The University of Jordan

Authorization Form

I, Tagreed Mohammed Al-Jarrah, authorize the University of Jordan to supply copies of my Thesis to libraries or establishments or individuals on request, according to the University of Jordan regulations.

Signature: Tagreed

Date: 1/12/2010

الجامعة الأردنيـــــة
كلية الدراسات العليا

التاريــخ: ١ / ١٢ / ٢٠١٠

نموذج رقم (١٨)
اقرار والتزام بالمعايير الأخلاقية والأمانة العلمية
وقوانين الجامعة الأردنية وأنظمتها وتعليماتها
لطلبة الماجستير

أنا الطالبة: تغريد عاطف محمد الجراح الرقم الجامعي: (8070842)
تخصص: هندسة الميكانيك الكليـــــــة: الهندسة والتكنولوجيا

عنوان الرسالة: OPTIMAL FUZZY PID CONTROLLER

اعلن بأنني قد التزمت بقوانين الجامعة الأردنية وأنظمتها وتعليماتها وقراراتها السارية المفعول المتعلقة باعداد رسائل الماجستير عندما قمت شخصيا" باعداد رسالتي وذلك بما ينسجم مع الأمانة العلمية وكافة المعايير الأخلاقية المتعارف عليها في كتابة الرسائل العلمية. كما أنني أعلن بأن رسالتي هذه غير منقولة أو مستلة من رسائل أو كتب أو أبحاث أو أي منشورات علمية تم نشرها أو تخزينها في أي وسيلة اعلامية، وتأسيسا" على ما تقدم فانني أتحمل المسؤولية بأنواعها كافة فيما لو تبين غير ذلك بما فيه حق مجلس العمداء في الجامعة الأردنية بالغاء قرار منحي الدرجة العلمية التي حصلت عليها وسحب شهادة التخرج مني بعد صدورها دون أن يكون لي أي حق في التظلم أو الاعتراض أو الطعن بأي صورة كانت في القرار الصادر عن مجلس العمداء بهذا الصدد.

توقيع الطالب: .... التاريخ: ١ / ١٢ / ٢٠١٠

تعتمد كلية الدراسات العليا
هذه النسخة من الرسالــــة
التوقيع.......... التاريخ..........

الجامعة الأردنيــة
كلية الدراسات العليا

التاريخ: / /

نموذج رقم (٢٧)
تسليم رسالة ماجستير جامعية للمكتبة

الدكتور مدير المكتبـة

تحية طيبة وبعد،،،

لقد ناقش الطالب / الطالبة: تحرير عاطف الجراح ورقمه الجامعي: 8070842

تخصص الماجستير: الهندسة الميكانيكية

يوم: الخميس الموافق: ٤ / ١١ / ٢٠١٠ ، وكانت النتيجة ناجحاً.

عنوان الرسالة (باللغة التي كتبت بها الرسالة)

OPTIMAL FUZZY PID CONTROLLER

نرجو استلام النسخة الورقية التي تمت الموافقة عليها في صيغتها النهائية من قبل المشرف ولجنـة المناقشة، ونسخة من الرسالة على القرص المضغوط (CD)، وذلك لإيداعها في المكتبة حسب الأصول.

وتفضلوا بقبول فائق الاحترام،،،

المشرف	رئيس قسم التخصص أو نائب رئيس لجنة الدراسات العليا في كلية التخصص	نائب عميد كلية الدراسات العليا
التوقيع	التوقيع	التوقيع
التاريخ: 2/12/10	التاريخ: ٩/١٢/٢٠١٠	التاريخ: ٩/١٢/٢٠١٠

تعتمد كلية الدراسات العليا
هذه النسخة من الرسالة
التوقيع ..... التاريخ ٩/١٢/٢٠١٠

مواصفات الاقراص المدمجة الخاصة بالرسائل الجامعية

- ان يضم القرص المدمج كافة المعلومات الواردة في النسخة الورقيه من الرسالة وذلك ضمن ملف واحد.
- ان يكون ترتيب الرسالة على القرص حسب ترتيب النسخة المطبوعة ورقيا.
- ان يحتوي القرص على صورة (save as jpg) عن اجازة الرسالة موقعة وموثقة من اعضاء لجنة المناقشة ومعتمدة من قبل الجامعة.
- تخزين الرسالة في ملف آخر على شكل (Acrobat reader PDF) لتسهيل تفعيل الرسالة على شبكة الانترنت ضمن قاعدة الرسائل الجامعية كاملة النص.

OPTIMAL FUZZY PID CONTROLLER

By

Tagreed M. Al - Jarrah

Supervisor

Dr. Musa O. Abdalla

This Thesis was Submitted in Partial Fulfillment of the Requirements for

the

Master's Degree of Mechanical Engineering

Faculty of Graduate Studies

The University of Jordan

December, 2010

COMMITTEE DECISION

This thesis (Optimal Fuzzy PID Controller) was successfully defended and approved on 25/11/2010

Examination Committee	**Signature**
Dr. Musa Omar Abdalla, (Supervisor) Assoc. Prof. of Mechanical Engineering	1/12/10
Dr. Moudar Zgoul, (Member) Assist. Prof. of Mechanical Engineering	1/12/2010
Dr. Mohammed Al-Janaideh, (Member) Assist. Prof. of Mechatronics Engineering	1/12/2010
Dr. Mohammed Ghaith, (Member) Assist. Prof. of Mechanical Engineering (Al-Balqa Applied University)	1/12/2010

Dedication

I praise Almighty Allah for giving me the strength, passions, courage and guidance to achieve this work, despite all the difficulties.

I would like to express my gratitude and thanks to my family for their continuous guidance, advice, support, inspiration and love.

Acknowledgment

I am heartily thankful to my supervisor, Dr. Musa O. Abdalla for his tireless support, endless patience, and belief in my abilities. His encouragement, guidance and support from the initial to the final levels enabled me to develop a complete understanding of the subject.

Lastly, I offer my regards and blessings to all of those who supported me in any respect during the completion of this project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

DA	Direct Action
FFC	Feed Forward Compensator
FL	Fuzzy Logic
GA	Genetic Algorithm
GS	Gain Scheduling
IAE	Integral Absolute of Error
ISE	Integral Square of Error
ITAE	Integral Time Absolute of Error
LMI	Linear Matrix Inequality
LQG	Linear Quadratic Gaussian
LQR	Linear Quadratic Regulator
MIMO	Multi Input Multi Output
NB	Negative Big
NM	Negative Medium
NS	Negative Small
ODE	Ordinary Differential Equation
PB	Positive Big

PID	Proportional Integral Derivative
PM	Positive Medium
PMDC	Permanent Magnet Direct Current
PS	Positive Small
PSO	Particle Swarm Optimization
PWM	Pulse Width Modulation
VVVF	Variable Voltage Variable Frequency
ZE	Zero

NUMENCLATURE

B_{CW}	Viscous of the counter weight
B_{EL}	Viscous of the elevator
C_1, C_2	Position weights
$e\ (t)$	Error
$\dot{e}(t)$	Derivative of error
i_a	Armature constant current
J	Moment of inertia
K_b	The emf motor constant
K_{CW}	Stiffness of the counter weight
K_D	PID Derivative gain
K_{EL}	Stiffness of the elevator
K_f	Coefficient of viscous friction
K_I	PID Integral gain
K_m	Motor armature constant
K_P	PID Proportional gain
L_a	Armature inductance
M_{CW}	Mass of the counter weight

M_{EL}	Mass of the elevator
R	Radius
R_a	Armature resistance
r_1, r_2	Uniformly random number
t	Time
T	Tension
T_f	Coulomb friction value (offset)
$\tau_m\ (t)$	Motor torque
$u(t)$	Controller output
$V(t)$	Elevator Actual Speed
V_d	Reference speed
$v_{id}\,(t)$	Particle's velocity
V_{in} (t)	Input Voltage
w_i	Inertia weight
we_i	Weight of rule
x_{gd}	Global best position
$x_{id}\ (t)$	Particle's position

OPTIMAL FUZZY PID CONTROLLER

By
Tagreed M. Al – Jarrah

Supervisor
Dr. Musa O. Abdalla

ABSTRACT

One of the challenging issues in Fuzzy system design is generating the rule-base, which is essentially the control strategy of a system. Traditionally, the construction of Fuzzy Logic controller rules has been mainly based on the operator's control experience or actions. Unfortunately, acquiring rules from experts is not an easy task, moreover it is very difficult for a knowledge engineer to extract rules from static data bases. On the other hand, selecting a set of important fuzzy rules from a given rule base is an important issue in fuzzy rule-base modeling.

In this work, a novel Fuzzy Logic controller design methodology is presented. The method utilizes the Particle Swarm Optimization binary search algorithm to generate the output outcomes in a Fuzzy Logic controller rule base without human experience intervention, as a first optimal screening. In addition, the proposed method utilizes Particle Swarm Optimization algorithm to simplify the rule-base of the Fuzzy Logic controller, as a second screening.

The proposed technique is compared with the well established Lyapunov based Fuzzy Logic controller design in generating the rules. The proposed method generated superior system output results with shorter rule-base list.

Finally, the controller's effectiveness and performance are tested, verified and validated using a gearless traction elevator control application. The novel controller's results are compared with traditional Proportional, Integral and Derivative controller and classical Fuzzy Logic controller, the proposed controller showed superiority in controlling the elevator system based on different control standards.

CHAPTER 1

Introduction

Fuzzy based controllers' designs did not stop since Zadeh established the basis for fuzzy sets in 1965 and approximate reasoning in (1975), which was closely followed by the industrial implementation of Mamdani's work (1974).

Currently, Fuzzy controllers found their way to many commercial products, due to their effectiveness yet simplicity in design. Small survey to the market place will reveal the existence of fuzzy controller in many appliances, such as: washing machines, refrigerators, microwave ovens …etc. Personal items such as digital cameras, toys, inverted pendulum based scooters …etc. Military and automotive applications, such as: automatic gear boxes, smart guided missiles, aircraft control …etc.

In this work, a Fuzzy Logic (FL) PID controller is proposed; to aid in solving elevators control for the ever increasing transportation in high rise buildings.

1.1 Fuzzy Logic PID Controller

Literature survey of Fuzzy Logic (FL) based controllers reveals three main directions of design. These directions are presented in some details in the subsequent sections, which may be listed as: Direct control Action (DA), Feed Forward Compensator scheme (FFC) and Gain scheme. For Gain Scheduling (GS) type controllers, Antonio (1999) presented a novel method based on the fuzzification of the set-point weight for the tuning of the PID controller. A fuzzy inference system was used to determine the value of the weight that multiplies the set-point for the proportional action depending on the current output error and its derivative. This method is done by added the output of the fuzzy module to a constant parameter resulting in a coefficient that multiplies the set-point. The parameters of the PID are tuned using the Ziegler-Nichols method. Fuzzy module parameters can be tuned by hand or by means of an auto tuning procedure based on GA. Woo, et al. (2000) presented a method to tune the scaling factors of the PID type Fuzzy controller on line based on to the system error information. The PID type Fuzzy controller can be decomposed into the equivalent proportional, integral and derivative control coefficients. The new method adjusts the input scaling factor corresponding to the integral coefficient of the PID-type FL controller keeping the proportional controller component not to change too much so as to guarantee quick reaction against the error. The simulation shows a good performance of the proposed method. Visilio (2001) presented a comparison between different methodologies, in which fuzzy logic is used to determine the parameters of the PID controller also with a Fuzzy PID-like controller, in which the control variable is determined directly by means of a fuzzy inference system. GA was used to tune the parameters of the fuzzy inference system. Simulation shows that the set-point weighting technique appears superior to the other methodologies. Bandyopadhyay, et al. (2001) proposed a new fuzzy-genetic technique for autotuning the PID controller making the error zero for the next sampling instant as indicated by the format of dead-beat control. The fuzzy inference mechanism which is based on Takagi-Sugeno model was used to determine the expected value of the controller output, while GA was used to derive the rule base. Simulation results showed the superiority of this algorithm.

Guzelkaya, et al. (2003) used the same method used above to adjust the input and the output scaling factors. They proposed a new method for tuning the parameters of PID-type FL controller by using the error and the rate information of the system response together. The fuzzy module that adjusts the related coefficients has two inputs one of which is called 'normalized acceleration' which gives the relative rate information and the second input is the classical error. The output of the fuzzy module is determined as follows: when the system response is slow, the derivative effect of the PID-type FL controller must decrease, and when the error is small and the system response is fast, the derivative effect of the PID-type FL controller must increase, through simulation, which are done on the second order system with varying parameters and delay time, the new self tuning algorithm was compared to other tuning methods. Simulation shows that the new method is more efficient with lower number of parameters to be tuned and it is more robust to the system parameter changes than other methods. Kazemian (2005) presented the development and tuning methods for a novel self-organizing Fuzzy PID controller. In the first tuning method, the fuzzy inference readjusting the gains of the PID controller individually, while in the second tuning method the fuzzy inference readjusting the proportional PID gain and using the Ziegler-Nichols method to determine the required values of the other two gains. The results of the step input experiments for the Fuzzy PID controller and the self-organized Fuzzy PID controller are the same for both methods. The results of the step input show a superiority of the novel self organizing Fuzzy PID controller over the Fuzzy PID controller and the conventional PID controller.

For DA type Fuzzy PID controller, James, et al. (2000) proposed a design for a new Fuzzy PID controller, with its control performance evaluation and stability analysis having a capability of controlling some known nonlinear systems. All possible combinations corresponding to the three different input components were viewed as a cube divided into forty eight sectors to construct the deffuzzification rules. Simulation results showed the effectiveness of the controller for nonlinear and linear systems. Georg, et al. (2001) proposed evaluating different PID controller structures. The structures were evaluated in terms of two levels of tuning. The first level tunes the nonlinear PID gains and the second level tunes the linear gains in addition to the scaling factors. A greater flexibility and better functional properties were achieved by using the decoupled rule and one input rule structures. In the next paper Georg proposed a new tuning method based on two-level tuning strategy for tuning Fuzzy PID controllers. Tang, et al. (2001) proposed an optimal Fuzzy PID controller. The conventional analog PI+D controller was discretized by applying the bilinear transform, then designed the Fuzzy PI and the Fuzzy D controllers separately, and finally combined them together. The plane was divided into twenty regions for both PI and D controllers to construct the deffuzzification rules. Multi-objective GA was used to optimize the gains of the Fuzzy PID controller. An example is given to show that the proposed controller is suitable for controlling the nonlinear plants.

Hybrid type of Fuzzy PID controllers is a combination of DA and GS types. Isin, et al. (2006) designed a new hybrid Fuzzy PID controller by suggesting a combination between the classical PID and Fuzzy PID controller in a blending mechanism that depends on a certain function of actuating error. The proposed hybrid Fuzzy PID controller is anticipated to generate a better simulation results when compared to the pure classical PID controller or the pure Fuzzy controller applications.

1.2 Lyapunov Synthesis Approach

Many methods have been investigated on the stability of Fuzzy control systems. Buhar and Leephakpreeda (1994) have applied describing function methods to evaluate the stability of Fuzzy control systems. The Lyapunov's stable method is one of the most useful tools for handling the stability problems. Based on this method, Michael and Gideon (1999) presented a new method, based on extending the classical Lyapunov synthesis method to the design of Fuzzy controllers. Chaio-Shiung (2001) presented a systematic procedure to analyze and design a stable Fuzzy controller for a class of nonlinear systems. Changjin (2002) proposed a novel approach to design stable Fuzzy controller with perception based information using fuzzy arithmetic based Lyapunov synthesis in the frame of computing with words. Petr, et al. (2003) presented a constructive and automated method for the design of a gain scheduling FL controller based on Lyapunov method and convex optimization. Gang (2003) proposed a kind of controller synthesis based on a piecewise smooth Lyapunov function. Bong-Jae and Sangchul (2006) presented anew fuzzy Lyapunov function approach for the stability analysis and the Fuzzy controller synthesis of a class of the continuous time Takagi-Sugeno Fuzzy control system. Tiejun, et al. (2007) proposed two novel stable fuzzy model predictive controllers based on piecewise Lyapunov functions and the min-max optimization of a quasi-worst case infinite horizon objective function. Jeng, et al. (2007) presented a new FL controller for discrete time systems based on Lyapunov stability criterion. Kaushik, et, al. (2009) proposed a new approach for designing stable adaptive Fuzzy controllers, which employs a hybridization of a conventional Lyapunov theory based approach and a particle swarm optimization (PSO) based stochastic optimization approach.

1.3 Particle Swarm Optimization (PSO) Synthesis Approach

Particle Swarm Optimization (PSO) is a population-based stochastic optimization technique developed by Eberhart and Kennedy (1995), Cui, et al. (2004) described a swarm based FL controller mobile sensor network approach for collaboratively locating the hazardous contaminants in an unknown large scale area.

Wahsh, et al. (2005) used the PSO technique to tune the parameters of the PI-controller for control of permanent magnet synchronous motor. Gao and Tong (2006) proposed a novel PID controller tuning and on-line tuning approach based on the PSO to design robust PID parameters by transforming the problems of PID controller into correspondent optimization problems. Kao, et al. (2006) presented a novel design method for the self-tuning PID control in a slider-crank mechanical system. Ermanu, et al. (2007) presented power system stabilizer design based on optimal PD and PI Fuzzy controller. Allaoua, et al. (2008) proposed the application of Fuzzy Logic for DC motor speed control using Particle Swarm Optimization. Elwer and Wahsh (2009) presented a modern approach for speed control of a PMSM using the PSO algorithm to optimize the parameters of the PI-controller. (Bingul and Karahan, 2010) controlled a 2 DOF planar robot by FL controller tuned with a PSO.

1.4 Elevator Control Strategies

Many studies were carried out in controlling the elevator systems. (Kang, et al., 2000) proposed a new strategy to reduce the vertical vibration of the left car while keeping high speed control and as a result improve the efficiency of riding elevators. Both experimental evaluation and computer simulation proved the feasibility of this strategy. Mannan, et al. (2001) proposed an electro-hydraulic system for the control of an elevator with twin cylinders that are located on each side of the elevator car. A PD Fuzzy controller is applied to velocity control, where as a constrained step PD controller guarantee the minimum non-synchronous error between the motions of two cylinders. Sha, et al (2002) introduced an approximation linear model for a hydraulic elevator that includes an improved dynamic friction model and investigated a sliding mode control for velocity tracking in the discrete domain. Simulation experiments showed that this approach offers an effective and improved solution for the hydraulic elevator control. Huayong, et al. (2004) studied the computational simulation and experimental research on the variable voltage variable frequency (VVVF) hydraulic elevator speed control. The research results provided a theoretical basis for the design and application of the VVVF hydraulic elevator. Kim, et al. (2005) proposed a two-stage non-linear robust-controller, using the Lyapunov redesign method to control the velocity of the hydraulic elevator. Zhou, et al. (2008) introduced a hybrid backup power system, including batteries, ultra capacitors and hydrogen fuel cells in order to get a reliable and effective continuous function elevator in spite of power failure caused by any reason.

CHAPTER 2

Fuzzy Logic in a Nutshell

2.1 Introduction

In this chapter, Fuzzy logic is presented in an informal way. Only the main bold lines of this relative new science are presented, Abdalla (2009). The reader is directed for a more well established references for a complete treatment of the subject. The development of fuzzy set theory was initiated by Zadeh, who discovered the existence of fuzzy sets and proposed it in his seminal paper “Fuzzy sets”, Zadeh (1965). Seven years later, he proposed the basic idea of Fuzzy Logic (FL) controller, Zadeh (1972, 1973). In the next year, the first industrial implementation of a FL controller was reported by Mamdani and Assilian (1974). Today, Fuzzy Logic Controllers can be found in a growing number of products, from washing machines to speedboats, which range from air condition units to hand-held auto focus cameras. The motivation is often that the fuzzy set theory provides an alternative design approach to the traditional modeling and design of control systems, this is especially more effective when system knowledge and the dynamics of the system, in the traditional sense, are uncertain and time varying.

Humans use approximate reasoning to draw vital conclusions. For example, humans use their eyes to approximate or estimate distances while driving, yet their controller (brain) regulate the speed and path following and other functions in a remarkable way! Such things had triggered a complete process of design approaches not based on system's model but instead it utilizes established experience. Researchers and designers started to believe if humans can take vital decisions based on non-precise data why not machines!

The tricky part of these eccentric approaches was to have the ability to capture this type of logic and reasoning in a rigorous mathematical manner. Actually, Zadeh (1965) succeeded in proposing a new way of perception based on non-conventional logic, he later called it Fuzzy Logic (FL). In classic logic things are on two extremes: either True or False, which was inherited from the Greeks and other civilizations. The drawback of such reasoning was apparent, that is an event may not be True and False at the same time. But if one carefully studies human behavior he/she will soon come to the conclusion that humans use contradictory remarks to the conventional logic. An example on that would be pattern clustering; a classification based on height will trigger many human's linguistic words, such as: vary tall, tall, kind of medium, medium, not too short, short… etc. While classic or conventional logic fails to address all these human accepted variations, suddenly a more need of a new type of logic was arising. Also, from a control point of view the sudden change in the classification was not accepted, that is in conventional logic the transition between tall and short (i.e. only two possibilities), a person classification would change over a night or after putting on his shoes! Even though the digital world has succeeded in implementing the classical logic but when it comes to control this non-smooth transition becomes a major barrier in the implementation.

