12. [online]. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6695753 [Accessed June 3, 2014].

Kovacic, Z. and Bogdan, S. (2010). *Fuzzy Controller Design: Theory and Applications*. S. Kovačič, Zdenko. Bogdan, ed. Fort Worth, USA: Taylor & Francis Group and the CRC Press the Academic Division of Informa plc. [online]. Available from: http://books.google.com/books?hl=en&lr=&id=hXyg2cwilugC&oi=fnd&pg=PR5&dq=Fuzzy+Controller+Design+Theory+and+Applications&ots=jFKBZ19Lq2&sig=2Rtkwgur rN_0QYZzLsc1W9T-2uU [Accessed November 25, 2013].

Kumbasar, T. and Hagra, H. (2013). A Type-2 Fuzzy Cascade Control Architecture for Mobile Robots. *IEEE International Conference on Systems, Man, and Cybernetics*, pp.3226–3231. [online]. Available from: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6722303> [Accessed June 3, 2014].

Kuo, Y. and Chen, C. (1998). Generic LR Fuzzy Cells for Fuzzy Hardware Synthesis. *IEEE Transactions on Fuzzy Systems*, 6(2), pp.266–285. [online]. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=669026 [Accessed April 3, 2014].

Leottau, L. and Melgarejo, M. (2010). A Simple Approach for Designing a Type-2 Fuzzy Controller for a Mobile Robot Application. *Fuzzy Information Processing Society (NAFIPS), Annual Meeting of the North American*, pp.1–6. [online]. Available from: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5548418> [Accessed June 3, 2014].

Leottau, L. and Melgarejo, M. (2011). An Embedded Type-2 Fuzzy Controller for a Mobile Robot Application. In *Recent Advances in Mobile Robotics*. p. 20. [online]. Available from: http://cdn.intechopen.com/pdfs/24930/InTech-An_embedded_type_2_fuzzy_controller_for_a_mobile_robot_application.pdf [Accessed October 14, 2013].

Liang, Q. and Mendel, J. (1999). An Introduction to Type-2 TSK Fuzzy Logic Systems. *IEEE International Fuzzy Systems Conference Proceedings*, pp.1534–1539. [online]. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=790132 [Accessed June 11, 2014].

Linda, O. and Manic, M. (2012). Monotone Centroid Flow Algorithm for Type Reduction of General Type-2 Fuzzy Sets. *IEEE Transactions on Fuzzy Systems*, 20(5), pp.805–819. [online]. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6135785 [Accessed March 17, 2014].

- Linda, O. and Manic, M. (2011). Uncertainty Modeling with Interval Type-2 Fuzzy Logic Systems in Mobile Robotics. *IECON - 37th Annual Conference of the IEEE Industrial Electronics Society*, (1), pp.2441–2446. [online]. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6119692 [Accessed June 3, 2014].
- Linda, O., Manic, M. and Member, S. (2010). Importance Sampling Based Defuzzification for General Type-2 Fuzzy Sets. *International Conference on Fuzzy Systems*, (1), pp.18–23. [online]. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5584256 [Accessed March 26, 2014].
- LIU, F. (2008). An efficient centroid type-reduction strategy for general type-2 fuzzy logic system. *Information Sciences*, 178(9), pp.2224–2236. [online]. Available from: <http://linkinghub.elsevier.com/retrieve/pii/S0020025507005385> [Accessed October 14, 2013].
- Liu, X., Mendel, J.M. and Wu, D. (2012). Study on enhanced Karnik–Mendel algorithms: Initialization explanations and computation improvements. *Information Sciences*, 184(1), pp.75–91. [online]. Available from: <http://linkinghub.elsevier.com/retrieve/pii/S0020025511003811> [Accessed January 28, 2014].
- Liu, Z., Zhang, Y. and Wang, Y. (2007). A Type-2 Fuzzy Switching Control System for Biped Robots. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, 37(6), pp.1202–1213. [online]. Available from: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4343995>.
- Lucas, L.L.A. LA, Centeno, T.M. and Delgado, M.R. (2007). General Type-2 Fuzzy Inference Systems: Analysis, Design and Computational Aspects. *IEEE Transactions on Fuzzy Systems*, (5), pp.1–6. [online]. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4295522 [Accessed March 26, 2014].
- Lynch, C. et al. (2006). Using Uncertainty Bounds in the Design of an Embedded Real-Time Type-2 Neuro-Fuzzy Speed Controller for Marine Diesel Engines. *IEEE International Conference on Fuzzy Systems*, pp.7217–7224.
- Mamdani, E. (1977). Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis. *IEEE Transactions on Computers*, C-2A,(12), pp.1182–1191. [online]. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1674779 [Accessed February 23, 2014].
- Maowen, N. (2011). *Analysis and Applications of the KM Algorithm in Type-2 Fuzzy Logic Control and Decision Making*. PhD thesis. NATIONAL UNIVERSITY OF SINGAPORE. [online]. Available from: <http://scholarbank.nus.sg/handle/10635/32462> [Accessed October 14, 2013].
- Mbede, J. and Melingui, A. (2012). zSlices Based Type-2 Fuzzy Motion Control for Autonomous Robotino Mobile Robot. *International Conference on Mechatronics and Embedded Systems and Applications (MESA)*, pp.63–68. [online]. Available from: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6275538> [Accessed June 3, 2014].
- McNeill, J. a. et al. (2011). All-Digital Background Calibration of a Successive Approximation ADC Using the $\Sigma\Delta$ Split ADC Architecture. *IEEE*