Consequently, Zadeh's reasoning was very simple yet very powerful, which maybe simply put why not consider other shades of truthiness. That is why not to be true and at the same time false, or in a mathematical sense, the truthiness must be in the range [0, 1]. It turns out that these shades may solve the human like mystery of approximate reasoning.

It actually enables us to address human confusing linguistic terms such as “somewhat tall “, which implies tallness and shortness in a mixed manner! On the other hand, smooth transition of classifications may be created to address the need for controller designs. Figure 1 illustrates the height concept in a human perception based way. Such graphs are called membership functions because they represent the degree of belonging of a certain element to a set, in this case the height set.

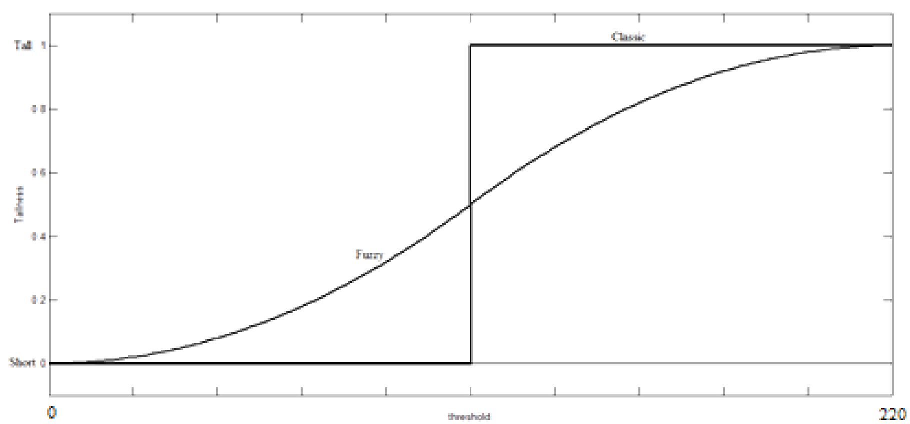

Figure 1. Classification logic of human height

In classical logic this belonging (i.e. membership) of an element is simply {0, 1}. That is it either part of the tall set {1} or it is not {0}. However, different shades [0,1] may be used in the fuzzy logic approach! Mathematically speaking, in classic set theory, a subset “A” of set X can be defined by its characteristic function X_A as a mapping from the elements of X to the elements of the set {0,1},

$$X_A : x \rightarrow \{0,1\}$$

And the truthiness of a statement like "x is in A" is determined by the ordered pair $(x, X_A(x))$. Similarly, in fuzzy logic the mapping is performed through membership functions on the range [0, 1],

$$\mu_A : x \rightarrow [0, 1]$$

Where X and A are fuzzy sets. Also, in the previous statement truthiness is determined by the ordered pair (x, μ_A (x)). Where $\mu_A(x)$ reflects the degree of membership or belonging of element x in fuzzy set A for each $x \in$ X. Hence, the A set may be completely described by

$$A = \{(x, \mu_A(x): x\epsilon X\} \quad (1)$$

2.2 Fuzzy Sets

Conventional set theory was originally developed by George Cantor (1845-1918). Later Zadeh and others extended most of the fuzzy set theories in a rigorous mathematical way to resemble the complete classical set theory. Elements of a fuzzy set are taken from a universe of discourse, or simply universe. This universe contains all the elements that come into consideration. For example the universe for a sensor is basically the collection all possible readings from that sensor.

On the other hand, every element in the universe is a member of the fuzzy set to some grade, could be zero (no belong) or one (totally belong). The function that generates this grade of belongness or no belongness is the membership function $\mu(x)$, which could be continuous or discrete. In fuzzy based control this function is preferably continuous, normalized and convex. Finally, similar to an algebraic variable in taking numbers as values, linguistic variables take words or sentences as values in Fuzzy Logic (Zadeh, 1975 and Zimmermann, 1993).

For an example, let *x* be a linguistic variable for height, one may define the set of words (terms) for the linguistic variables, which are fuzzy sets, to be something like
T= {very tall, tall, medium, short, very short}. Actually, each term is a fuzzy variable, which is clearly depicted in Figure 2.

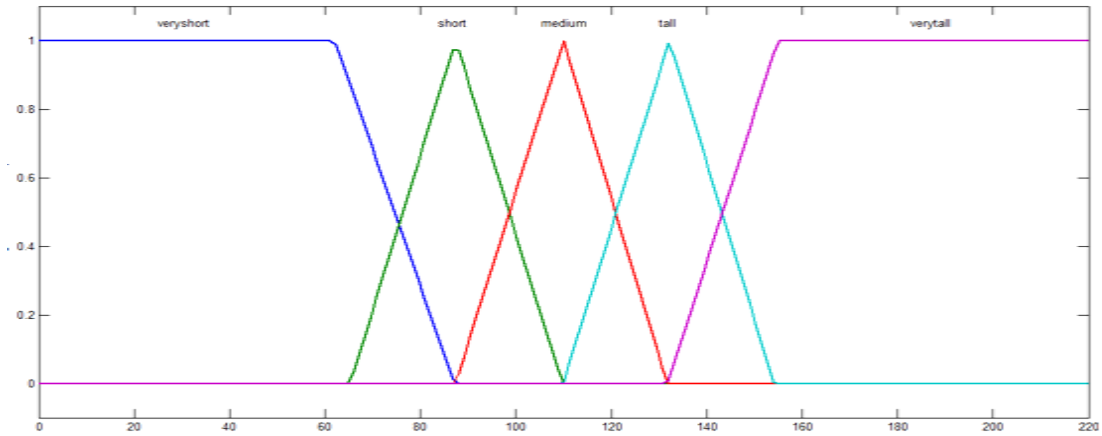

Figure 2. The fuzzy sets for the height example

2.3 Operations on Fuzzy Sets

The new definition of a fuzzy set, which differ from the concept of a classic or crisp set, has entailed new approaches for the operations on fuzzy sets. Consider the fuzzy sets A and B in the universe U,

$$A = \{(x, \mu_A(x))\}\ ,\ \mu_A(x) \in [0,1] \qquad (2)$$

$$B = \{(x, \mu_B(x))\}\ ,\ \mu_B(x) \in [0,1] \qquad (3)$$

The operations with A and B are defined on fuzzy sets by means of their membership functions $\mu_A(x)$ and $\mu_B(x)$, correspondingly. Here is a short review of these operations;

- **Equality**: The fuzzy sets A and B are said to be equal and are denoted by A=B if and only if for every $x \in$ U,

$$\mu_A(x) = \mu_B(x) \tag{4}$$

- **Inclusion**: The fuzzy set A is included in the fuzzy set B denoted by $A \subseteq B$, if for every $x \in$ U, $\mu_A(x) \leq \mu_B(x)$. Then A is called a subset of B.

- **Proper Subset**: The fuzzy subset A is called a proper subset of the fuzzy set B denoted by $A \subset B$ when A is a subset of B and A ≠B that is:

$$\mu_A(x) \leq \mu_B(x) \text{ for every } x \in \text{U},$$

$$\mu_A(x) < \mu_B(x) \text{ for at least one } x \in \text{U}$$

- **Complementation**: The fuzzy sets A and $\bar{A}$ are complementary if

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \tag{5}$$

- **Intersection**: The operation intersection of A and B is denoted as A∩B and is defined by:

$$\mu_{A\cap B}(x) = \min\big(\mu_A(x), \mu_B(x)\big), x \in \text{U} \tag{6}$$

- **Union**: The operation union of A and B is denoted as A∪B and is defined by:

$$\mu_{A\cup B}(x) = \max\big(\mu_A(x), \mu_B(x)\big), x \in \mathrm{U} \quad (7)$$

- **Difference**: The operation difference is denoted by A-B and is defined by:

$$\mu_{A-B}(x) = \min\big(\mu_A(x), \mu_{\bar{B}}(x)\big) \quad (8)$$

On the other hand, a linguistic *modifier* is an operation that modifies the meaning of a term. The intensification modifiers such as *very* and *extremely* (or *very very*), and dilution modifiers such as *somewhat* (or *more less*), slightly and *greatly* are the most frequently used modifiers. The intensification modifiers can be given in the following form

$$int\ \mu(x) = \mu^n(x) \quad (9)$$

Where int refers to an intensification modifier with $n \geq 2$. The value of n = 2 in the case the modifier *very* and n = 3 for the modifier *extremely*. Dilution modifiers have a similar definition equation, except that the power is inverted

$$dil\ \mu(x) = \mu^{\frac{1}{n}}(x) \quad (10)$$

The value of n = 2 in the case of the modifier *somewhat*, n= 3 in the case of the modifier *slightly* and n = 1.4 for *greatly*. These values are typically argued for in the literature. However, other values are also possible.

2.4 Fuzzy Logic Operations

Fuzzy logic is a form of multi-valued logic derived from fuzzy set theory to deal with reasoning that is approximate rather than accurate. In contrast to "crisp logic", where binary sets have binary logic, fuzzy logic variables may have a truth value that ranges between zero and one and is not constrained to the two truth values of classical propositional logic.

Fuzzy connectives are used to join simple fuzzy propositions to make compound propositions. Negations (~), disjunctions (∪), conjunctions(∩), and implications (⇒) are widely used as fuzzy connectives.

A fuzzy implication is a generalization of the classical two-valued fuzzy logic. In literature, there is three important classes of fuzzy implication operators; S-implications, R-implications and t-norm implications. Their basic definitions are enlisted for reference.

- **S-implications**: It is given as

$$x \Rightarrow y \equiv (S(n(x), y)) \tag{11}$$

Where S is a t-conorm and n is a negation on [0,1]. Typical examples of S-implications are the Lukasiewiez and Kleene-Dienes implications, see Table 1.

- **t-norm implications**: if T is a t-norm then $x \Rightarrow y = T(x, y)$. Typical examples of t-norm implications are the Mamdani and Larsen implications, see Table 1.

- **R-implications**: these are obtained by residuation of the continuous t-norm T, that is

$$x \Rightarrow y \equiv sub\{z \in [0,1] | T\ (x, z) \leq y\} \qquad (12)$$

Typical examples of R-implications are the Gödel and Gains implications, see Table 1.

The most often used fuzzy implications operators are shown in Table 1, (Abdalla, 2009). These implications range from simple definitions to more complicated ones. The diversity is arising from the author's effort to satisfy all the classical logic situations. However, in control theory simpler definitions mean easier in the implementation.

Table 1. Fuzzy implication operators

Name	Definition
Early Zadeh	$x \Rightarrow y = \max\{1 - x, \min(x, y)\}$
Lukasiewiez	$x \Rightarrow y = \min\{1, 1 - x + y\}$
Mamdani	$x \Rightarrow y = \min\{x, y\}$
Larsen	$x \Rightarrow y = xy$
Standard Strict	$x \Rightarrow y = \begin{cases} 1 & \text{if } x \leq y \\ 0 & \text{otherwise} \end{cases}$
Gödel	$x \Rightarrow y = \begin{cases} 1 & \text{if } x \leq y \\ y & \text{otherwise} \end{cases}$
Gaines	$x \Rightarrow y = \begin{cases} 1 & \text{if } x \leq y \\ y/x & \text{otherwise} \end{cases}$
Kleene-Dienes	$x \Rightarrow y = \max\{1 - x, y\}$
Kleene-Dienes-Lukasiewiez	$x \Rightarrow y = 1 - x + xy$
Yager	$x \Rightarrow y = y^x$

2.5 Fuzzy Inference

The process of drawing conclusions from existing data is called inference. A system of fuzzy IF-THEN rules is considered as a knowledge base system where inference is made on the basis of three rules of inference; compositional of inference, modus ponens and generalized modus ponens.

Zadeh (1975) introduced the compositional rule of inference that plays the most important role in approximate reasoning. Fuzzy conditional statements in the form: IF A then B denoted by A $\Rightarrow$B with the fuzzy sets A (antecedent) and B (consequent), which may be considered as fuzzy relations.

In conventional logic, reasoning is based on "modus ponens" (deduction) and "modus tollens" (induction). The two mechanisms are contrasted in Table 2.

Table 2. Deduction and Induction

	Deduction	Induction
Rule	IF x is A then y is B	IF x is A then y is B
Premise	x is A	y is not B
Conclusion	y is B	x is not A

In Fuzzy Logic (FL), the modus ponens and modus tollens are extended to the generalized modus ponens and generalizrd modus tollens, respectively.

A generalized (fuzzy) modus ponens inference rule is defined by the following reasoning scheme:

Premise 1: If x is A $\Rightarrow$ y is B

Fact: x is A'

Consequence y is B'

Were A, A', B and B' are all fuzzy sets, and x and y are the so-called linguistic variable. The consequence B' is determined as a composition of the fact and the fuzzy implication operator

$$B' = A' \circ (A \rightarrow B) \tag{13}$$

That is,

$$B'(v) = sup_{u \in U} \{ \min A', (A \rightarrow B)(u, v) \}, v \in V \tag{14}$$

The consequent B' is nothing else but the shadow of A $\rightarrow$ B on A'. The generalized modus ponens, which reduces to classical modus ponens where A'=A and B'=B, is closely related to the forward data-driven inference which is particularly useful in the Fuzzy Logic (FL) control.

A generalized (fuzzy) modus tollens can be written in IF-THEN form as

Premise 1: If x is A $\Rightarrow$ y is B

Fact: y is B'

Consequence x is A'

Which reduces to modus tollens when $B=\overline{B}'$ and $A'=\overline{A}$, is closely related to the backward goal driven inference. The consequence A' is determined as a composition of the fact and the fuzzy implication operator

$$\overline{A}' = \overline{B}' \circ (A \rightarrow B) \qquad (15)$$

Finally, fuzzy inference systems have been successfully applied in fields such as automatic control, data classification, decision analysis, expert systems and computer vision. Several fuzzy inference systems have been described by different workers but the commonly used are Mamdani type (1974) and Takagi-Sugeno type (1985), which is also known as Takagi-Sugeno-Kang type.

The Takagi-Sugeno type is similar to the Mamdani in many respects. The first two parts of the fuzzy inference process, i.e. fuzzifying the inputs and applying the fuzzy operator, are exactly the same. The main difference between them is that, the output membership functions are linear or constant for Takagi-Sugeno type fuzzy inference, where as the output membership functions are fuzzy sets for the Mamdani type.

Mamdani fuzzy inference method is the commonly seen fuzzy methodology, another advantages of the Mamdani method is that it is intuitive and well suited to human input. In this work, the Mamdani fuzzy inference method will be used.

CHAPTER 3

Fuzzy Logic Controller Design

Controller design is one of the most important topics in control theory. The word control itself denotes the ability to reshape the dynamical system response (behavior) based on a user desired input. Typically, controllers design objectives include: stabilization of unstable systems, disturbance rejection, enhancement of performance and beating system's uncertainty. The Fuzzy Logic (FL) controller objectives are the same, but the difference is in the controller's design methodology.

3.1 Introduction

Nowadays, there are two schools of thoughts in controller design; the first approach depends on enforcing the controller's objectives through a model-based design techniques. In this method the designer is required to have a functioning mathematical model of the dynamical system. The design process could be in the frequency domain, time domain or both. Classical control techniques such as PID, Lead Lag …etc controllers perform the design in Laplace or frequency domain. However, techniques such as Linear Quadratic Regulator (LQR), Linear Quadratic Gaussian (LQG) … etc, carries out the design in the time domain. While modern techniques, such as LMI, H_∞ …etc, performs the design in both domains time and frequency. Such techniques in both domains start its design process with the model. This model could be based on analytical physical analysis of the dynamical system or it could be based on experimental data. Unfortunately, this model could be complex or not even available for the designer.

The second approach forces the controller objectives in a model-less approach. The reasoning behind these techniques is simply based on the observation of humans and animals ability to function many control tasks without any knowledge of the system's model! For example human's children learn walking (stabilization) by continuous training and transfer of knowledge from the parents. Later when children grow, they learn car driving through skills transfer. This had triggered the model-less design methodology among researchers and engineers, creating Fuzzy Logic (FL) and neural-network control based techniques. These techniques do not require the systems mathematical model instead known skills are captured through training.

Actually, according to Takagi and Sugeno (1985), Fuzzy Logic (FL) design approach stems out its control ability from three sources :

- Expert experience and control engineering knowledge, such as known rules of thumbs operators hand books … etc.
- Based on operator's control action. Capturing someone experience in an abstract way (training and observation).
- Based on learning. Basically, a self organizing controller that extracts a trend or pattern using data mining, such as neural-network technique.

3.2 Fuzzy Logic Controller Configuration

Fuzzy Logic (FL) controller may directly replace the standard Proportional, Integral and Derivative (PID) in the Loop. Figure 3 depicts a direct control action for a tracking system.

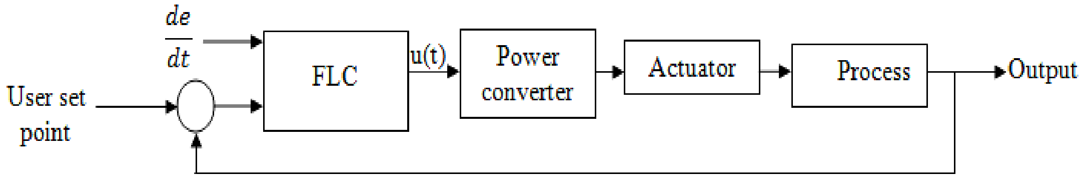

Figure 3. Direct control action for a tracking system

In this case the FL controller's decision is based on the deviation or the error and its rate of change. Typically, Fuzzy Logic (FL) direct control scheme may replace the conventional PID controller with an intrinsic intelligent added to it.

Figure 4 depicts the FL controller used as a disturbance compensator in the forward path (feedforward control). In this case the FL controller is a complimentary controller and it is used to enhance the main controller's (could be linear or non linear) control laws.

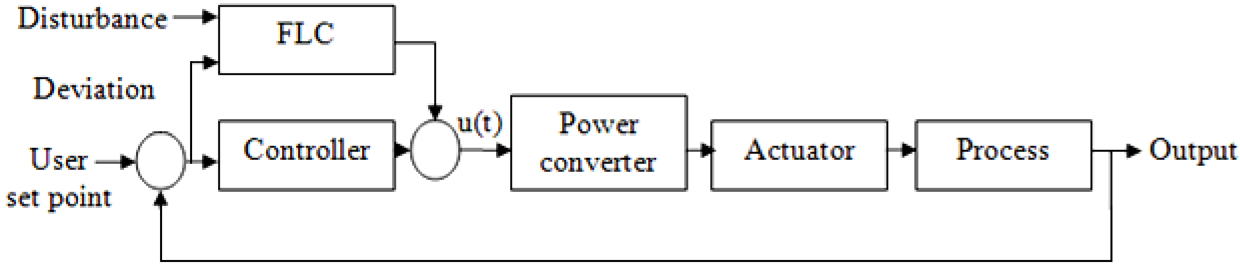

Figure 4. Fuzzy Logic feed forward compensator scheme

Finally, the FL controller may be used to adaptively tune the controller's parameters. This is known as Gain Scheduling (GS), which is used for the cases were the non-linear plants change their operating point (Linearized plants around equilibrium points). Figure 5 illustrates the conceptual design of this scheme.

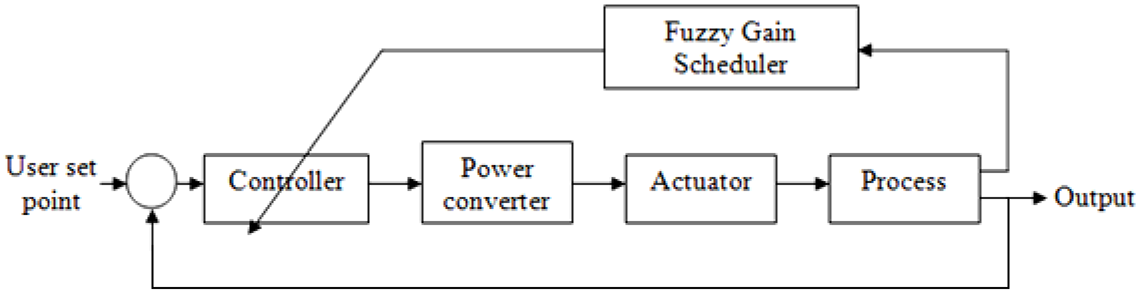

Figure 5. Fuzzy Logic gain scheduling scheme

3.3 Fuzzy PID Controller

The three main Fuzzy Logic controller models are: Mamdani, Sugeno and Tsukamoto models. Mamdani's Fuzzy Logic controller is the most commonly used Fuzzy Logic controller model due to its effectiveness and simplicity. This model expects the output membership functions to be fuzzy sets. After the aggregation process, there is a fuzzy set for each output variable that needs defuzzification.

Sugeno model has many similarities to the Mamdani type. In fact the first two parts of the fuzzy inference process (fuzzifying the inputs and applying the fuzzy operator) are exactly the same. The main difference between the two models is that the output membership functions are only linear or constant for the Sugeno model.

In Tsukamoto model, the consequent of each fuzzy if-then rule is represented by a fuzzy set with a monotonical membership function. As a result, the inferred output of each rule is defined as a numeric value induced by the rule firing strength. The overall output is taken as the weighted average of each rule's output. However, the Tsukamoto fuzzy model is not used often since it is not as transparent as either the Mamdani or the Sugeno models.

In this work, Mamdani Fuzzy Logic controller model is used. A typical Fuzzy Logic (FL) controller structure is depicted in Figure 6. According to Lee (1990), a FL controller is comprised of four principal components; a fuzzification interface, a knowledge base, an inference system and a defuzzification interface.

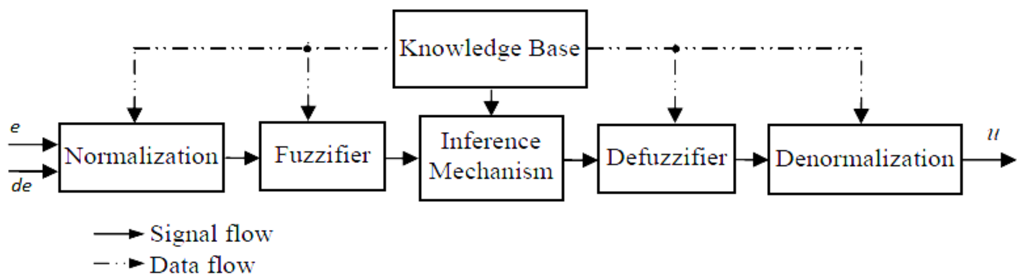

Figure 6. Fuzzy Logic controller internal structure

3.3.1 Fuzzification

The fuzzification module converts the crisp values of the control inputs into fuzzy values, so that they are compatible with the fuzzy set representation in the rule base. The knowledge base consists of a data base of the plant. It provides all the necessary definitions of the fuzzification process such as membership functions, fuzzy set representation of the input-output variables and the mapping functions between the physical and the fuzzy domains.

One of the most challenging issues in fuzzy systems design is generating membership functions for the fuzzy design variables. A quick overview of the most popular membership functions is provided here for reference, (Massad, et al., 2008).

- **Triangular membership function**

This is the most popular membership function due to its simplicity and easiness in calculations. It is specified by three parameters $\{a,b,c\}$ as follows:

$$triangle(x;a,b,c) = \left\{ \begin{matrix} 0 & ,x < a \\ \frac{x-a}{b-a} & ,a \le x \le b \\ \frac{c-x}{c-b} & ,b \le x \le c \\ 0 & ,c \le x \end{matrix} \right\} \quad (16)$$

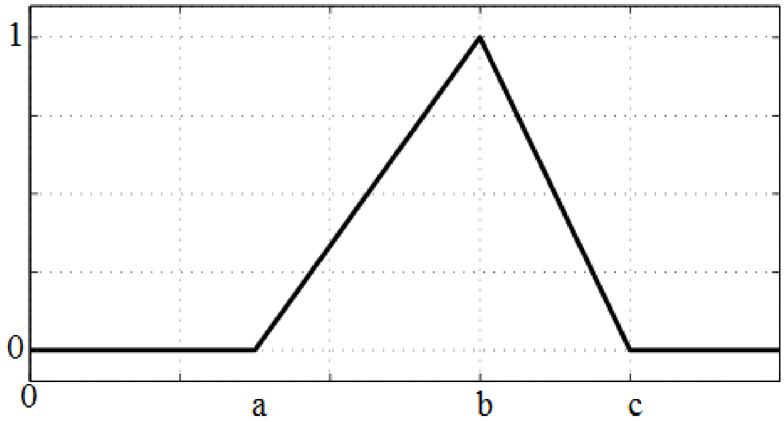

Figure 7. Triangular membership function

As shown in Figure 7, the parameters $\{a,b,c\}$ with $a<b<c$ determine the x coordinates of the three corners of the underlying triangular membership function.

- **Trapezoidal membership function**

A trapezoidal membership function is specified by four parameters $\{a, b, c, d\}$ as follows:

$$trapezoid(x; a, b, c, d) = \left\{ \begin{array}{cc} 0 & , x < a \\ \frac{x-a}{b-a} & , a \leq x \leq b \\ 1 & , b \leq x \leq c \\ \frac{d-x}{d-c} & , c \leq x \\ 0 & , d \leq x \end{array} \right\} \quad (17)$$

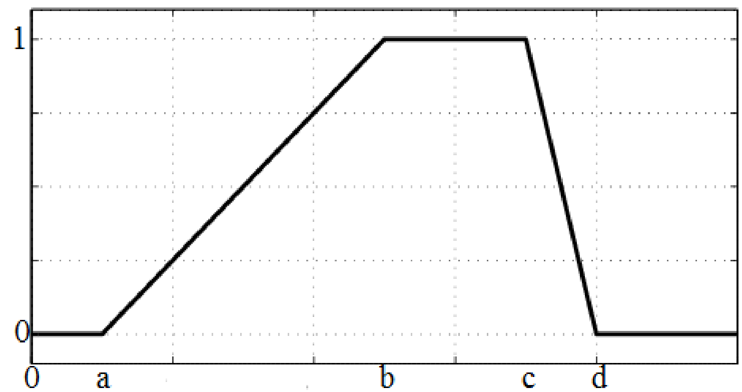

Figure 8. Trapezoidal membership function

As shown in Figure 8, the parameters $\{a,b,c,d\}$ with $a<b<c<d$ determine the x coordinates of the four corners of the underlying trapezoidal membership function.

- **Gaussian membership function**

A Gaussian membership function can be specified by two parameters $\{c, \sigma\}$

$$gaussian(x, c, \sigma) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2} \quad (18)$$

A Gaussian membership function is determined completely by c and σ; where c represents the membership function's center and σ determines the membership's function's width.

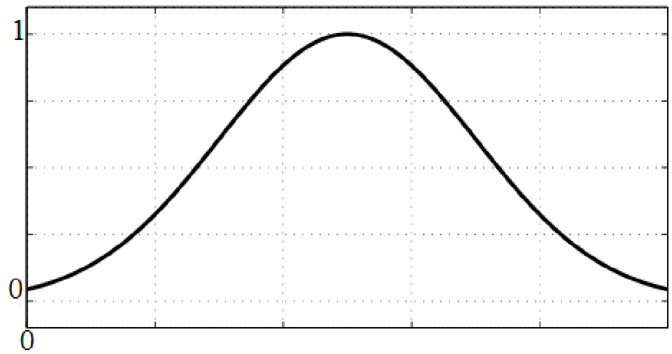

Figure 9. Gaussian membership function

- **Generalized bell membership function**

A generalized bell membership function (or bell membership function) is specified by three parameters $\{a,b,c\}$

$$bell(x;a,b,c,) = \frac{1}{1+\left|\frac{x-c}{a}\right|^{2b}} \tag{19}$$

Where the parameter b is usually positive. The parameter c locates the center of the curve. It is also called the Cauchy membership function.

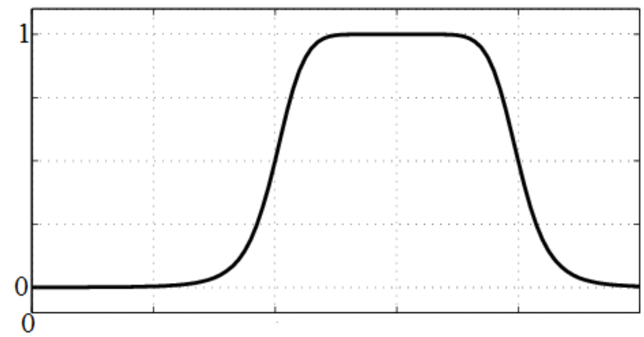

Figure 10. Generalized bell membership function

- **Sigmoid membership function**

A sigmoid membership function is defined by:

$$sig(x; a, c) = \frac{1}{1+\exp[-a(x-c)]} \qquad (20)$$

Depending on the sign of the parameter a, the sigmoid membership function is inherently open to the left or to the right.

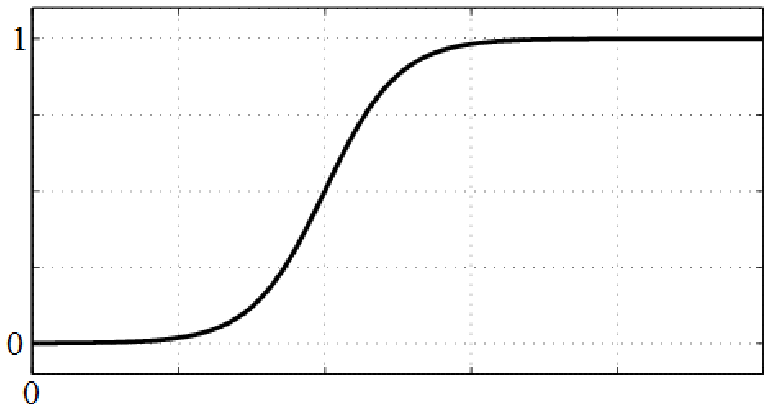

Figure 11. Sigmoid membership function

3.3.2 Rule base

Another challenging issue in fuzzy systems design is generating the rule base that is essentially the control strategy of the system. There are at least four main sources for generating control rules (1998); that is, experience and control engineering knowledge, based on the operator's control actions, based on a fuzzy model of the process or based on learning, and optimization. In this work, the learning optimization approach will be used.

Fuzzy rules are described in terms of IF-THEN conditions. These rules cover all linguistic terms for the required inputs and match them to conclusions: $If\ x\ is\ A\ then\ y\ is\ B$. As one can imagine, the more linguistic terms there are for a given universe of discourse (crisp input) then the number of inputs greatly affects the size of the rule set.

In order to determine to what degree a rule applies to the input parameters, a rule's firing strength may be calculated. There are many methods that can be used to determine the fire strength of a rule such as MAX-MIN method (Mamdani's approach). The MAX-MIN method of determining the fire strength of a particular rule involves taking the degree of membership values for each input into the rule. The first strength is then determined by the smallest of the fire strengths. Now, for multiple firing of the rules, there are two techniques that are used:

- **Combining the rules**: In this technique the rule may be aggregated

$$R = agg(R_1, R_2, \dots R_n)$$

Where connectives and/or must be applied.

- **Combining the fired outputs:** Here the strength of firings are aggregated to generate the total output.

$$C = agg(C_1, C_2, \dots C_i)$$

Where C_i are the corresponding outputs from the i^{th} rule.

In this work, the output aggregation has been adopted in order to implement the binary search optimization algorithm.

3.3.3 Defuzzification

the defuzzification module translates the generated controller's fuzzy control actions to non fuzzy control actions (crisp) in order to be compatible with the actuator. There are many techniques to perform the defuzzification and the most common in practice are enlisted for reference.

- **Center of Gravity method**

This method is also referred as the center of area method and this is the most widely used defuzzification technique. For the continuous case the defuzzified output value is obtained from the overall membership function as follows, (Massad, et al., 2008):

$$u^* = \frac{\int z\mu(z)dz}{\int \mu(z)dz} \qquad (21)$$

Where $\mu(z)$ is the aggregated overall membership function, z is the output quantity. For discrete values the difuzzified output is obtained using weighted average method as follows, (Massad, et al., 2008):

$$u^* = \frac{\sum_{k=1}^{n} z_k \mu(z_k)}{\sum_{k=1}^{n} \mu(z_k)} \quad (22)$$

Where $\mu(z_k)$ are the k=1,…., n sampled values of the aggregated output membership function.

- **Mean-Max (Middle Of Maxima) method**

In this defuzzification technique, the average output value is obtained as follows:

$$z^* = \frac{z_1+z_2}{2} \quad (23)$$

Where z_1 is the first value and z_2 is the second value, where the overall membership function, $\mu(z)$, is maximum.

- **First Of Maxima method**

When this defuzzification technique is used, the first value of the overall output membership function with maximum membership $\mu(z)$ degree is taken. This technique is seldom used.

- **Least Of Maxima method**

Here, the least value of the overall output membership function with minimum membership $\mu(z)$ degree is taken. This technique is also seldom used.

In this work the Center of Gravity technique is used to obtain the output for discrete values, Figure 12 below, (Fuzzy Control System, Wikipedia), demonstrates max-min inference and Center of Gravity defuzzification for a system with input variables "x", "y", and "z" and an output variable "n". In Center of Gravity defuzzification the values are OR'd, that is, the maximum value is used and values are not added, and the results are then combined using a centroid calculation.

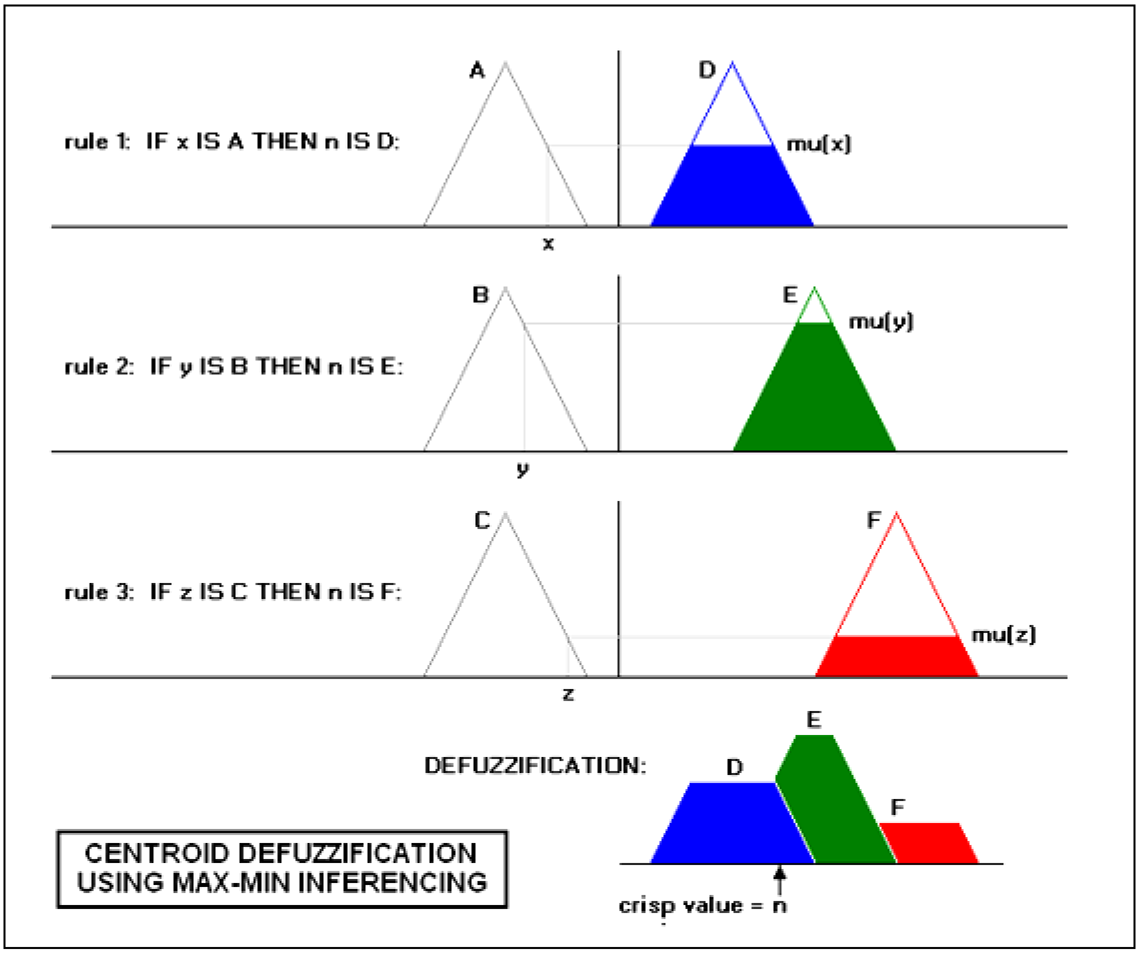

Figure 12. Max-min inference and center of gravity defuzzification

3.4 Proportional Integral Derivative (PID) Controller Overview

Conventional Proportional, Integral and Derivative (PID) controllers despite of their simplicity and fixed structure design they are still used in the industry. The main challenge faced for these controllers is their tuning. There has been hundreds of proposed methods to tune the PID controllers for all sorts of applications, interested readers may consult Technical report AOD-00-01 by A. O 'Dwyer (2001). The popularity of PID controllers is mainly due to their good performance and simplicity in design. Actually PID controllers proved their existence and capabilities for the past fifty years in industrial applications for linear and nonlinear systems. However, other types of controllers should be used for complex systems, such as system with time delays, significant oscillatory behavior, parameters variations and MIMO plants.

Basically, the PID controller inherent its ability to enhance system's performance from its three terms: Proportional (P), Integral (I), and Derivative (D). Figure 13 depicts a typical basic PID controller implementation.

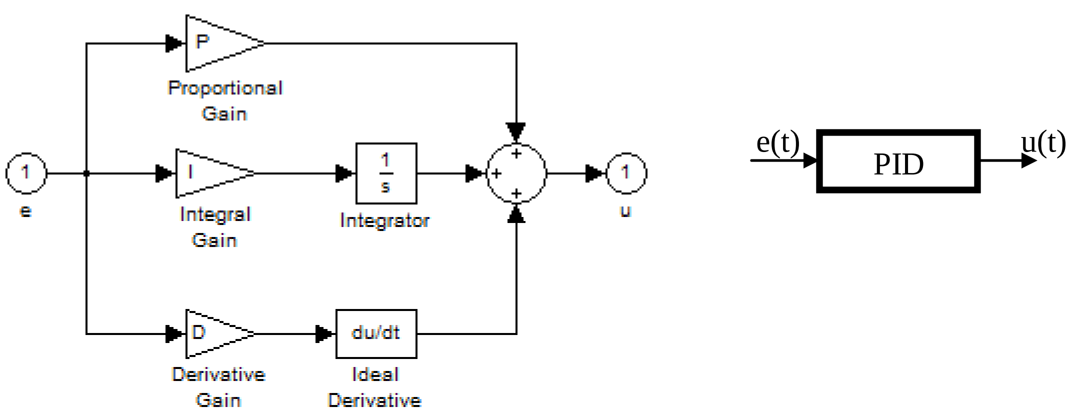

Figure 13. PID controller structure block diagram

Mathematically, the PID controller takes the following form:

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau)d\tau + K_D \dot{e}(t) \tag{24}$$

Where $e(t)$ is the system's deviation from the user's set point as depicted in Figure 14.

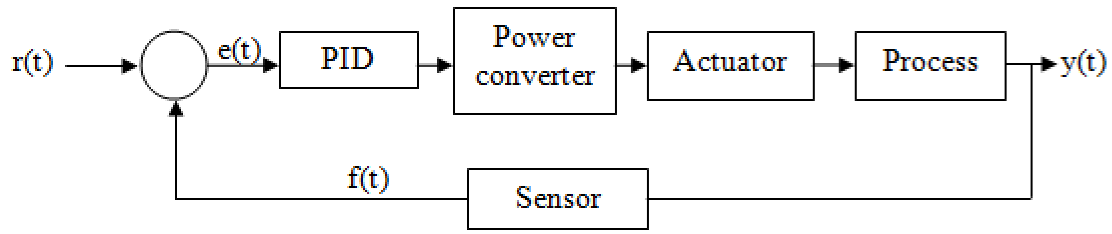

Figure 14. PID controller loop

$$e(t) = r(t) - f(t) \tag{25}$$

Note that for tracking systems $f(t) = y(t)$.

Even though the PID controller response space is small compared to other types of controllers still finding optimal parameters for the PID controller (K_P*, K_I*, K_D*) is truly challenging. Unfortunately, even though the classical PID controller is simple in its structure but it lacks flexibility and intelligence. To make the PID controller smarter, the Fuzzy based PID controller is proposed. Subsequent sections will shed more light on the Fuzzy Logic PID controller structure and parameter tuning.

3.5 Fuzzy Logic Based PID Controller

Depending on the Fuzzy PID controller's structure it may be used to replace, complement or tune the PID controller. But in all cases the conventional PID controller's performance will be boosted. According to Isin et al. (2006), Fuzzy PID controllers maybe classified into three major categories as: Direct Action (DA), Gain Scheduling (GS), and Hybrid type Fuzzy PID controllers.

In Direct Action (DA) type Fuzzy PID controller, the controller is placed within the feedback control loop and the PID actions are determined directly by means of the controller's fuzzy inference. The DA type can be classified according to the number of inputs as single input, double input and triple input DA Fuzzy PID controllers. The DA type is the most commonly used Fuzzy PID controller due to the simple features yet effective non-linear properties. Figure 15 depicts the configuration of the DA type Fuzzy PID controller.

The proposed Fuzzy PID controller, that is a combination between the PD type FL controller and I control, is illustrated in Figure 15. The Fuzzy Logic controller has two inputs; the error and the rate of change of the error. The basic rule base is given by:

$$ELSE_{i,j} \text{ [IF } e(t) \text{ is } E_i \text{ and } \dot{e}(t) \text{ is } \dot{E}_J \text{ then } u \text{ is } U_m \text{]}$$

The total number of rules is equal to $N_1 \times N_2$. Where N_1 and N_2 are the number of the linguistic variables for the error and the rate of change of the error respectively. Input error scaling factor is K_e, error change scaling factor is K_{ce} and output scaling factor is K_{out}. Consequently, the Fuzzy controller output is given as

$$u = K_{out}\ [K_e e(t) + K_{ce}\dot{e}(t) + K_{ie} \int_0^t e(\tau)d\tau] \tag{26}$$

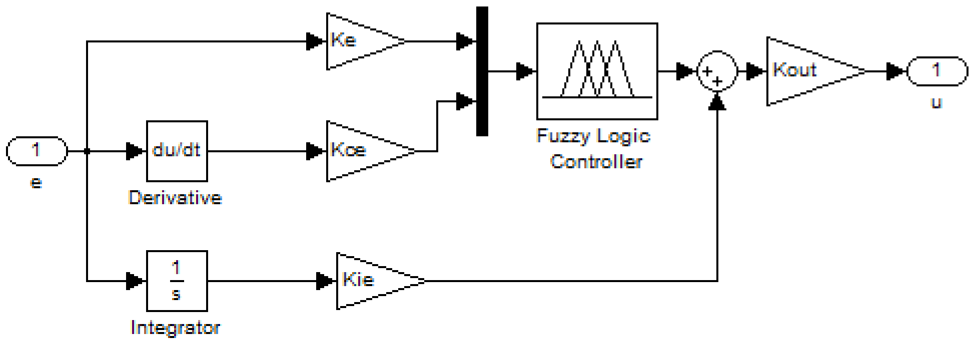

Figure 15. Direct Action Fuzzy PID controller

Any Fuzzy PID structure together with its fuzzy knowledge base usually results in nonlinear PID actions. The nonlinearity is adjusted by modifying the knowledge base parameters (rules, membership functions or support sets). Computational or numerical search techniques are commonly used to produce optimum nonlinear controllers using fuzzy paradigms.