- Transactions on Circuits and Systems I: Regular Papers*, 58(10), pp.2355–2365.
- Melingui, A. and Chettibi, T. (2013). Adaptive Navigation of an Omni-drive Autonomous Mobile Robot in Unstructured Dynamic Environments. *Proceeding of the IEEE International Conference on Robotics and Biomimetics (ROBIO)*, (December), pp.1924–1929. [online]. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6739750 [Accessed June 3, 2014].
- Mendel, J.M. (2007). Advances in Type-2 Fuzzy Sets and Systems. *Information Sciences*, 177(1), pp.84–110. [online]. Available from: <http://linkinghub.elsevier.com/retrieve/pii/S0020025506001356> [Accessed August 6, 2013].
- Mendel, J.M. (2000). Interval Type-2 Fuzzy Logic Systems: Theory and Design. *IEEE Transactions on Fuzzy Systems*, 8(5), pp.535–550. [online]. Available from: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=873577>.
- Mendel, J.M. (2002). Uncertainty Bounds and Their Use in the Design of Interval Type-2 Fuzzy Logic Systems. *IEEE Transactions on Fuzzy Systems*, 10(5), pp.622–639. [online]. Available from: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1038818>.
- Mendel, J.M., John, R. and Liu, F. (2006). Interval Type-2 Fuzzy Logic Systems Made Simple. *IEEE Transactions on Fuzzy Systems*, 14(6), pp.808–821. [online]. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4016089 [Accessed October 14, 2013].
- Mendel, J.M. and John, R.I.B. (2002). Type-2 Fuzzy Sets Made Simple. *IEEE Transactions on Fuzzy Systems*, 10(2), pp.117–127. [online]. Available from: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=995115>.
- Mendel, J.M. and Liu, F. (2008). On New Quasi-Type-2 Fuzzy Logic Systems. *IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence)*, pp.354–360. [online]. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4630390 [Accessed April 14, 2014].
- Mendel, J.M., Liu, F. and Zhai, D. (2009). α -Plane Representation for Type-2 Fuzzy Sets: Theory and Applications. *IEEE Transactions on Fuzzy Systems*, 17(5), pp.1189–1207. [online]. Available from: <http://dl.acm.org/citation.cfm?id=1771963> [Accessed January 13, 2014].
- Mendel, J.M.J. (1995). Fuzzy Logic Systems for Engineering: A Tutorial. *Proceedings of the IEEE*, 83(3), pp.345–377. [online]. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=364485 [Accessed February 23, 2014].
- Mohammad Khansari-Zadeh, S. and Billard, A. (2014). Learning Control Lyapunov Function to Ensure Stability of Dynamical System-based Robot Reaching Motions. *Robotics and Autonomous Systems*, 62(6), pp.752–765. [online]. Available from: <http://dx.doi.org/10.1016/j.robot.2014.03.001>.
- Mucientes, M. et al. (2007). Design of a Fuzzy Controller in Mobile Robotics Using Genetic Algorithms. *Applied Soft Computing*, 7(2), pp.540–546. [online]. Available from: <http://linkinghub.elsevier.com/retrieve/pii/S1568494606000731> [Accessed

October 16, 2014].

Nie, M. and Tan, W. (2012). Analytical Structure and Characteristics of Symmetric Karnik–Mendel Type-Reduced Interval Type-2 Fuzzy PI and PD Controllers. *IEEE Transactions on Fuzzy Systems*, 20(3), pp.416–430. [online]. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6064887 [Accessed November 15, 2013].