In the Gain Scheduling (GS) type of Fuzzy PID controller, conventional control and fuzzy computing is combined together in the scheme of fuzzy gain scheduling. Tuning parameters of the controller are stored in a fuzzy rule base beforehand, and during control the fuzzy system gives suitable parameter gains for the controller. The GS Fuzzy PID controller is depicted in Figure 16.

Typical application for such a controller is the diesel engine. The model for the diesel engine is nonlinear and it has different equilibrium points that are function of the angular velocity of the engine (rmp). This generates multiple linearized models depending on the engine rpm, which make it challenging to tune the PID controller. The Fuzzy controller will tune the conventional PID controller on real time based on gain scheduling.

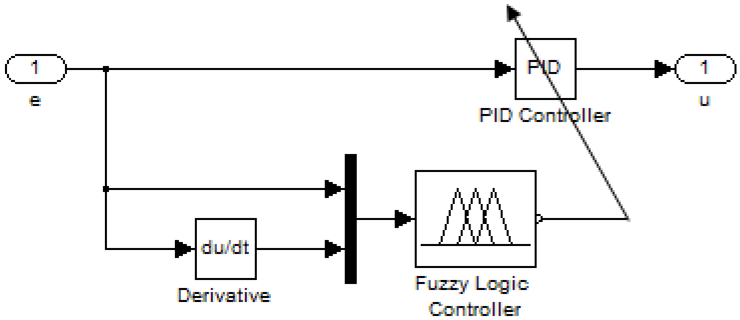

Figure 16. Gain Scheduling Fuzzy PID controller

On the other hand, Hybrid Fuzzy PID controller has two parts; conventional PID controller and Fuzzy PID controller. The classical PID and Fuzzy PID controllers are combined by a blending mechanism that depends on a certain function of actuating error. Moreover, an intelligent switching scheme is induced on the blending mechanism that makes a decision on the priority of the two controller parts. Figure 17, shows the Hybrid Fuzzy PID controller configuration.

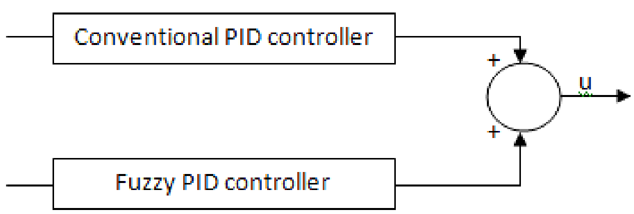

Figure 17. Hybrid Fuzzy PID controller

3.6 Fuzzy Rules Synthesis

In this part, a novel technique will be proposed to synthesize the rule base for the Fuzzy Logic (FL) controller or the Fuzzy PID based controller. There are many techniques in literature that investigate this problem. However, since Lyapunov based technique provides rules synthesis and stability analysis then it will be used for comparison basis with this proposed technique.

3.6.1 Fuzzy Lyapunov Rules Synthesis

For a given control system, stability is usually the most important attribute to be determined. The Lyapunov method of stability analysis that was introduced by the Russian mathematician A.M. Lyapunov is, in principal, the most general method for determining of stability of nonlinear systems as well as linear systems.

Lyapunov direct method relies on the construction of an energy like positive definite function F, called Lyapunov function with all the desired properties: bounded below (with a minimum at the equilibrium), and decreasing along states trajectories (i.e. when the dynamical system attains stability it's energy levels drop). If such a function is found then stability can be inferred, (Abdalla, 2009).

The strength of the Lyapunov idea lies in the fact that a conclusion about stability can be reached without precise knowledge or even being able to compute the trajectories of the system (i.e. qualitative analysis). Indeed all we need to establish is that the scalar valued Lyapunov function is decreasing along the evolutions of the system. This can be established without knowing the solutions by computing the derivative of the Lyapunov function along the solutions and check if it is a negative definite function (i.e. energy is dropping), (Abdalla, 2009).

The Lyapunov idea can be extended to the case of fuzzy controllers designed to guarantee closed loop stability. Zadeh (1996) proposed the first Fuzzy Lyapunov synthesis that is based on transforming classical Lyapunov synthesis from the domain of exact mathematical quantities and symbols to the domain of computing with words.

Fuzzy Lyapunov synthesis followed the classical Lyapunov synthesis method by constructing a Lyapunov function candidate F and then determining the conditions required to make it indeed a Lyapunov function of the closed loop system (that is, Conditions that guarantee $\frac{dF}{dt} \leq 0$). It turns out that, because of assuming fuzzy knowledge about the plant to be controlled, the derived conditions can be stated as fuzzy If-Then rules. These fuzzy rules constitute the rule base for the fuzzy controller, (Margaliot et al., 1999).

In this work, the Fuzzy Lyapunov synthesis was used to design a trajectory tracking controller. This methodology uses a Lyapunov function candidate to obtain the rules of the Mamdani type Fuzzy PID controller, then the rules are implemented to track a desired trajectory. The first step is to find a positive definite Lyapunov function, based on the error and the derivative change of the error, secondly computing the derivative of that function. Now, based on the conditions that guarantee negative definiteness of $\frac{dF}{dt} \leq 0$ and by noting that the second derivative of the error is proportional to the output of the controller, the set of fuzzy IF-Then rules may be constructed. In the Fuzzy PID controller application chapter, this technique on an elevator application example will be demonstrated.

3.6.2 Fuzzy Rules Synthesis via Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a population based stochastic optimization technique developed by Eberhart and Kennedy (1995), inspired by social behavior of bird flocking or fish schooling.

In PSO, the potential solutions, called **particles**, fly through the problem solution space by following the current optimum particles. A major advantage of PSO is its ability to handle optimization problems with multiple local optima reasonably well and its simplicity of implementation. Also, it does not require gradient information of the objective function being considered, only its values. PSO is proving itself to be an efficient method for several optimization problems, and in certain cases it does not suffer from the problems encountered by other Evolutionary Computation techniques. PSO has been successfully applied in many areas: function optimization, artificial neural network training and fuzzy system control.

PSO simulates a commonly observed social behavior, where members of a group tend to follow the lead of the best of the group. The procedure of PSO is illustrated as following steps, (Abdalla, 2009):

i. **Initialization:** Randomly generate a population of the potential solutions, called "particles," and each particle is assigned a randomized velocity.

ii. **Velocity Update**: The particles then "fly" through search hyperspace while updating their own velocity, which is accomplished by considering its own past flight and those of its companions.

The particle's velocity and position are dynamically updated by the following equations:

$$v_{id}^{NEW} = w_i . v_{id}^{OLD} + C_1 . r_1 . \left(x_{pd} - x_{id}^{OLD}\right) + C_2 . r_2 . \left(x_{gd} - x_{id}^{OLD}\right) \qquad (27)$$

$$x_{id}^{NEW} = x_{id}^{OLD} + v_{id}^{NEW} \qquad (28)$$

Where the acceleration coefficients C_1 and C_2 are two positive constants; w_i is an inertia weight and r_1, r_2 are a uniformly generated random numbers within the range [0, 1], which is generated every time for each iteration. Eberhart, et al. (2001) and Hu, et al. (2001) suggested using $C_1=C_2=2$ and w_i =0.5+rand/2. Equation (27) shows that, when calculating the new velocity for a particle, the previous velocity of the particle (v_{id}), their own best location that the particles have discovered previously (x_{id}) and the global best location (x_{gd}) all contribute some influence on the outcome of velocity update.

The global best location (x_{gd}) is to be identified, based on its fitness, as the best particle among the population. All particles are then accelerated towards the global best particle as well as in the directions of their own best solutions that have been visited previously. While approaching the current best particle from different directions in the search space, all particles may encounter by chance even better particles in route, and the global best solution will eventually emerge. Equation (28) shows how each particle's position (x_{id}) is updated in the search of solution space.

To summarize the PSO algorithm a flow chart is provided in Figure 18. The flow chart describes the PSO algorithm in an abstract way.

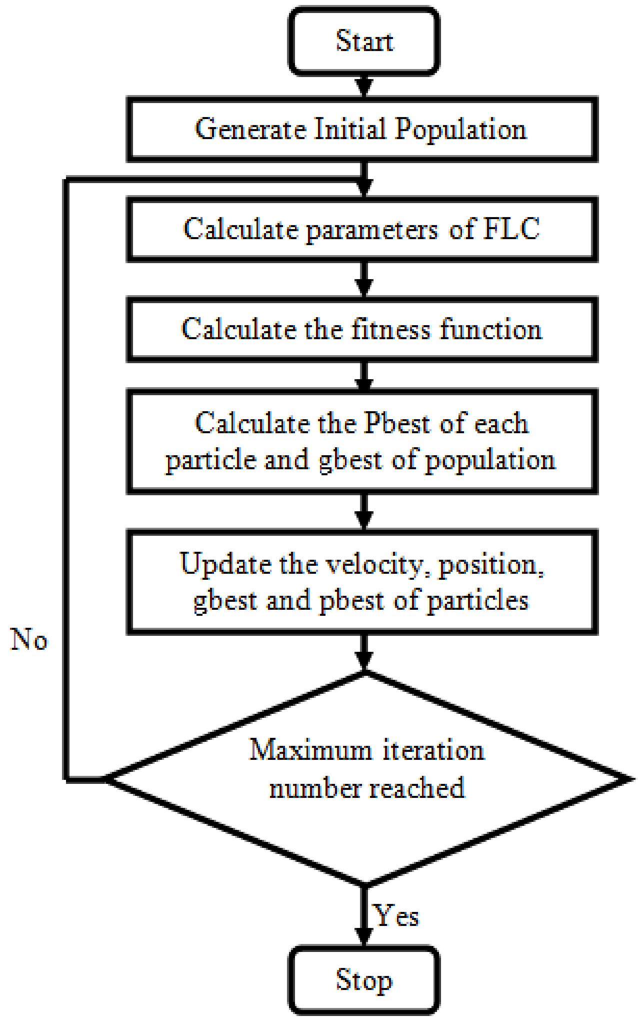

Figure 18. Particle Swarm Optimization algorithm

Generating of the knowledge base of a fuzzy rule base system presents several difficulties because the knowledge base depends on the nature of application in hand, and this makes the accuracy of the fuzzy rule base system directly depends on its composition.

The usual solution for improving the fuzzy rule base system performance in dealing with the data base components involves a tuning process of the preliminary data base definition once the rule base has been derived. This process only adjusts the membership function definitions and doesn't modify the number of linguistic terms in each fuzzy partition since the rule base remains unchanged. The objective of this work is to introduce a method to automatically generate the knowledge base of a fuzzy rule base system based on learning approach.

In the weighted fuzzy rule based system, the rules in its rule base are endued with weights which will be optimized by the Particle Swarm Optimization (PSO). These rules could be arranged according to their weights; some rules with low weights would be deferred or abandoned, while some high weighted rules would be executed firstly. Through ranking the weights, the rules could be ranked and get a simplified rule base. Figure 19 illustrates the new structure of the Fuzzy controller that will be used.

The weighted fuzzy rule based system methodology will be further treated and demonstrated using an elevator application example in subsequent chapters.

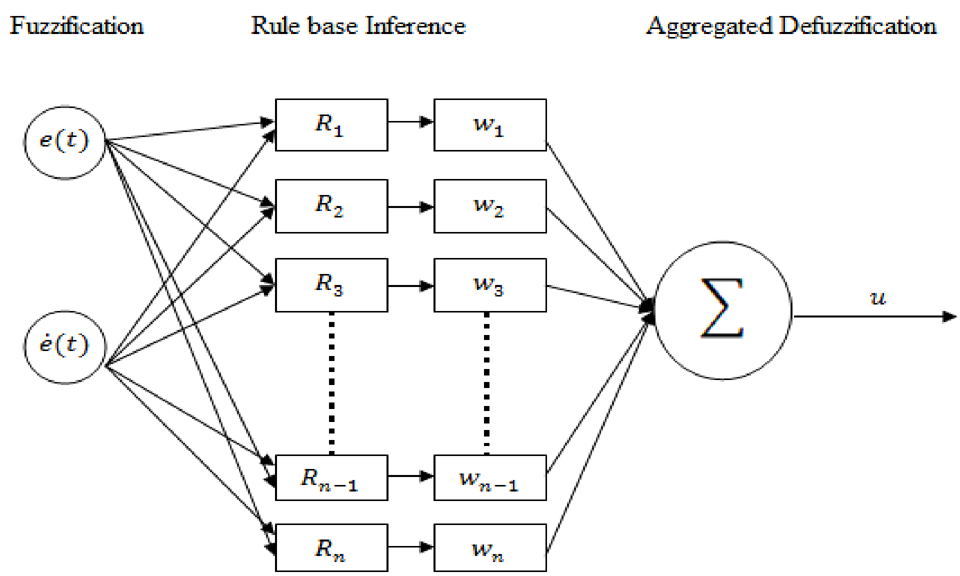

Figure 19. The new structure of the Fuzzy Logic controller

The output of the Fuzzy Logic (FL) controller is then obtained by the average of the weighted firing of the rules as follows:

$$u^* = \frac{\sum_{i=1}^{m} we_i \times \mu_{ik} \times c_i}{\sum_{i=1}^{m} we_i \times \mu_{ik}} \quad (29)$$

Where we_i is the weight of the rule. Now, to obtain the parameters of the function of the weighted fuzzy rule-base system described in Equation 29, the Particle Swarm Optimization (PSO) algorithm can be used to estimate the parameters of the weighted fuzzy rule-based system, which include the position and the shape of the membership function, the fuzzy rules and its weights. The implementation procedure of Particle Swarm Optimization (PSO) Algorithm was shown previously in Figure 18.

The complete algorithm will be fully illustrated by means of an example in the Fuzzy Logic Proportional-Integral-Derivative (FL PID) controller application chapter.

CHAPTER 4

Fuzzy PID Controller Application

In this part a functioning Fuzzy PID controller will be designed and simulated to control an elevator system. First, the elevator as a test-bed will be introduced, that is a complete mathematical model will be adopted and later it will be used for simulation purposes only. Secondly, the building blocks of the Fuzzy PID controller will be designed.

4.1 Overview of Elevators

Elevators may be classified according to their driving method into three categories, Ramsey and Sleeper (2007); electric, hydraulic and pneumatic elevators. Hydraulic elevators use hydraulic oil driven machine to raise and lower car and its load, however lower speeds and piston length (stroke) restricts the use of this type. Electric traction elevators are elevators in which the energy is applied by means of an electric driven machine. Medium to high speeds and virtually limitless rise allow this elevator type to serve high-rise, medium-rise and low-rise buildings. Electric traction elevators can be further divided into geared and gearless categories: geared traction elevators designed to operate within the general range of 100 to 450 ft/min, restricting their use to medium rise buildings, and gearless traction elevators that available in units with speeds of 500 to 1500 ft/min. They offer the advantages of a long life and smooth ride, (Ramsey and Sleeper, 2007).

Typically, elevator machines are either roped with a single or double wrap arrangement. Single wrap arrangement provides traction by the use of grooves that will pinch the ropes with varying degrees of pressure depending on the groove's shape and it's undercutting. The most effective single-wrap arrangement gives 180 degrees of the rope contact with the sheave without deflecting the sheave. On the other hand, double-wrap arrangement provides greater traction than the single wrap arrangement and is used in many automatic high speed installations.

Conventional elevators are either roped as 1:1 or 2:1 rope arrangement for both car and counter-weight. A 1:1 roping arrangement gives no mechanical advantage, while 2:1 roping permits the use of a high speed, low-power and therefore lower cost traction machine.

The most popular electrical elevator models based on roping techniques are shown in Figure 20. For a complete and thorough discussion of such schemes the reader is directed to consult some elevator design based handbook.

In this work double wrap 2:1 gearless traction elevator will be used as a test-bed for the controllers design.

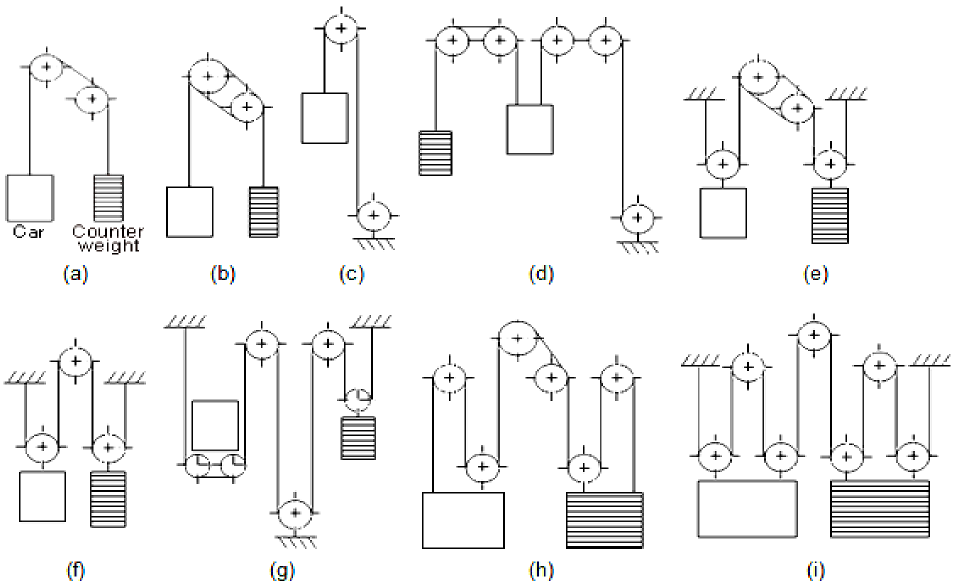

Figure 20. The most popular electrical elevator models based on roping techniques
a) 1:1 Half wrap b) 1:1 Full wrap c) 1:1 Drum winding
d) 1:1 Drum winding e) 2:1 Full wrap f) 2:1 Half wrap
g) 2:1 Half wrap h) 3:1 Half wrap i) 4:1 Half wrap

The long-life, smoothness and high horsepower of gearless traction elevators provide a durable elevator service that can outline the building itself. The first high-rise application of gearless traction elevator was in the Beaver building New York City in 1903, which was followed by such notable installations such as the singer building which was demolished in 1972 and the Woolworth buildings, to name few. On higher speed gearless traction machines of 4 mps or more, the double wrap principle is generally applied to obtain traction and to minimize rope wear. The 2:1 arrangement allows the use of a higher speed, and therefore a smaller, but faster elevator. The economy of the faster motor, which can be built smaller and lighter than lower speed DC motor, also makes 2:1 roping alternative for a full range of speed requirements from 0.5 to 3.5 mps or more and for any lifting capacity. Gearless machines are greatly capable of acceleration rates of 1.2 mps^2 and can be made to accelerate faster. The limiting factor is not the accelerating rate, but the rate of change of acceleration "Jerk" that is felt by the riding passenger. This is a matter of personal tolerance, but in general, the upper limit of 2.4 mps^3 is usually the maximum, (Strakosch, 1998).

A typical speed profile of an elevator car -for the first floor- is shown in Figure 21. The speed profile describes the motion status of the car. When a car starts to move, it enters acceleration mode until it reaches the contract speed. This speed is maintained up to when the car has to stop. Before the car reaches the stop position, it has to slow down for a safe stop at the destination floor. Besides the motion status of the car, other useful information given by the speed profile includes the time the car takes to reach the contract speed, the time the car spends to travel one floor at contract speed, the time taken to decelerate before the car stops, the distance traveled to reach the contract speed and the distance traveled to slow down from the contract speed before the car stops.

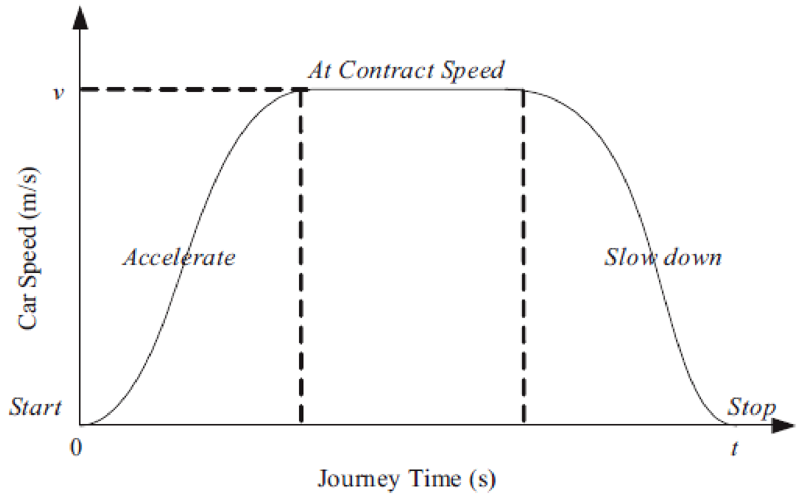

Figure 21. Typical speed profile for an elevator system

4.2 Elevator Modeling

Modeling is the process of formulating mathematical equations, which are used to describe the system's dynamical behavior. It involves identifying the system elements, inputs and outputs, the physical mechanisms and laws governing the behavior of these elements.

In this work the controller's design will be verified using computer simulations. All simulation results in this work are based on the double wrap 2:1 gearless electric (DC) traction elevator physical model that is depicted in Figure 22. Some assumptions were made in the development of this model; the hoisting ropes are mass-less, the dynamics of the compensating and the governor roping are ignored and the break system is excluded from the model; that is the motor will breakdown the system.

A summary of the ODE of the elevator mathematical model is provided here as a reference (Boutler, 2000):

$$\dot{i_a} = \frac{V_{in}}{L_a} - \frac{R_a}{L_a} - \frac{K_b}{L_a}\dot{x}_2 \quad (30)$$

$$\ddot{x}_2 = \frac{R_2}{J_2}(T_3 - T_2) + K_m i_a \quad (31)$$

$$\ddot{x}_1 = \frac{R_1^2 K_{EL}}{2J_1}(x_{EL} - x_1) + \frac{R_1^2 B_{EL}}{2J_1}(\dot{x}_1 - \dot{x}_{EL}) + (X_0 K_0 + T_2)\frac{R_1^2}{2J_1} \quad (32)$$

$$\ddot{x}_{EL} = \frac{K_{EL}}{M_{EL}}(x_1 - x_{EL}) + \frac{B_{EL}}{M_{EL}}(\dot{x}_1 - \dot{x}_{EL}) \quad (33)$$

$$\ddot{x}_{CW} = \frac{K_{CW}}{M_{CW}}(x_3 - x_{CW}) + \frac{B_{CW}}{M_{CW}}(\dot{x}_3 - \dot{x}_{CW}) \quad (34)$$

$$\ddot{x}_3 = \frac{K_{CW} R_3^2}{2J_3}(x_{CW} - x_3) + \frac{B_{CW} R_3^2}{2J_3}(\dot{x}_{CW} - \dot{x}_3) + (T_3 + T_2)\frac{R_3^2}{2J_3} \quad (35)$$

Where i_a is the armature current, V_{in} is the armature voltage, R_a is the armature resistance, L_a is the armature inductance, K_b is the voltage constant, x_2 is the drive sheave position, x_1 is the elevator sheave position, $x_{EL}$is the elevator car position, x_3 is the counter weight sheave position, x_{CW} is the counter weight position, K_{CW} is the stiffness factor for the counter weight, K_{EL} is the stiffness factor for the elevator car, B_{EL} is the damping coefficient for the elevator car, B_{CW} is the damping coefficient for the counter weight, T_i is the tension, V_i is the speed, H_i is the height, τ_m is the motor torque, J_1 is the moment of inertia for the elevator sheave, J_2 is the moment of inertia for the drive sheave, J_3 is the moment of inertia for the counter weight sheave, R_1 is the radius for the elevator sheave, R_2 is the radius for the drive sheave, R_3 is the Mass for the counter weight sheave, M_{EL} is the mass for the elevator car and M_{CW} is the mass for the counter weight.

The elevator system's model will be simulated using Simulink / Matlab for testing the designed controllers. A switching technology through Pulse Width Modulation (PWM) and a universal bridge is used for enabling speed regulation of the PMDC motor system. Figure 23 depicts the closed loop system (feedback) of the major blocks, while Figures 24 and 25 illustrate the details of the subsystems. The speed profile is given as an input for the controller in addition to the desired height (floor level).

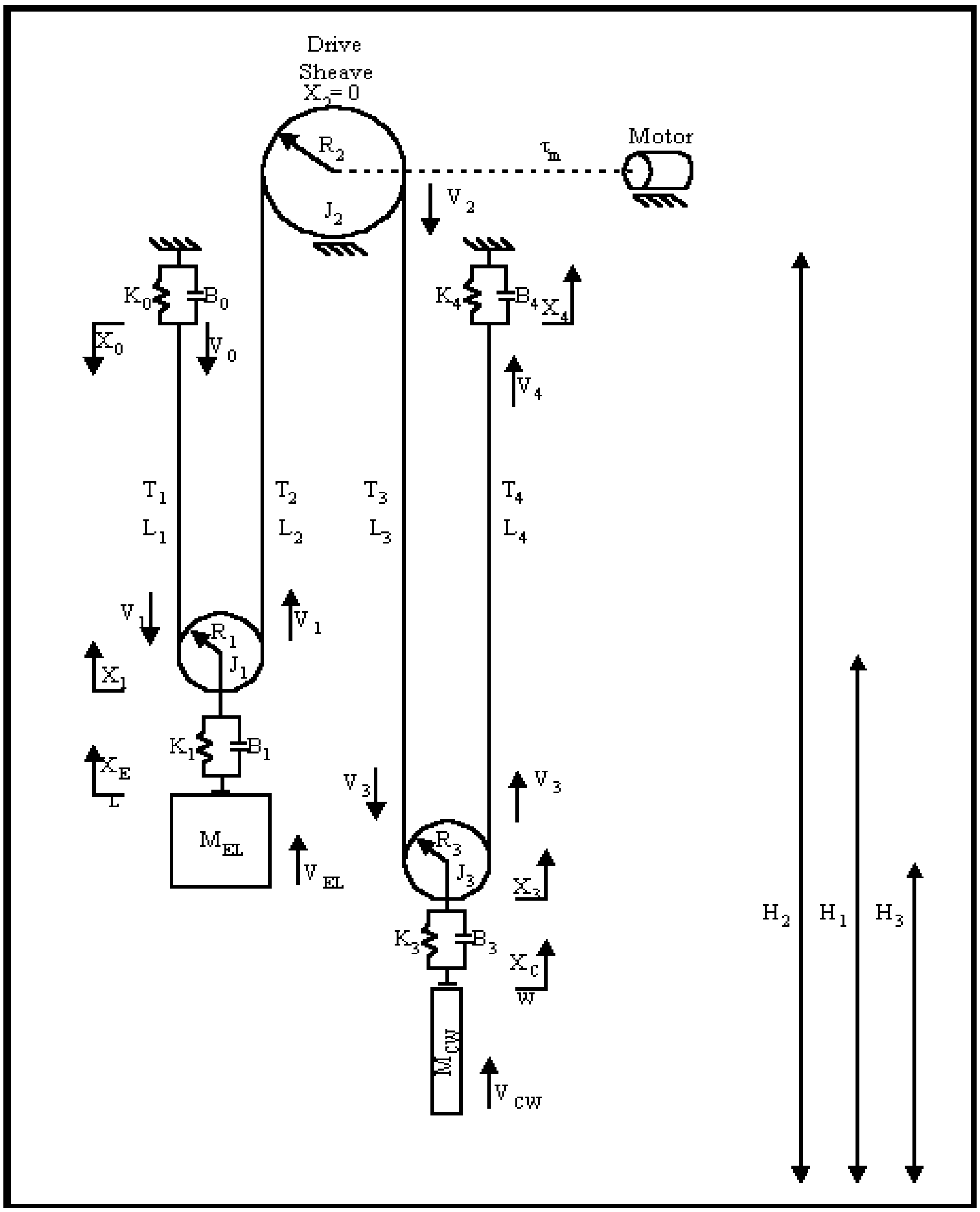

Figure 22. The double wrap gearless traction elevator physical model

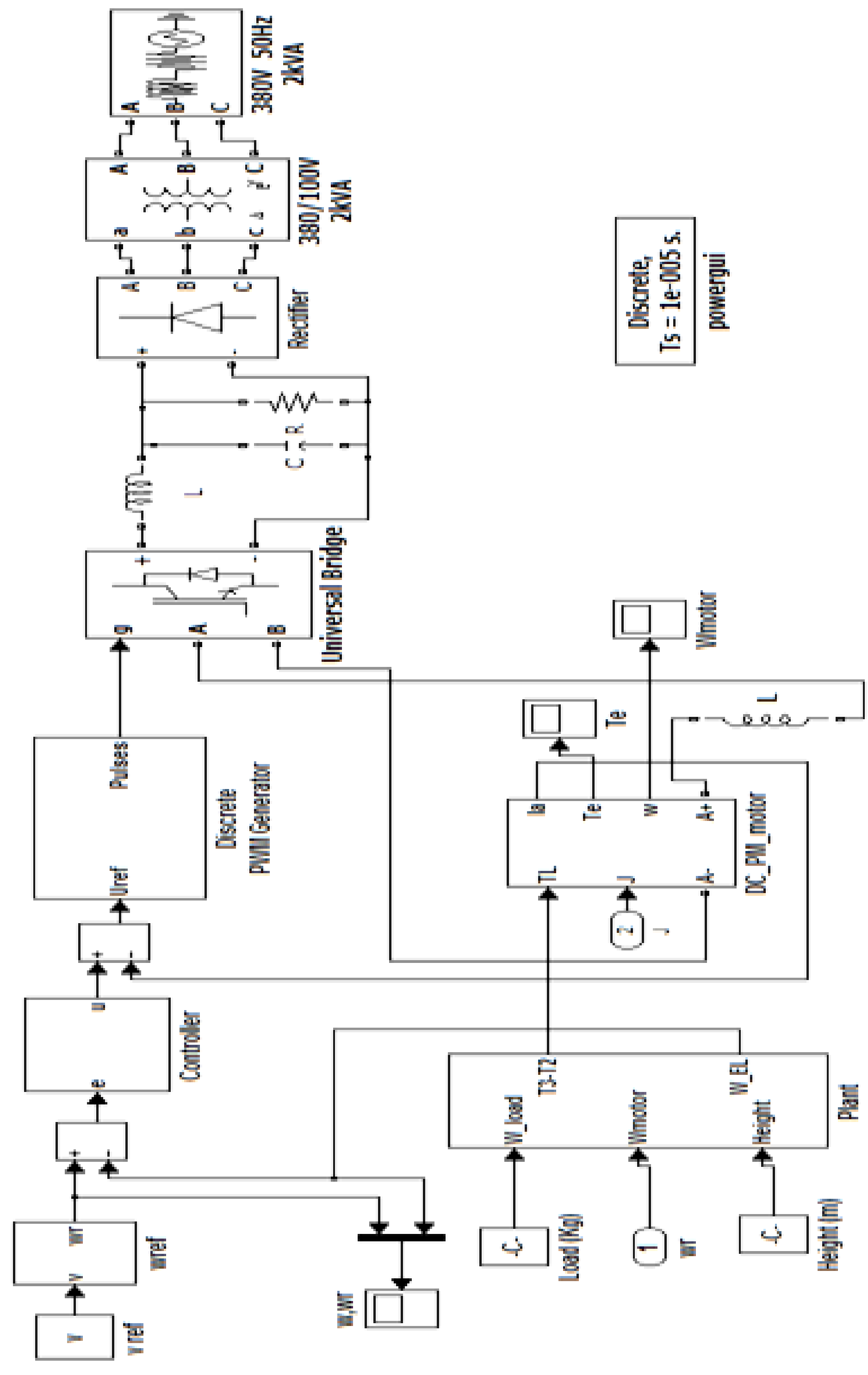

Figure 23. Elevator’s speed control closed loop system

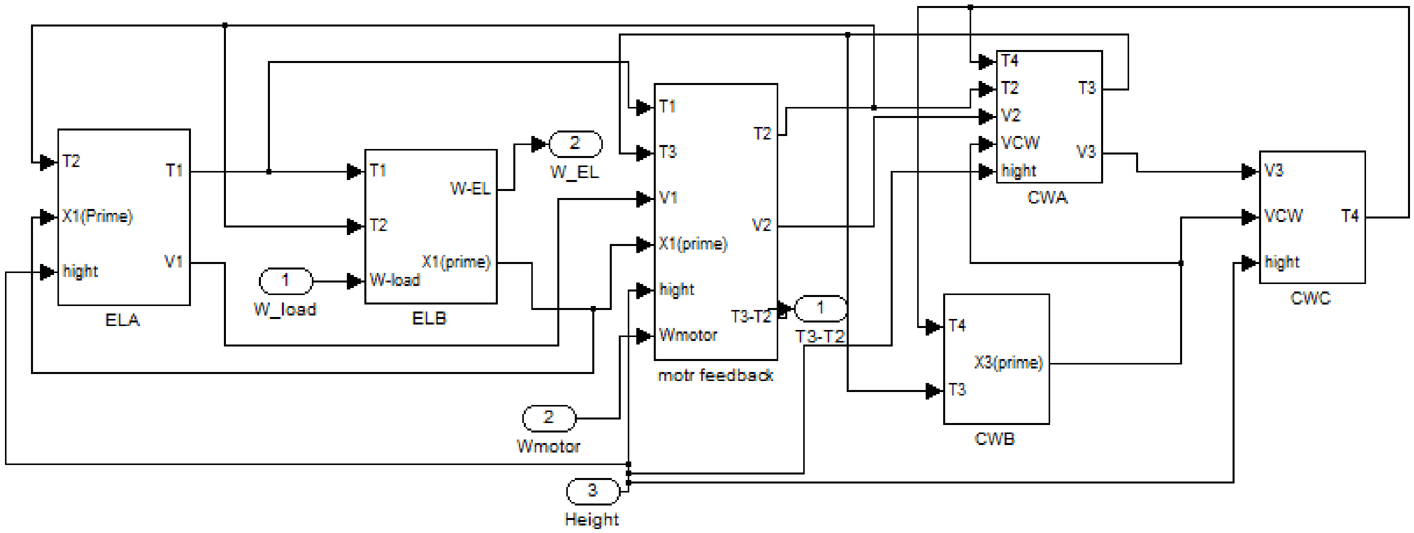

Figure 24. Elevator’s internal subsystems

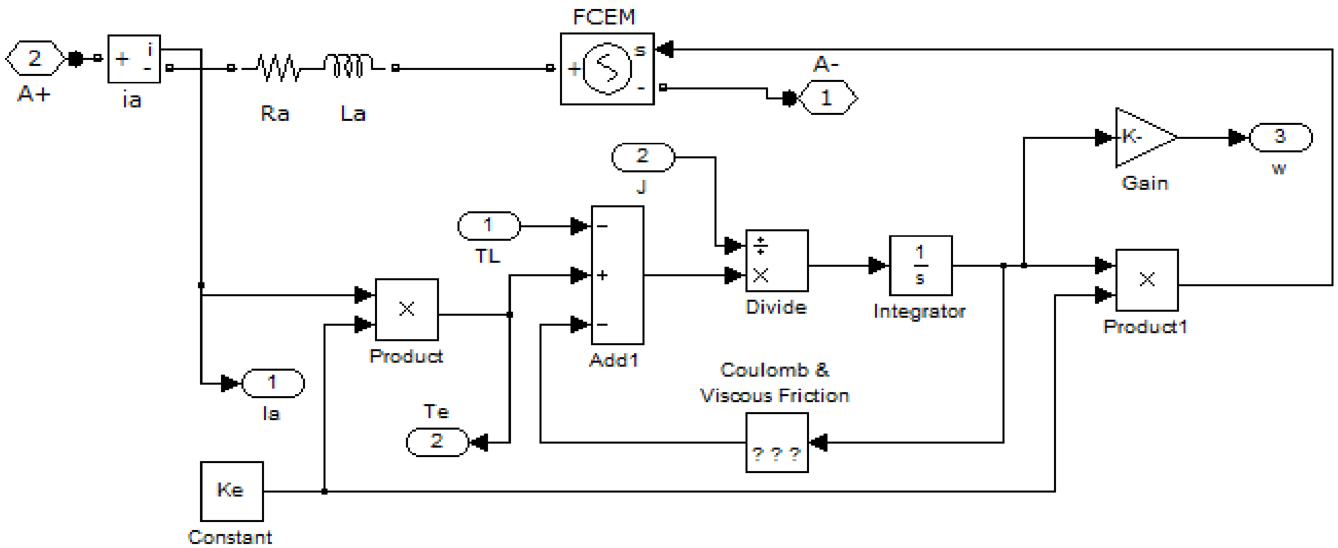

Figure 25. Permanent Magnet Direct Current (PMDC) motor subsystem

4.3 Elevator Fuzzy PID Controller Design

The Fuzzy PID controller will directly replace the conventional PID controller in the closed loop system. However, all the flexibility and high performance of the fuzzy portion will enhance the capabilities of the PID controller and it will enrich its trajectories space. Figure 26 depicts the closed loop system with the proposed Fuzzy PID controller in the forward path.

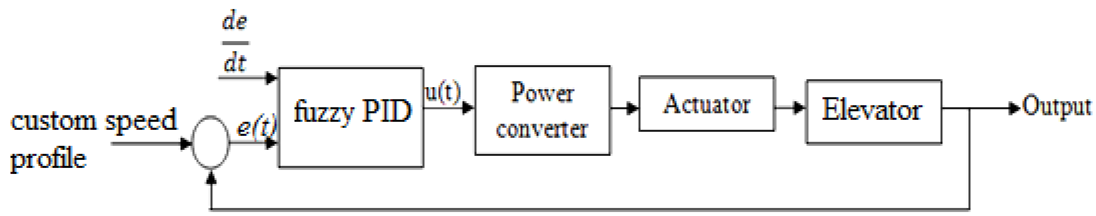

Figure 26. Fuzzy PID controller loop

The proposed direct action Fuzzy PID controller is depicted in Figure 27. The integral parameter (K_i) is chosen using numerical optimization. The other part of the Fuzzy PID controller is the PD, which is implemented using pure Fuzzy Logic (FL) techniques. The major steps in the Fuzzy PID design constitutes creating a knowledge base of the rules (rule base and inference), establishing membership functions for the inputs (fuzzification) and implementing member functions for the outputs (defuzzification).

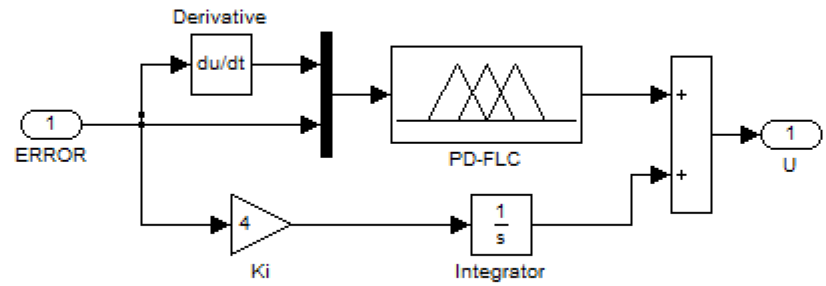

Figure 27. Proposed Fuzzy PID controller

In this work the sensed input signals that are fed to the Fuzzy PID controller are the error and the rate of change of the error ($e(t)$,$\dot{e}(t)$).

$$e(t) = V_d(t) - V(t) \tag{36}$$

Where $V_d(t)$ is the reference speed (speed profile), and $V(t)$ is the elevator's actual speed. Typical speed profile that we have adopted is illustrated in Figure 28.

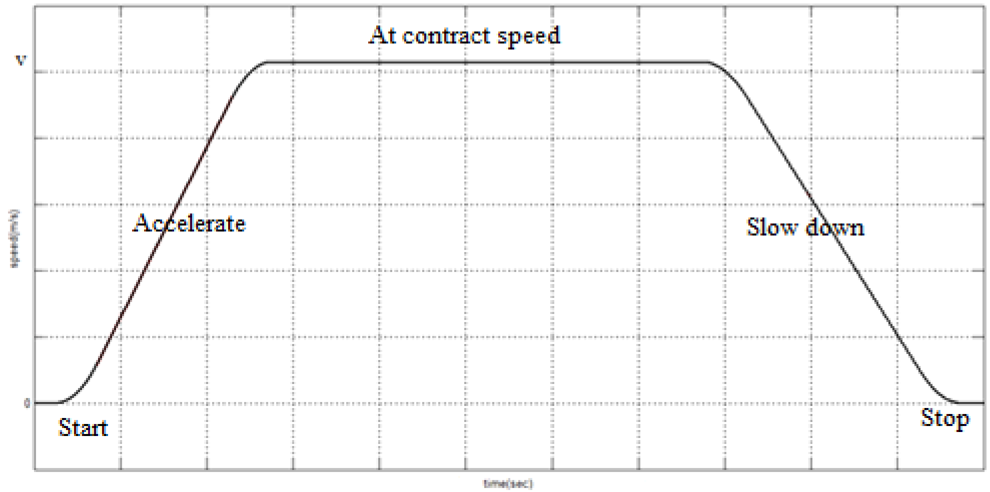

Figure 28. The speed profile for an elevator system

4.3.1 Lyapunov Fuzzy PID Controller Design

During recent years considerable efforts have been devoted to guarantee the stability of the Fuzzy Logic controller. Several stability analysis methods have been established, and stable control designs have been introduced, (Woo et. al, 2007), (Tanaka et al., 2003), (Tanaka et al., 2007), (Yeh et al., 2008) and (Chen et al., 2006). The fuzzy Lyapunov function approach has been proposed to guarantee the stability of the Fuzzy Logic controller.