Nie, M. and Tan, W.W. (2008). Towards an Efficient Type-Reduction Method for Interval Type-2 Fuzzy Logic Systems. *IEEE International Conference on Fuzzy Systems*, 2, pp.1425–1432.

Nurmaini, S. and Hashim, S. (2009). Motion Planning in Unknown Environment Using an Interval Fuzzy Type-2 and Neural Network Classifier. *International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSAs)*, pp.50–55. [online]. Available from: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5069917> [Accessed June 3, 2014].

Nurmaini, S., Hashim, S. and Jawawi, D. (2009). Environmental Recognition Using RAM-Network Based Type-2 Fuzzy Neural for Navigation of Mobile Robot. *2009 International Conference on Computer and Automation Engineering*, pp.296–301. [online]. Available from: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4804536> [Accessed May 26, 2014].

Nurmaini, S. and Hashim, S.Z.M. (2008). An Embedded Fuzzy Type-2 Controller Based Sensor Behavior for Mobile Robot. *2008 Eighth International Conference on Intelligent Systems Design and Applications*, pp.29–34. [online]. Available from: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4696302> [Accessed November 15, 2013].

Nurmaini, S. and Tutuko, B. (2011). International Conference on Electrical Engineering and Informatics. *International Conference on Electrical Engineering and Informatics (ICEEI)*, 17(July), pp.1–7. [online]. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6021703 [Accessed June 3, 2014].

Nurmaini, S., Zaiton, S. and Norhayati, D. (2009). An Embedded Interval Type-2 Neuro-Fuzzy Controller for Mobile Robot Navigation. In *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*. pp. 4315–4321. [online]. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5346800 [Accessed June 3, 2014].

Ogawa, T. et al. (2011). SAR ADC Algorithm with Redundancy Sampling Quantization in time domain. *IEICE Trans on Fundamentals*, E93–A(2), pp.1–32.

Rao, A.S.R.S. (2007). A Mid-Point Theorem for The U Type Shape of Functions. *Bulletin of Cybernetics and Informatics*, (1), p.8. [online]. Available from: <http://arxiv.org/abs/math.CO/0511365>.

Ren, Q., Baron, L. and Balazinski, M. (2006). Type-2 Takagi-Sugeno-Kang Fuzzy Logic Modeling using Subtractive Clustering. *Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS)*, pp.120–125. [online]. Available from: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4216787>.

- Rodríguez-Castaño, A., Heredia, J. and Ollero, A. (2000). Fuzzy Path Tracking and Position Estimation of Autonomous Vehicles Using Differential GPS. *Mathware & Soft Computing*, 1(7), pp.257–264. [online]. Available from: <http://ic.ugr.es/Mathware/index.php/Mathware/article/view/139> [Accessed May 23, 2014].
- Runkler, T. (1996). Extended Defuzzification Methods and Their Properties. *Proceedings of IEEE 5th International Fuzzy Systems*, 3(11), pp.694–700. [online]. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=551822 [Accessed February 26, 2014].
- Runkler, T.A. (1997). Selection of Appropriate Defuzzification Methods Using Application Specific Properties. *IEEE Transactions on Fuzzy Systems*, 5(1), pp.72–79. [online]. Available from: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=554449>.
- Saade, J. and Diab, H. (2004). Defuzzification Methods and New Techniques for Fuzzy Controllers. *Iranian Journal of Electrical and Computer Engineering*, 3(2). [online]. Available from: http://www.sid.ir/EN/VEWSSID/J_pdf/89020040213.pdf [Accessed February 27, 2014].
- Saade, J.J. and Diab, H.B. (2000). Defuzzification Techniques for Fuzzy Controllers. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 30(1), pp.223–229. [online]. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=826965 [Accessed February 27, 2014].
- Saberi, M. and Lotfi, R. (2014). Segmented Architecture for Successive Approximation Analog-to-Digital Converters. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 22(3), pp.593–606.
- Salaken, S.M. et al. (2015). Linear Approximation of Karnik-Mendel Type Reduction Algorithm. *IEEE International Conference on Fuzzy Systems*, Novem.
- Saleh, J.A. et al. (2009). Soft Computing Techniques in Intelligent Wall Following Control for a Car-like Mobile Robot. *3rd International Conference on Signals, Circuits and Systems (SCS)*, pp.1–6. [online]. Available from: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5412440>.
- Shi, Y. (2009). *A Deep Study of Fuzzy Implications*. PhD thesis. Ghent University, Belgium. [online]. Available from: <https://biblio.ugent.be/publication/758878> [Accessed January 9, 2014].
- Sidhu, S., Saini, J. and Khosla, A. (2012). Interval Type-2 Fuzzy System for Autonomous Navigational Control of Non-Holonomic Vehicles. *International Journal of Information Technology and Knowledge Management*, 5(1), pp.195–200. [online]. Available from: [http://www.csjournals.com/IJITKM/PDF 5-1/Article_39.pdf](http://www.csjournals.com/IJITKM/PDF%205-1/Article_39.pdf) [Accessed May 26, 2014].
- Skulavik, T. et al. (2013). The Defuzzification Methods Influence on Fuzzy Control of Nuclear Reactor. *International Symposium on Computational and Business Intelligence*, pp.119–122. [online]. Available from: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6724336> [Accessed February 26, 2014].