To illustrate the Lyapunov based Fuzzy Logic PID controller, the control of an elevator system will be considered. Typically, the three stages of the Fuzzy Logic (FL) controller (illustrated in Figure 29) will be presented as follows

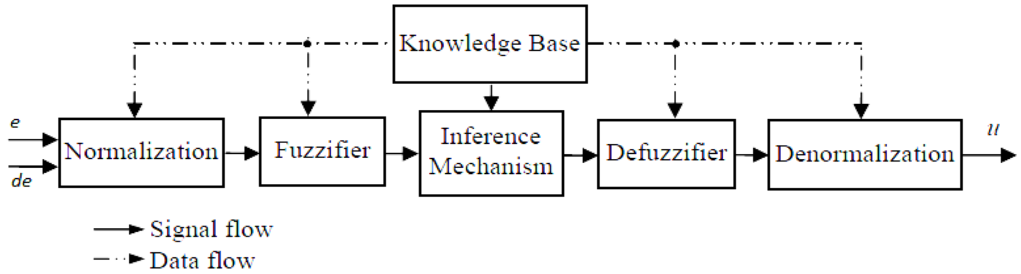

Figure 29. The three stages of the Fuzzy Logic controller

Fuzzification: The membership function is a graphical representation of the magnitude of participation of each input. There are different membership functions associated with each input and output response. In this work, both triangular and trapezoidal membership functions for inputs and output variables were used due to their simplicity and effectiveness of implication. The number of membership functions, to some extent, determines the quality of control, that can be achieved using the Fuzzy Logic controller. As the number of membership functions increase to a certain limit, the quality of control may reach Platue.

The membership functions for the two inputs, namely the error $e(t)$and the rate of change of the error $\dot{e}(t)$, and the output $u(t)$, namely the reference voltage for the discrete PWM generator, are shown in Figure 30. Seven linguistic variables were selected to span the whole input/output range, which are fully defined and their ranges of the linguistic variables set as shown in Table 3.

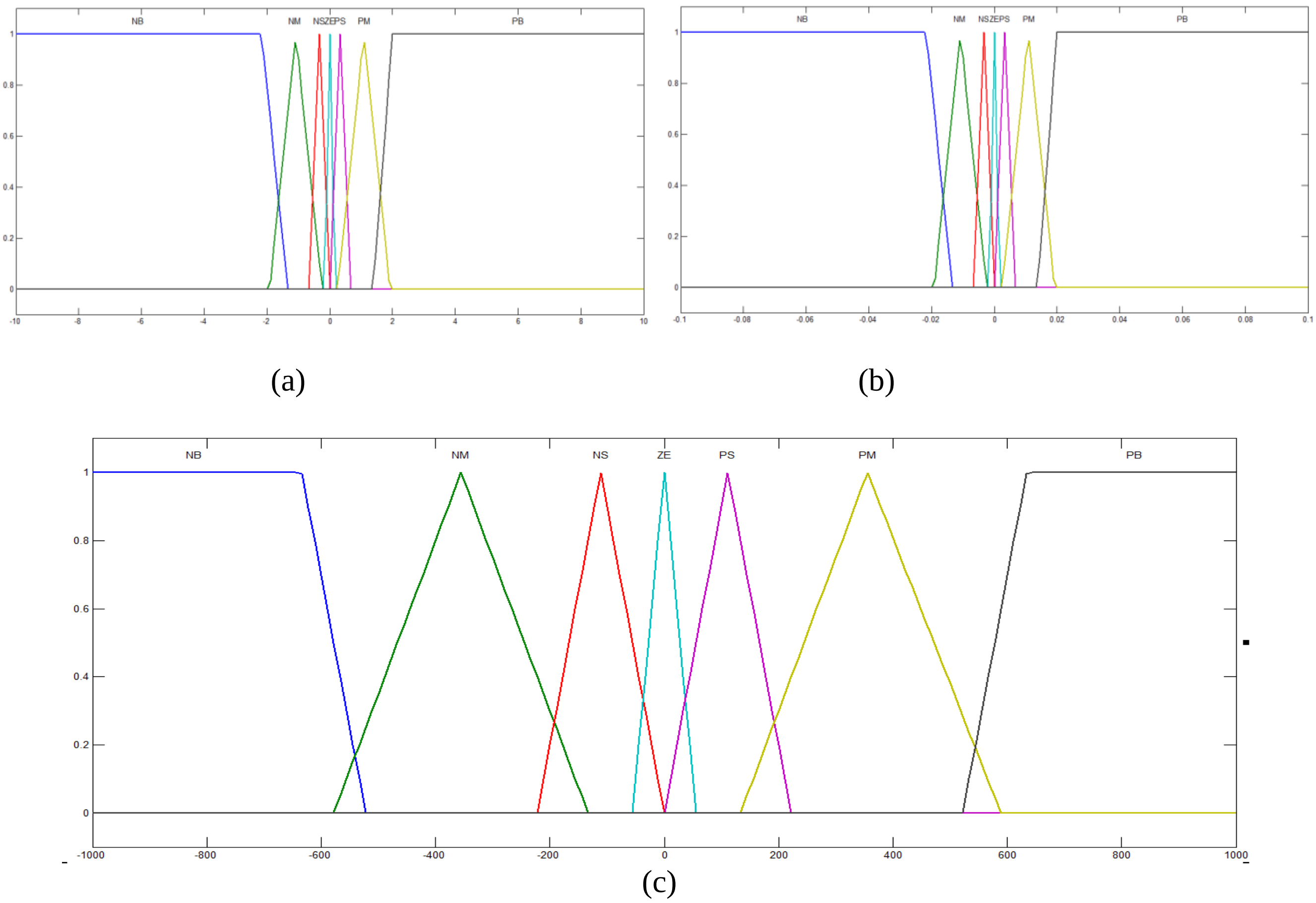

Figure 30. The membership functions (Lyapunov based) (a) membership functions for the error (b) membership functions for the change of error (c) membership functions for the reference voltage for the discrete PWM generator.

Table 3. Membership functions liguistic variables ranges

Variable	Linguistic variable	Range
$\boldsymbol{e}$(t)	Negative Big (NB)	[-10 -10 -2.2 -1.3]
	Negative Medium (NM)	[-1.9 -1.1 -0.25]
	Negative Small (NS)	[-0.7 -0.3 0]
	Zero (ZE)	[-0.17 0 0.17]
	Positive Small (PS)	[-0.17 0 0.17]
	Positive Medium (PM)	[-0.17 0 0.17]
	Positive Big (PB)	[1.4 2 10 10]
$\dot{\boldsymbol{e}}$(t)	Negative Big (NB)	[-0.1 -0.1 -0.02 -0.01]
	Negative Medium (NM)	[-0.02 -0.01 -0.003]
	Negative Small (NS)	[-0.007 -0.003 0]
	Zero (ZE)	[-0.002 0 0.002]
	Positive Small (PS)	[0 0.003 0.007]
	Positive Medium (PM)	[0.002 0.01 0.02]
	Positive Big (PB)	[0.01 0.02 0.1 0.1]
$\boldsymbol{u}$(t)	Negative Big (NB)	[-1000 -1000 -634 -522]
	Negative Medium (NM)	[-578 -356 -133]
	Negative Small (NS)	[-222 -111 0]
	Zero (ZE)	[-56 0 56]
	Positive Small (PS)	[0 111 222]
	Positive Medium (PM)	[133 355 588]
	Positive Big (PB)	[522 634 1000 1000]

Rule base: The main part of the Fuzzy PID controller is the rule base and the inference mechanism. The rule base is normally expressed in a set of fuzzy linguistic rules, with each rule triggered (fired) with varying belief for support.The i^{th} linguistic control rule can be expressed as: R_i: If e_i is A_i and de_i is b_i then u_i is C_i, Where A_i, B_i and C_i are fuzzy variables (sets) characterized by fuzzy membership functions.

In this work, the fuzzy Lyapunov synthesis was used to design a trajectory tracking controller. This methodology uses a Lyapunov function candidate to obtain the rules of the Mamdani-type Fuzzy Logic controller, which are later implemented to track a user's trajectory. Here, a continuously differentiable function $F(x), x = [e\ \dot{e}]^T$ will be referred as a Lyapunov function if the following requirements are met:

i. $F(x) = 0$, for only x=0,

ii. $F(x) > 0, x \in R^2 - \{0\}$,

iii. $\frac{dF}{dt} < 0, x \in R^2 - \{0\}$,

were $R^2 - \{0\}$ is some neighborhood of zero excluding the origin itself.

Assuming that the reference speed V_d and its derivatives $\dot{V}_d$ and $\ddot{V}_d$ are bounded and available to the controller, let's choose $F(x)$ as Lyapunov function of a quadratic form:

$$F(x) = \frac{1}{2}(e^2(t) + \dot{e}^2(t)) \tag{37}$$

Where the error is given as:

$$e(t) = V_d(t) - V(t) \tag{38}$$

It is obvious that conditions i and ii hold for the proposed $F(x)$ and the only need is to check condition iii to verify stability. Hence, the derivative of the Lyapunov function yields

$$\dot{F}(x) = \dot{e}(t)(e(t) + \ddot{e}(t)) \tag{39}$$

As the second derivative of error proportional to the output of the FL controller (u), then

$$\dot{F}(x) = \dot{e}(t)(e(t) + u(t)) \tag{40}$$

The corresponding requirement of Lyapunov stability becomes

$$\dot{F}(x) = \dot{e}(t)(e(t) + u(t)) < 0 \tag{41}$$

Consequently one may draw conclusions from three possible combinations depending on the sign of the error:

- **Case 1**: If $e(t)$ and $\dot{e}(t)$ are both positive then $u(t) < -e(t)$
- **Case 2**: If $e(t)$ and $\dot{e}(t)$ are both negative then $u(t) > -e(t)$
- **Case 3**: If $e(t)$ and $\dot{e}(t)$ have opposite signs then $u(t) = 0$

Consequently, these combinations may be used to generate the fuzzy rules that are illustrated in Table 4.

Table 4. Lyapunov based FL PID controller rules

e	*NB*	*NM*	*NS*	*ZE*	*PS*	*PM*	*PB*
NB	PB	PB	PB	PB	PM	PS	PS
NM	PB	PB	PB	PB	PB	PM	PM
NS	PB	PB	PB	PB	PB	PB	PB
ZE	PB	PM	PB	ZE	NB	NM	NB
PS	NB	NB	NB	NB	NB	NB	NB
PM	NM	NM	NB	NB	NB	NB	NB
PB	NS	NS	NM	NB	NB	NB	NB

The generated Lyapunov based Fuzzy Logic PID controller surface is shown in Figure 31. Typically, a control surface is a plot that shows the controller's control signal as a function of controller's inputs, which provides a sense of the controller's trajectories for a two input based systems. In this case, this plot shows the control signals (reference voltage for the discrete PWM generator) as a function of the error and the rate of change of the error.

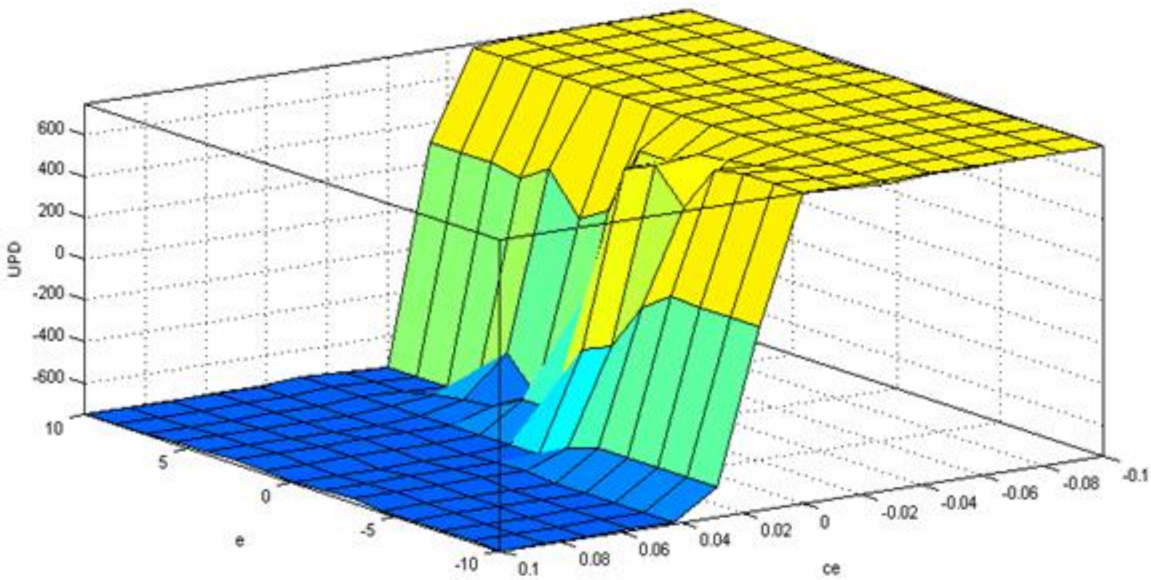

Figure 31. Lyapunov based Fuzzy PID controller control surface

Defuzzification: According to real world requirements, the linguistic variables have to be transformed to crisp outputs in order to be utilized by the actuator. Classical centre of gravity method is one of the best known defuzzification methods and it will be used in this work.

$$u^* = \frac{\sum_{i=1}^{m} \mu_{ik} \times c_i}{\sum_{i=1}^{m} \mu_{ik}} \tag{42}$$

Where μ_{ik} denotes the degree that instance matches the rule R_i , and $c_i \epsilon$ { C1, …. C_M} is the centroid of every fuzzy set.

4.3.2 PSO Based Fuzzy PID Controller Design

To illustrate the Particle Swarm Optimization (PSO) based Fuzzy Logic PID controller design we will consider the control of an elevator system. In this design methodology one should first generate the rule base (stage II) based on optimization binary PSO search then the fuzzification stage (stage I) will be implemented. This is counter intitutive and differs from classical Fuzzy Logic controller design. The reason behind this is the fact that we do not have off hand all the linguistic variables, instead they are generated by the PSO search results. So let's shed more light on rule base auto-generation.

Rule base: Traditionally, the construction of Fuzzy Logic controller, rules has been mainly based on the operator's control experience or actions. Unfortunately, acquiring rules from experts is not an easy task, moreover it is very difficult for a knowledge engineer to extract rules from static data bases (data mining). On the other hand, selecting a set of important fuzzy rules from a given rule base is an important issue in fuzzy rule base modeling. Even though it is conceivable that eliminating redundant or less important fuzzy rules from the rule base can result in a compact fuzzy model with better generalizing ability, the decision as to which rules are redundant or less important is also not an easy task.

In this work, the Particle Swarm Optimization (PSO) binary search algorithm is applied in two stages; in the first stage the PSO algorithm is used for generating a firsthand optimal decision learning rules while in the second stage it is used for obtaining a simplified set of rules that is generated from the first stage. Detailed descriptions of these two stages are given as follows:

First stage: The fuzzy decision rules of the Fuzzy Logic PID controller are not established by expert knowledge or operator's experience but instead are constructed using the PSO binary search algorithm.

The technique over the previously discussed elevator system application will be demonstrated. The elevator's proposed Fuzzy Logic PID controller has two inputs and one output, the inputs are the error and the rate of change of the error, the output is the reference voltage for the discrete PWM generator. Assuming, each input to have seven linguistic variables (NB, NM, NS, ZE, PS, PM and PB), accordingly, the number of rules generated is 49 (7x7) rules. Consequently, Figure 32 describes one rule (R_i) output, which means a rule might have seven possible outcomes for a control action (i.e. the output of the controller); Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (ZE), Positive Small (PS), Positive Medium (PM) and Positive Big (PB). An optimal outcome for the control action will be determined for each rule of the 49 rules by using the PSO binary search algorithm.

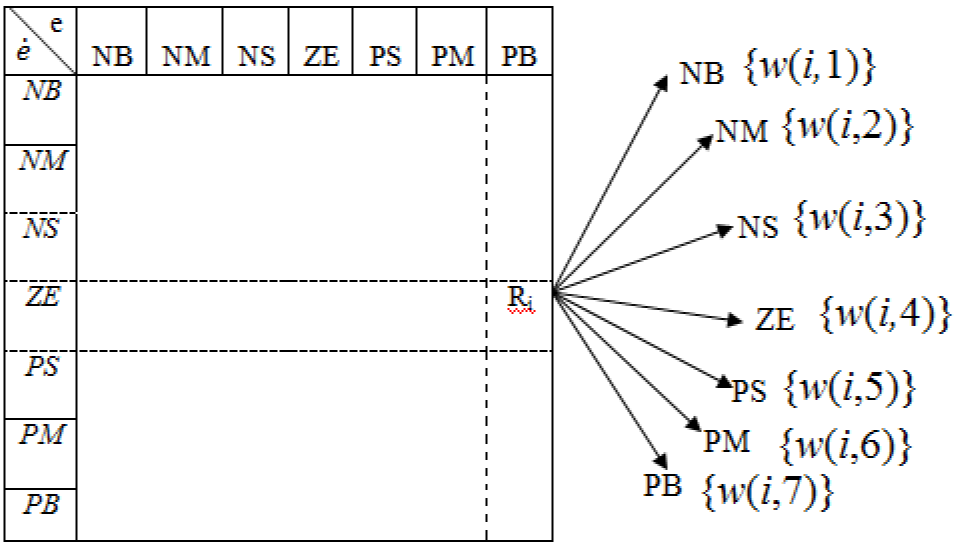

Figure 32. FL PID one rule outcome possibilities

To minimize the huge number of tuning parameters, the PSO algorithm is applied based on a binary approach (i.e. either the rule is applied or not). In this methodology the PSO algorithm will have 49 Parameters to be optimized, each parameter is taken as a decimal number and is converted into a seven digits binary number that constructs a set of weights [*w* (*i*, 1), *w* (*i*, 2), *w* (*i*, 3), *w* (*i*, 4), *w* (*i*, 5), *w* (*i*, 6), *w* (*i*, 7)] which may be described as a matrix as follows:

$$\begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ . \\ . \\ P_{49} \end{bmatrix} \rightarrow \begin{bmatrix} w(1,1), w(1,2), w(1,3), w(1,4), w(1,5), w(1,6), w(1,7) \\ w(2,1), w(2,2), w(2,3), w(2,4), w(2,5), w(2,6), w(2,7) \\ w(3,1), w(3,2), w(3,3), w(3,4), w(3,5), w(3,6), w(3,7) \\ . \\ . \\ w(49,1), w(49,2), w(49,3), w(49,4), w(49,5), w(49,6), w(49,7) \end{bmatrix}$$

Where P_i is the PSO ith rule with seven possible outcomes. This set of weights is then implemented to each possible rule as shown in Figure 32. Consequently, the set of weights for each rule will have seven cases;

- [1 0 0 0 0 0 0], in this case the ith rule optimal output for the controller will be NB.
- [0 1 0 0 0 0 0], in this case the ith rule optimal output for the controller will be NM.
- [0 0 1 0 0 0 0], in this case the ith rule optimal output for the controller will be NS.
- [0 0 0 1 0 0 0], in this case the ith rule optimal output for the controller will be ZE.
- [0 0 0 0 1 0 0], in this case the ith rule optimal output for the controller will be PS.
- [0 0 0 0 0 1 0], in this case the ith rule optimal output for the controller will be PM.
- [0 0 0 0 0 0 1], in this case the ith rule optimal output for the controller will be PB.

The optimal PSO parameters recommend list after the first stage is summarized in Table 5. These generated controller's optimal output outcomes mimics the system's operators best reaction experiences for the designated inputs.

Table 5. PSO based FL PID controller rules (First stage)

e	*NB*	*NM*	*NS*	*ZE*	*PS*	*PM*	*PB*
NB	PB	PM	PM	PB	PS	PS	PB
NM	PB	PB	PM	PB	PM	PM	PB
NS	PM	PB	PM	PB	PM	PB	PM
ZE	PB	PM	PS	ZE	NS	NM	NB
PS	NB	NB	NM	NB	NM	NM	NB
PM	NS	NM	NM	NB	NM	NB	NM
PB	NS	NM	NS	NB	NM	NB	NB

Second stage: The first stage screening of the controller's rules output was selected based on fixed set of input linguistic variables. In the second stage, the most significant input linguistic variables will be kept (i.e. the ones that contribute the most for the system's response or the most influential). In order to find those influential linguistic variables rules combination, a weighted fuzzy rule base is proposed. Again the PSO search algorithm will be used to tune these weights.

In the weighted fuzzy rule based system, every rule has a weight (number between zero and one). These rules could be arranged according to their weights; the rules with low weights would be deferred or abandoned, while the high weighted rules would be the most influential during excution. Through ranking the weights, the rules could be ranked to get a simplified rule base from the most important ones.

After applying the first stage, the optimum set of rules (49 rules) was generated using the PSO algorithm. In order to reduce this number of rules, the PSO algorithm is implemented to get a simplified set of rules. Figure 33 illustrates the new structure of the weighted Fuzzy Logic controller that will be used.