- Sugeno, M. and Kang, G.. (1988). Structure Identification of Fuzzy Model. *Fuzzy Sets and Systems*, 28(1), pp.15–33. [online]. Available from: <http://www.sciencedirect.com/science/article/pii/0165011488901133> [Accessed November 19, 2013].
- Takagi, T. and Sugeno, M. (1985). Fuzzy Identification of Systems and Its Applications to Modeling and Control. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15(1), pp.116–132. [online]. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6313399 [Accessed June 11, 2014].
- Theodoridis, T. and Hu, H. (2012). Toward Intelligent Security Robots: A Survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6), pp.1219–1230.
- Thiele, H. (1994). On T-quantifiers and S-quantifiers. In *Proceedings of 24th International Symposium on Multiple-Valued Logic (ISMVL)*. IEEE Comput. Soc. Press, pp. 264–269. [online]. Available from: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=302192>.
- Trabia, M., Shi, L. and Hodge, N. (2006). A Fuzzy Logic Controller for Autonomous Wheeled Vehicles. *University of Nevada, Las Vegas, Publications of Mechanical Engineering Faculty: Mobile Robots, Moving Intelligence*, pp.175–200. [online]. Available from: http://digitalscholarship.unlv.edu/me_fac_articles/36/ [Accessed May 23, 2014].
- Ulu, C., Guzelkaya, M. and Eksin, I. (2013). A Closed Form Type Reduction Method for Piecewise Linear Interval Type-2 Fuzzy Sets. *International Journal of Approximate Reasoning*, 54(9), pp.1421–1433. [online]. Available from: <http://linkinghub.elsevier.com/retrieve/pii/S0888613X13001552> [Accessed March 17, 2014].
- Wagner, C. and Hagrais, H. (2010). Toward General Type-2 Fuzzy Logic Systems Based on zSlices. *IEEE Transactions on Fuzzy Systems*, 18(4), pp.637–660. [online]. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5428821 [Accessed January 13, 2014].
- Wagner, C. and Hagrais, H. (2009). zSlices based General Type-2 FLC for the Control of Autonomous Mobile Robots in Real World Environments. *IEEE International Conference on Fuzzy Systems*, (3). [online]. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5277383 [Accessed June 3, 2014].
- Wang, G. and Wang, H. (2001). Non-Fuzzy Versions of Fuzzy Reasoning in Classical Logics. *Information Sciences*, 138(1–4), pp.211–236. [online]. Available from: <http://linkinghub.elsevier.com/retrieve/pii/S0020025501001311>.
- Wu, D. and Mendel, J. (2009). Enhanced Karnik–Mendel Algorithms. *IEEE Transactions on Fuzzy Systems*, 17(4), pp.923–934. [online]. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4505357 [Accessed January 31, 2014].
- Wu, D. and Mendel, J.M. (2007). Enhanced Karnik–Mendel Algorithms for Interval Type-2 Fuzzy Sets and Systems. *Fuzzy Information Processing Society, Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS)*, pp.184–189. [online]. Available from: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4271057>.

- Wu, D. and Nie, M. (2011). Comparison and Practical Implementation of Type-Reduction Algorithms for Type-2 Fuzzy Sets and Systems. *IEEE International Conference on Fuzzy Systems*, pp.2131–2138. [online]. Available from: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6007317> [Accessed March 26, 2014].
- Wu, D. and Tan, W. (2005). Computationally Efficient Type-Reduction Strategies for a Type-2 Fuzzy Logic Controller. *IEEE International Conference on Fuzzy Systems*, pp.353–358. [online]. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1452419 [Accessed March 12, 2014].
- Xue, J., Zhang, L. and Grift, T. (2012). Variable Field-of-View Machine Vision Based Row Guidance of an Agricultural Robot. *Computers and Electronics in Agriculture*, 84, pp.85–91. [online]. Available from: <http://www.sciencedirect.com/science/article/pii/S016816991200049X> [Accessed May 23, 2014].
- Yager, R.R. (1988). On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decision Making. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1), pp.183–190. [online]. Available from: <http://dl.acm.org/citation.cfm?id=46931.46950>.
- Yaonan, W. et al. (2011). Autonomous Mobile Robot Navigation System Designed in Dynamic Environment Based on Transferable Belief Model. *Measurement, Elsevier Ltd.*, 44(8), pp.1389–1405. [online]. Available from: <http://linkinghub.elsevier.com/retrieve/pii/S0263224111001540> [Accessed November 25, 2013].
- Yeh, C.-Y., Jeng, W.-H.R. and Lee, S.-J. (2011). An Enhanced Type-Reduction Algorithm for Type-2 Fuzzy Sets. *IEEE Transactions on Fuzzy Systems*, 19(2), pp.227–240. [online]. Available from: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5638621>.
- Zadeh, L. (1965). Fuzzy Sets. *Information and Control*. [online]. Available from: <http://www.sciencedirect.com/science/article/pii/S001999586590241X> [Accessed December 3, 2013].
- Zadeh, L.A. (1975a). The Concept of a Linguistic Variable and its Application to Approximate Reasoning--II. *Information Sciences*, 8, pp.199–249. [online]. Available from: <http://www.eecs.berkeley.edu/~zadeh/papers/The Concept of a Linguistic Variable and its Applications to A> [Accessed January 23, 2014].
- Zadeh, L.A. (1975b). The Concept of a Linguistic Variable and its Application to Approximate Reasoning-I. *Information Sciences*, 8(3), pp.199–249. [online]. Available from: <http://linkinghub.elsevier.com/retrieve/pii/0020025575900365>.
- Zeloufi, M., Dzahini, D. and Rarbi, F.E. (2015). A 12bits 40MSPS SAR ADC With a Redundancy Algorithm and Digital Calibration for The ATLAS- LArg Calorimeter Readout. *Recent Advances in Circuits*, pp.129–132.
- Zhang, Z., Zhang, R. and Liu, X. (2008). Research on Hierarchical Fuzzy Behavior Learning of Autonomous Robot. *2008 International Conference on Internet Computing in Science and Engineering*, pp.43–46. [online]. Available from: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4548232> [Accessed

June 3, 2014].

Zhao, P. et al. (2012). Design of a Control System for an Autonomous Vehicle Based on Adaptive-PID. *International Journal of Advanced Robotic Systems*, p.1. [online].

Available from:

http://www.intechopen.com/journals/international_journal_of_advanced_robotic_systems/design-of-a-control-system-for-an-autonomous-vehicle-based-on-adaptive-pid [Accessed May 23, 2014].

Zhou, S.-M. et al. (2008). Type-1 OWA Operators for Aggregating Uncertain Information With Uncertain Weights Induced by Type-2 Linguistic Quantifiers. *Fuzzy Sets and Systems*, 159(24), pp.3281–3296. [online]. Available from:

<http://linkinghub.elsevier.com/retrieve/pii/S0165011408003321> [Accessed February 3, 2014].

Zhou, S. and Chiclana, F. (2009). Alpha-Level Aggregation: A Practical Approach to Type-1 OWA Operation for Aggregating Uncertain Information with Applications to Breast Cancer Treatments. *IEEE Transactions on Knowledge and Data Engineering*, pp.1–15.

Zhu, C. et al. (2012). Implementation of Autonomous Navigation Based on Cloud Model for Robot. *9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pp.2381–2384. [online]. Available from:

<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6234062>.

Zhuang, Y. et al. (2013). 3-D-Laser-Based Scene Measurement and Place Recognition for Mobile Robots in Dynamic Indoor Environments. *IEEE Transactions on Instrumentation and Measurement*, 62(2), pp.438–450. [online]. Available from:

<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6305475>.

Zimmermann, H. (1991). *Fuzzy Set Theory - and Its Applications, Second, Revised Edition*. New York: Springer Science. [online]. Available from:

<http://link.springer.com/content/pdf/10.1007/978-94-015-7949-0.pdf> [Accessed April 3, 2014].