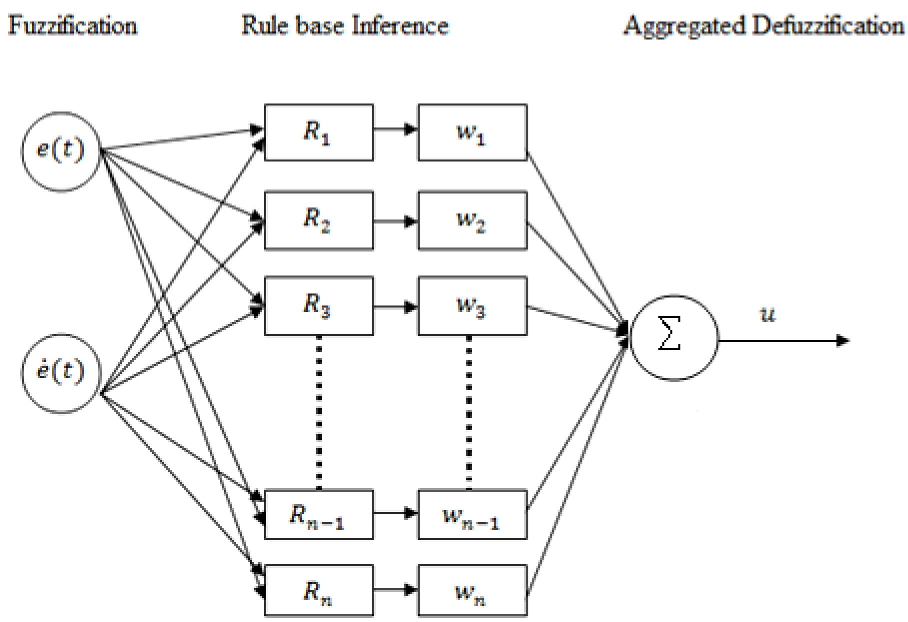

Figure 33. Weighted Fuzzy Logic controller

As illustrated in Figure 33, every rule has a weight that will be optimized using the PSO algorithm. In this methodology the PSO algorithm will have 49 parameters to be optimized, each parameter represents the weight of each rule.

After applying the second stage the number of rules reduced from (49) rules to (21) rules, as summarized in Table 6.

Table 6. PSO based FL PID controller rules

$\dot{e}$ \ *E*	*NS*	*ZE*	*PS*
NB	PM	PB	PS
NM	PM	PB	PM
NS	PM	PB	PM
ZE	PS	ZE	NS
PS	NM	NB	NM
PM	NM	NB	NM
PB	NS	NB	NM

Clearly, the linguistic variables of the error reduced from seven linguistic variables to three linguistic variables. The corresponding set of weights of these rules given by the PSO algorithm is enlisted in Table 7. Note that since this elevator application is more sensitive to the rate of change of the error (i.e. speed profile change); the $\dot{e}(t)$ linguistic variables are highlighted (fine), while the error in speed is taken as a coarse distribution (i.e. No need for large speed deviations). Also, note how the PSO generated weights about the ZE column of the error $e(t)$ were the largest. That is actually expected because fine tuning is performed by the controller in that region (high sensitivity).

Table 7. The set of weights given by PSO algorithm

weight	Value
$w1$	0.6238
$w2$	0.8518
$w3$	0.6069
$w4$	0.6193
$w5$	0.8967
$w6$	0.5895
$w7$	0.5948
$w8$	1.000
$w9$	0.5864
$w10$	0.2994
$w11$	1.000
$w12$	0.3128
$w13$	0.5912
$w14$	1.000
$w15$	0.5834
$w16$	0.6374
$w17$	0.8822
$w18$	0.5692
$w19$	0.6102
$w20$	0.8275
$w21$	0.5934

The generated PSO based Fuzzy Logic PID controller control surface is presented in Figure 34. The control surface demonstrates the controller's control signal as a function of controller's inputs, which provides a sense of the controller's trajectories for a two input based systems. In this case, this plot shows the control signals (reference voltage for the discrete PWM generator) as a function of the error and the rate of change of the error. Clearly the all trajectories are directed towards the origin, which gives an indication of the controller's stability.

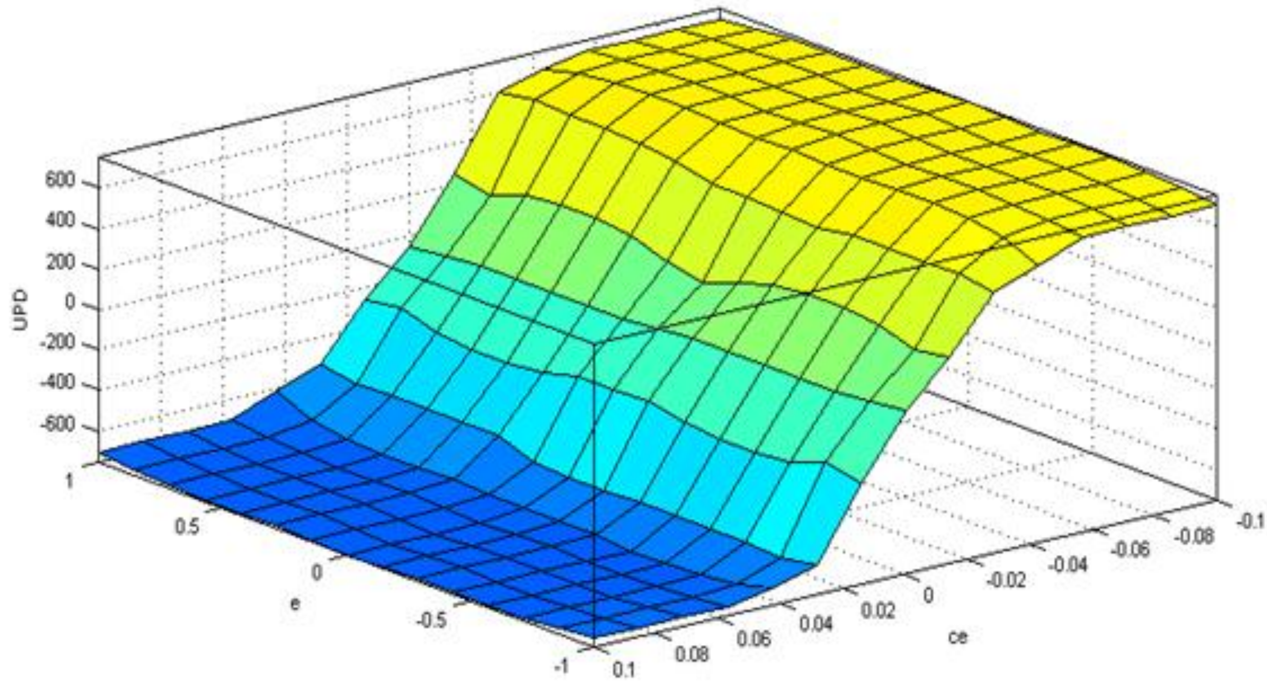

Figure 34. PSO based Fuzzy Logic PID controller control surface

Now that the rule base is constructed the rest of the controller's stages are designed accordingly.

Fuzzification: The membership function is a graphical representation of the magnitude of participation of each input. There are different membership functions associated with each input and output response. In this work, both triangular and trapezoidal membership functions for inputs and output variables were used due to their simplicity and effectiveness of implementation. The number of membership functions, to some extent, determines the quality of control that can be achieved using the fuzzy controller.

The elevator's system proposed Fuzzy Logic PID controller has two inputs, namely the error $e(t)$and the rate of change of the error $\dot{e}(t)$, and one output $u(t)$, namely the reference voltage for the discrete PWM generator, seven linguistic variables were tentatively selected to span the two inputs and the output ranges; Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (ZE), Positive Small (PS), Positive Medium (PM) and Positive Big (PB).

As a result of applying the two screening stages of the PSO algorithm, the seven linguistic variables for the first input of the Fuzzy PID controller ($e(t)$) were reduced to three linguistic variables while the number of linguistic variables for the second input ($\dot{e}(t)$) and the output of the Fuzzy PID controller are not changed.

The corresponding membership functions for the two inputs and the output of the Fuzzy Logic PID controller are shown in Figure 35, while the ranges of the linguistic variables set are summarized in Table 8.

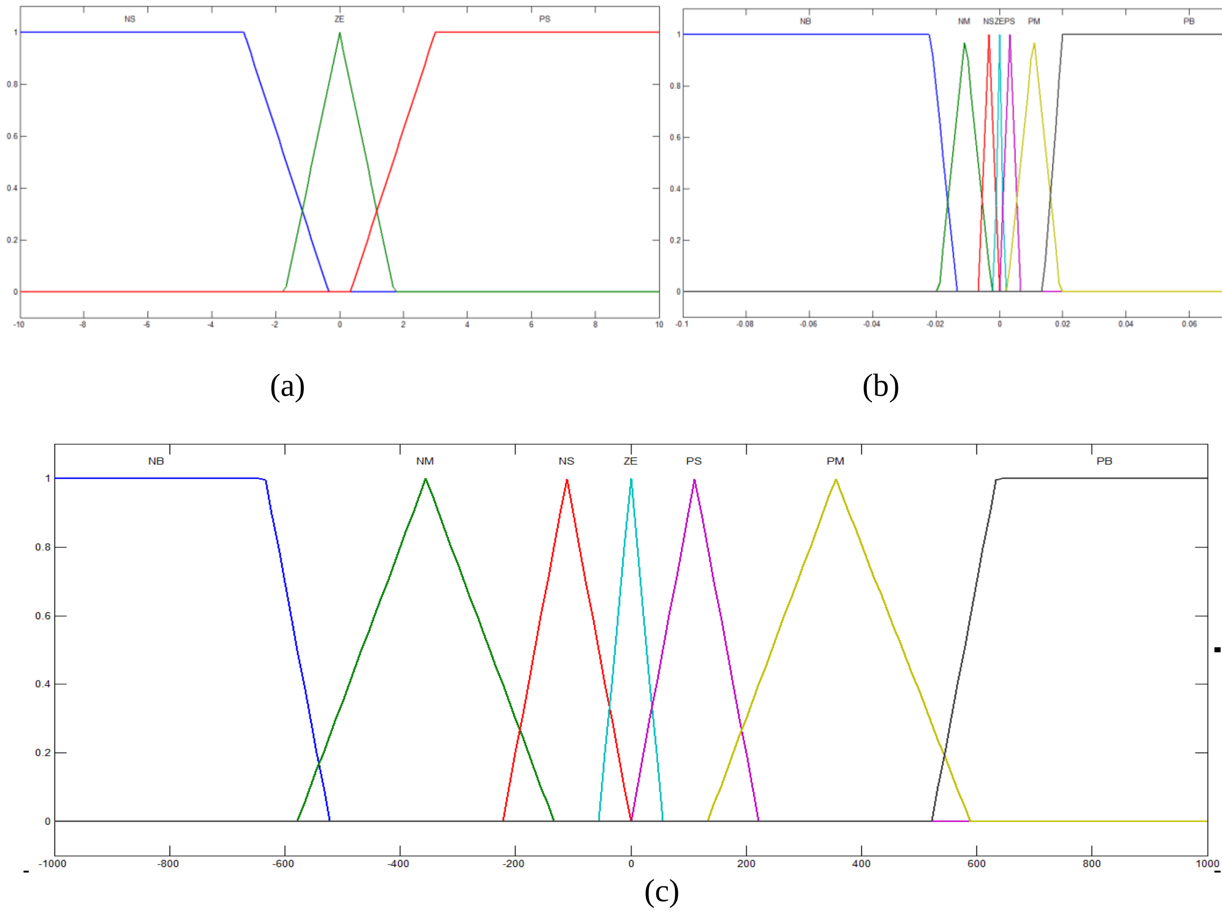

Figure 35. PSO generated membership functions (a) membership functions for the error (b) membership functions for the change of error (c) membership functions for the reference voltage for the discrete PWM generator.

Table 8. Membership functions liguistic variables ranges

Variable	Linguistic variable	Range
$\boldsymbol{e}$(t)	Negative Small (NS)	[-10 -10 -3 -0.34]
	Zero (ZE)	[-1.7 0 1.7]
	Positive Small (PS)	[0.34 3 10 10]
$\dot{\boldsymbol{e}}$(t)	Negative Big (NB)	[-0.1 -0.1 -0.02 -0.01]
	Negative Medium (NM)	[-0.02 -0.01 -0.003]
	Negative Small (NS)	[-0.007 -0.003 0]
	Zero (ZE)	[-0.002 0 0.002]
	Positive Small (PS)	[0 0.003 0.007]
	Positive Medium (PM)	[0.002 0.01 0.02]
	Positive Big (PB)	[0.01 0.02 0.1 0.1]
$\boldsymbol{u}$(t)	Negative Big (NB)	[-1000 -1000 -634 -522]
	Negative Medium (NM)	[-578 -356 -133]
	Negative Small (NS)	[-222 -111 0]
	Zero (ZE)	[-56 0 56]
	Positive Small (PS)	[0 111 222]
	Positive Medium (PM)	[133 355 588]
	Positive Big (PB)	[522 634 1000 1000]

Defuzzification: According to real world requirements, the linguistic variables have to be transformed to crisp outputs in order to be utilized by the actuator. A weighted centre of gravity method will be used in this work

$$u^* = \frac{\sum_{i=1}^{m} we_i \times \mu_{ik} \times c_i}{\sum_{i=1}^{m} we_i \times \mu_{ik}} \tag{42}$$

Where we_i is the weight of rule, μ_{ik} denotes the degree that instance matches the rule R_i , and $c_i \epsilon$ { C1, …. C_M} is the centroid of every fuzzy set.

This completes the design of the FL PID controller. The closed loop system (i.e. system and controller) response will be presented in the subsequent chapter.

CHAPTER 5

Simulating Results and Conclusions

In the previous chapters, a novel approach for designing a FL PID controller has been proposed. The design methodology was based on using an optimization algorithm to infer the FL rule base directly from the system without human experience intervention. In addition to that the FL controller rule base was also optimized in another pass to minimize the number of rules that is used. Now, this design methodology was verified against Lyapunov based rules extraction techniques, both techniques were setup and implemented to devise a FL PID controller for an elevator system. In this chapter the proposed methodology will be validated through numerical simulations, and results will be compared against the well established Lyapunov technique.

Now, to test the effectiveness of the Fuzzy PID controller in contrast to the Fuzzy Logic (FL) and the classical PID controller, Matlab computer simulations have been used for an elevator system test-bed. The parameters that were used for the 2:1 gearless elevator are fully summarized in Table 9. By noting that, the complete controllers' designs were introduced previously. Hence, this chapter will only provide the numerical results with some discussions and conclusions.

Table 9. Elevator system physical parameters

Armature resistance (R_a)	*0.49 Ω*
Armature inductance (L_a)	*4.3 mH*
motorarmature constant (K_m)	*0.49*
Coulomb friction value (Offset)(t_f)	*0.18*
Coefficient of viscous friction(Gain(K_f)	*4.6e-4*
Radius 1 (R1)	*0.2 m*
Radius2 (R2)	*0.3 m*
Radius3 (R3)	*0.2 m*
Moment of inertia (J1)	*0.08 $Kg.m^2$*
Moment of inertia (J2)	*0.15 $Kg.m^2$*
Moment of inertia (J3)	*0.08 $Kg.m^2$*
Mass of the elevator (M_{EL})	*100<MEL<500 Kg*
Mass of the weight (M_{CW})	*100 Kg*

5.1 Elevator's System Response

In this work, the proposed controllers' effectiveness will be demonstrated on performing two user calls. One call is make the elevator climb to the first floor while the other is to reach the tenth floor, or to reach four and fourty meters heights, respectively.

To validate the controller's ability to generate proper control laws that will track a designer's velocity profile, the results of traditionally used controllers have been compared. The traditional PID controller was fully tuned and optimized to control the elevator system, in order to carry out a fair comparison. On the other hand, a classical pure FL based controller was established and optimized to control the elevator's system as well. The used performance criteria will be based on the following merits:

1. Controller's ability to track the designers speed profile for each floor.

2. Controller’s ability to take the cart to the user’s designated destination accurately.
3. Controller’s smoothness in operating the elevator’s cart, which will be indicated using the cart’s acceleration and jerk.

Now, since concrete grounds have been established for comparison the standards that will be used to quantify these measures will be based on published **ASHREA** recommendations. According to Strakosch (1998), The recommended upper limits for **the acceleration and the jerk are 1.2 mps^2 and 2.4 mps^3 respectively**, while for the speed and position tracking, the following standard cost functions will be used

i. The Integral Absolute of the Error (IAE)

$$IAE = \int_0^{t_f} |e(t)| dt \quad (44)$$

Where $e(t)$ is the deviation from the designer’s speed profile for each floor.

ii. The Integral Square of the Error (ISE)

$$ISE = \int_0^{t_f} e^2(t) dt \quad (45)$$

iii. The Integral Time Absolute of the Error (ITAE)

$$ITAE = \int_0^{t_f} t|e(t)|\, dt \quad (47)$$

Noting that traditionally used measures of the system’s performance, such as: rise time, overshoot, steady state error …etc. are all implemented within these measures.

Classical PID controller: A well establish standard Proportional, Integral and Derivative (PID) controller is used to validate the work of the FL controllers. The PID controller has demonstrated a successful long running journey in the industrial applications for the past sixty years. It's popularity is mainly due to its fixed structure and easiness of utilization yet effectiveness of performance. It's design is simply done by tuning it's parameters K_P, K_I and K_D. In this work, the PSO based optimization techniques have been used to fully tune the PID controller's parameters to generate superb performance. The PID controller's performance is full captured through the Figures 36 and 37.

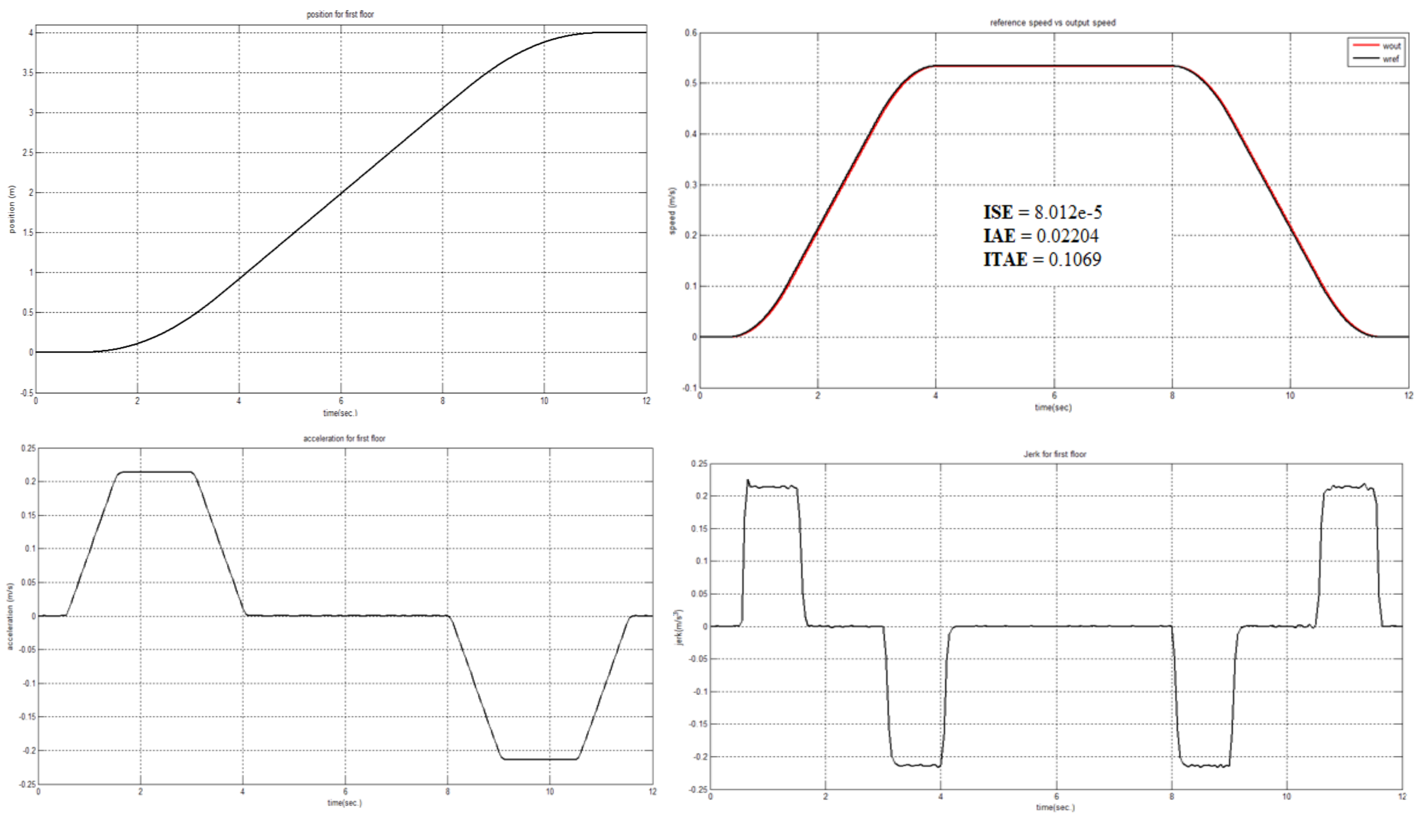

Figure 36. Classical PID system responses, a) Position at 4m height, b) Reference speed vs. actual speed for 4m height, c) Acceleration for 4m height, d) Jerk at 4m height

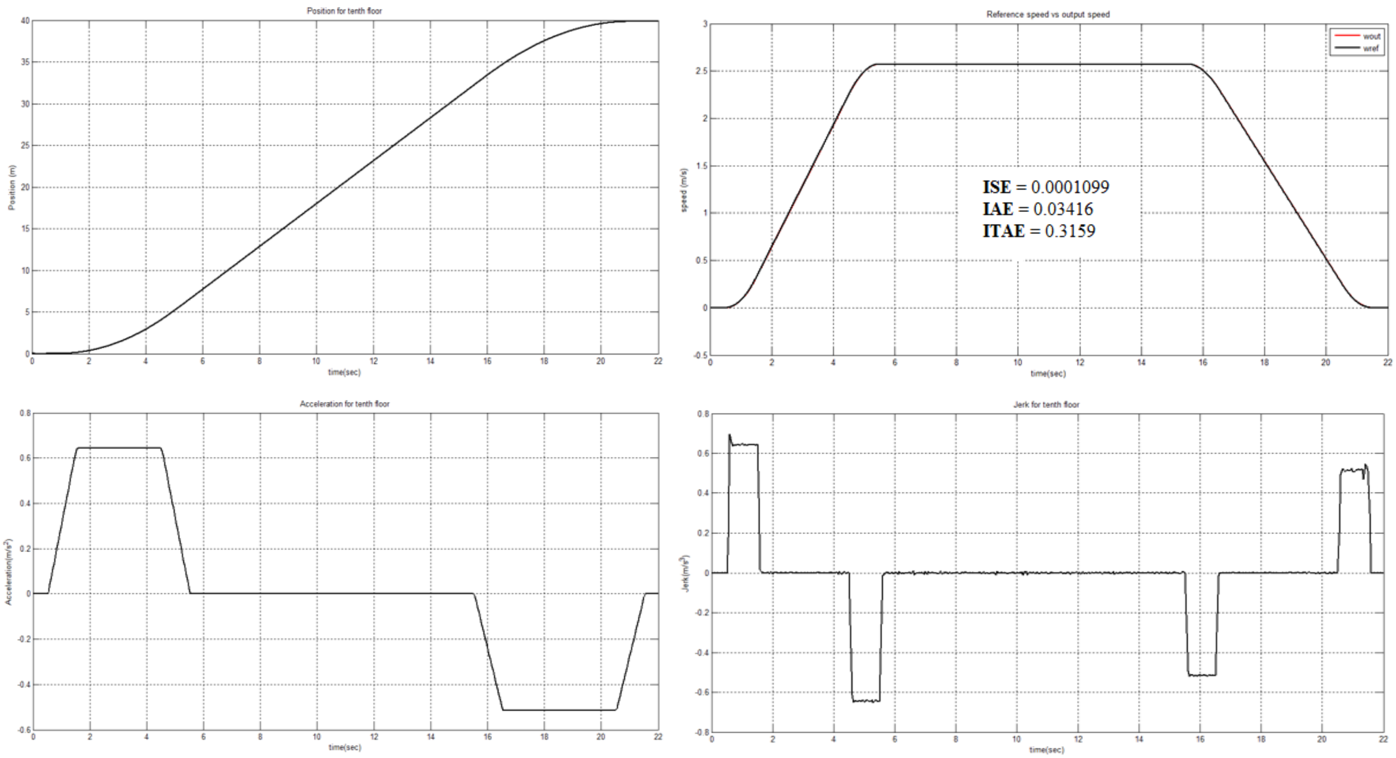

Figure 37. Classical PID system responses, a) Position at 40 m height b) Reference speed vs. actual speed for 40 m height c) Acceleration for 40 m height d) Jerk at 40 m height

Classical FL controller: The previously designed FL controller with the 49 rules was tested to operate the elevator test-bed for the first and tenth floor calls. Figures 38 and 39, demonstrate the FL controller responses, where tracking the speed and position are done nicely, with minimal amount of acceleration and jerk for the two executed scenarios. Please note that the rules that are used in this Classical FL controller design are based on trial and error and gained human experience.

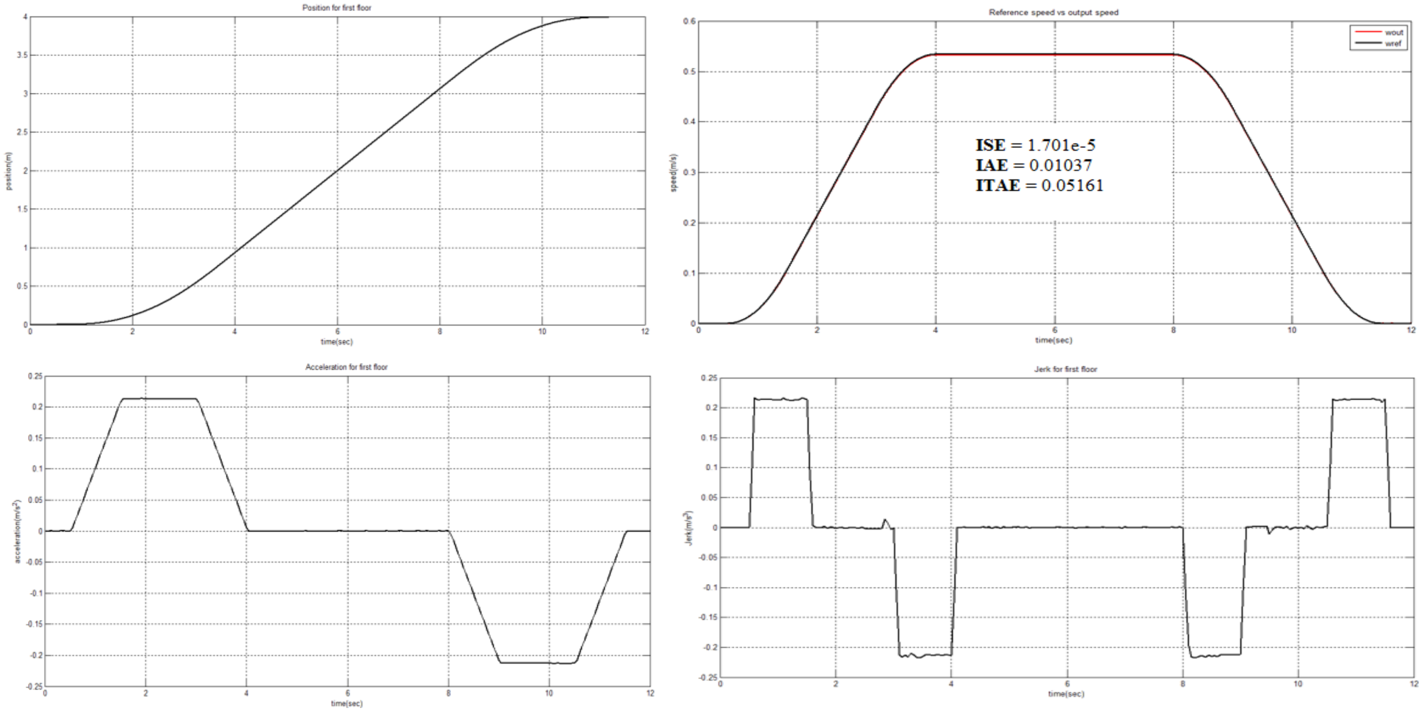

Figure 38. Classical FL controller system responses, a) Position at 4m height b) Reference speed vs. actual speed for 4m height c) Acceleration for 4m height d) Jerk at 4m height

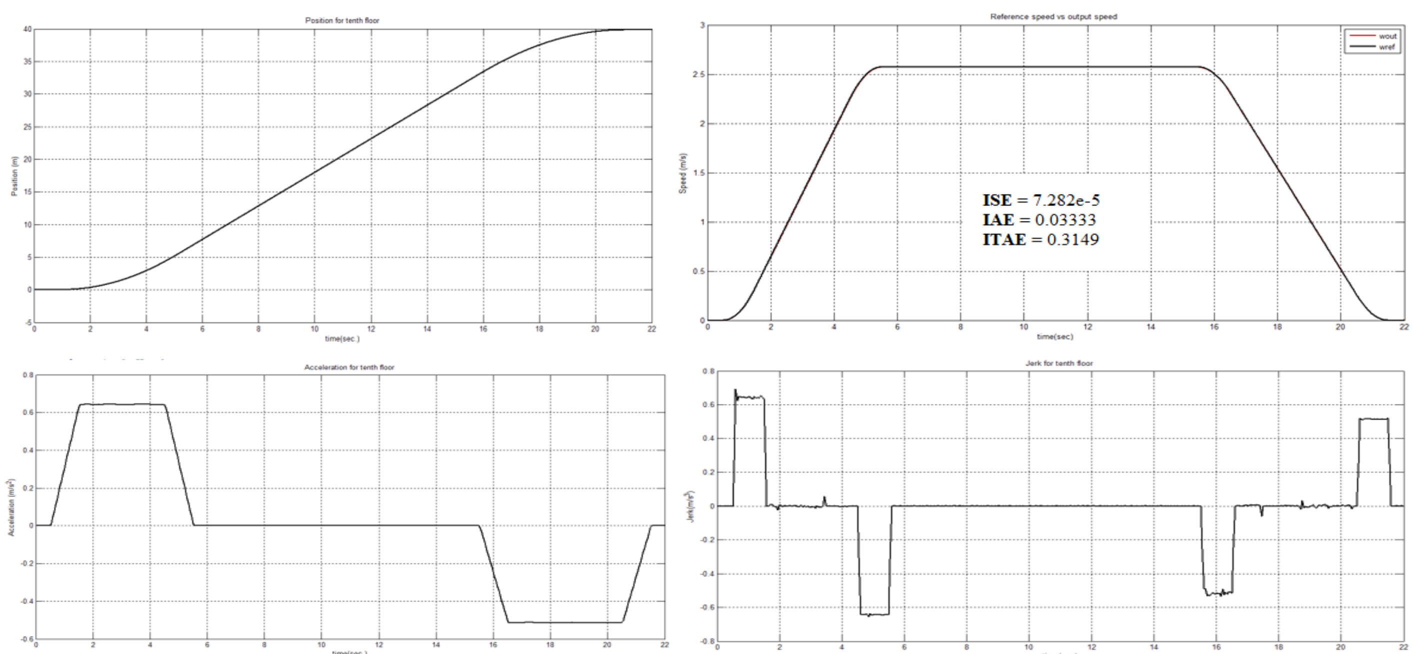

Figure 39. Classical FL controller system responses, a) Position at 40 m height b) Reference speed vs. actual speed for 40 m height c) Acceleration for 40 m height d) Jerk at 40 m height

PSO Optimized FL PID controller: The previously designed controller with 21 rules was tested for first and tenth floor calls for the elevator. Figures 40 and 41 depicts the position, velocity, acceleration and jerk trajectories for the first and tenth floors user calls, respectively.

The controller's ability to track the designer's speed profile is evident (i.e. total match) for the two scenarios. Also, the acceleration and jerk values are within human tolerated values. Finally, the elevator's ability to track the user's call is depicted in reaching the desired positions accurately.

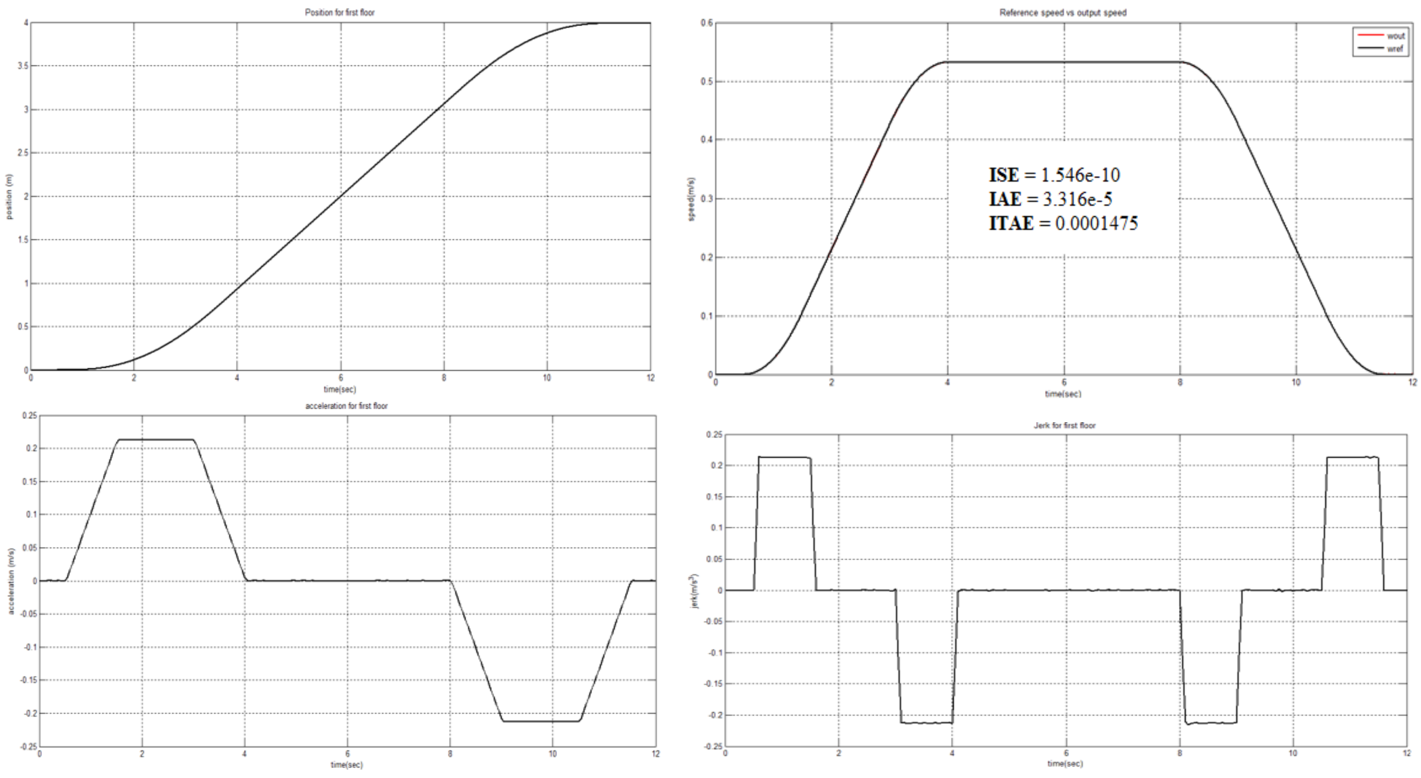

Figure 40. PSO FL PID system responses a) Position at 4 m height b) Reference speed vs. actual speed for 4m height c) Acceleration for 4 m height d) Jerk at 4 m height

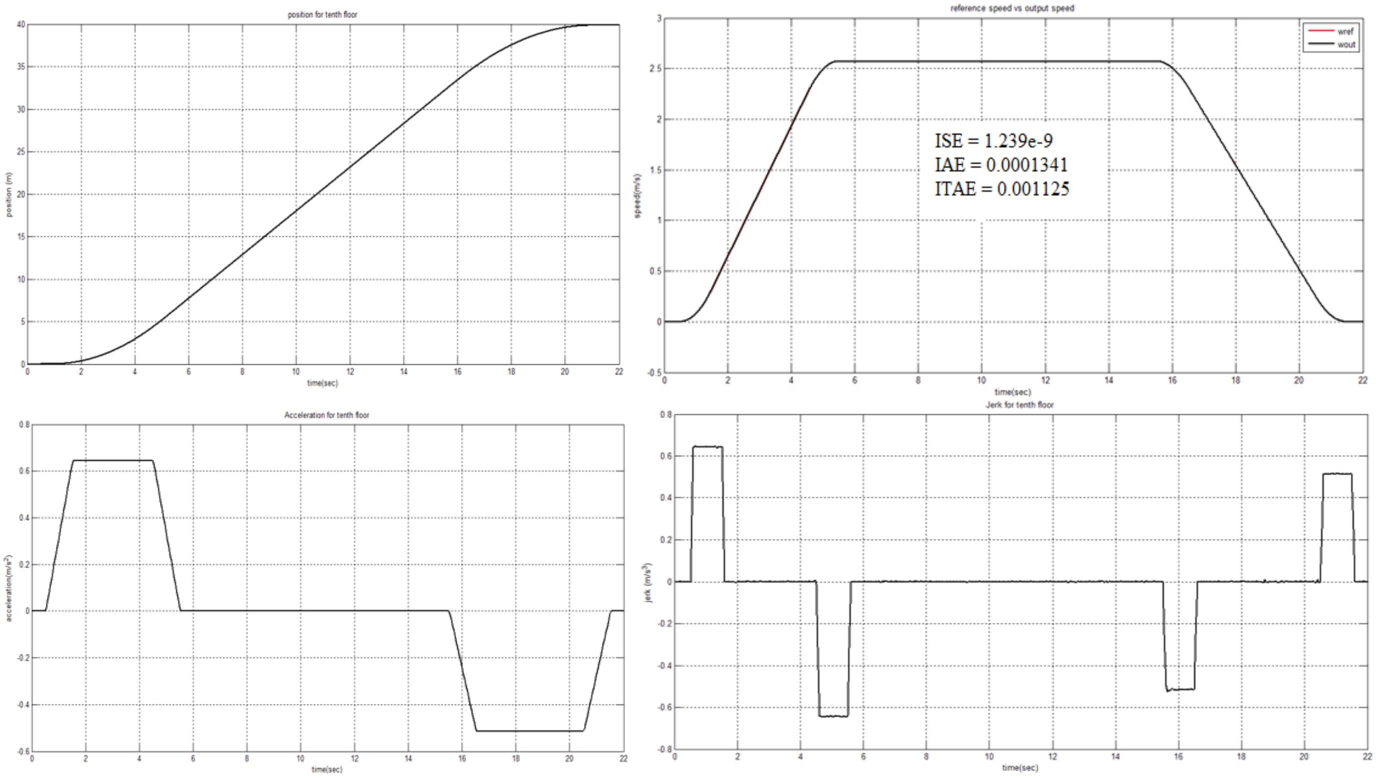

Figure 41. PSO FL PID system response, a) Position at 40 m height b) Reference speed vs. actual speed for 40 m height c) Acceleration for 40 m height d) Jerk at 40 m height

Now that the results of the three controllers are presented, they should be compared collectively. Figures 42-47 and Tables 10 and 11, summarizes the comparison of the speed profile tracking for the first and tenth floors user requests. The FL PID controller demonstrates superiority in both cases. Also, the ITAE comparisons show that the FL PID controller actions to be somewhat faster (i.e. because it is penalized over time). Please note that the Lyapunov FL controller is enlisted in the table for the sake of completeness in comparison, however, it’s figures is not included because it was very similar to the FL controllers.

Finally, one major advantage for the FL controllers is the fact that one may incorporate artificial intelligence in the controllers rule base and that is something cannot be done directly in the PID controller and this mainly due to the PID's controller fixed structure.

Table 10. Cost functions for the controllers at 4m height

Controller	ISE	IAE	ITAE
PID	8.012e-005	0.02204	0.1069
FLC	1.701e-005	0.01037	0.05161
Lyapunov-FL-PID	1.088e-008	0.0002618	0.001154
PSO-FL-PID	1.546e-10	3.316e-5	0.0001475

Table 11. Cost functions for the controllers at 40m height

Controller	ISE	IAE	ITAE
PID	0.0001099	0.03416	0.3159
FLC	7.282e-005	0.03333	0.3149
Lyapunov-FL-PID	4.76e-007	0.002643	0.02452
PSO-FL-PID	1.239e-9	0.0001341	0.001125

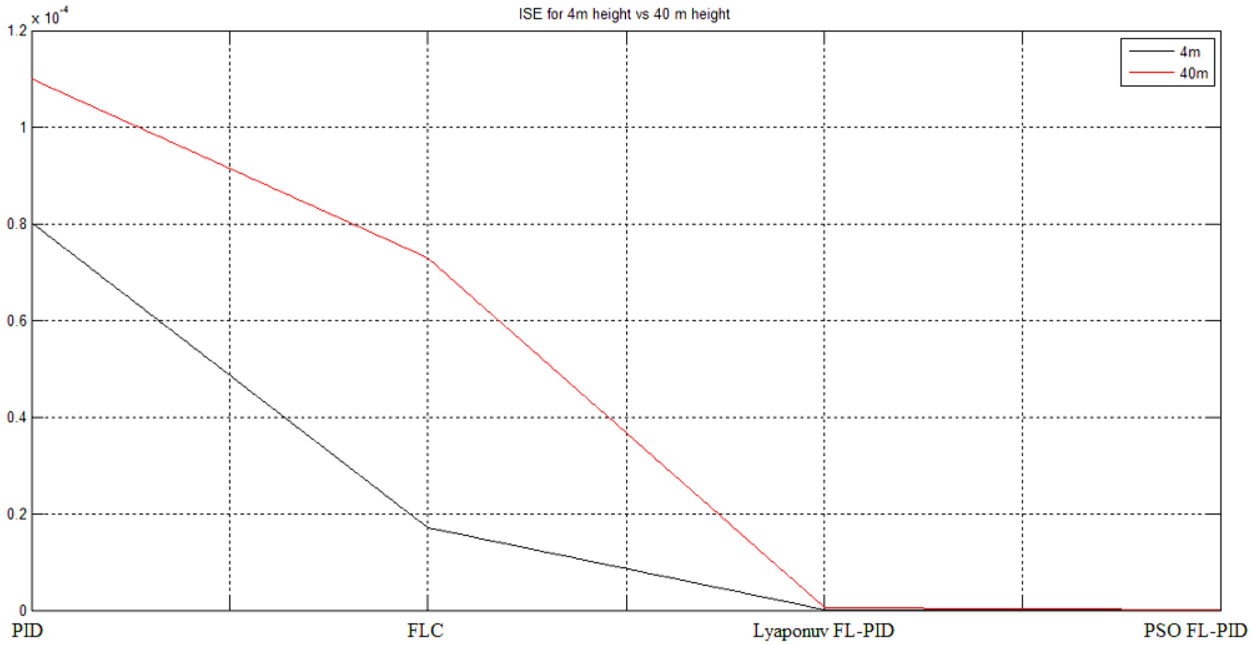

Figure 42. Integral square of error for 4m and 40m height

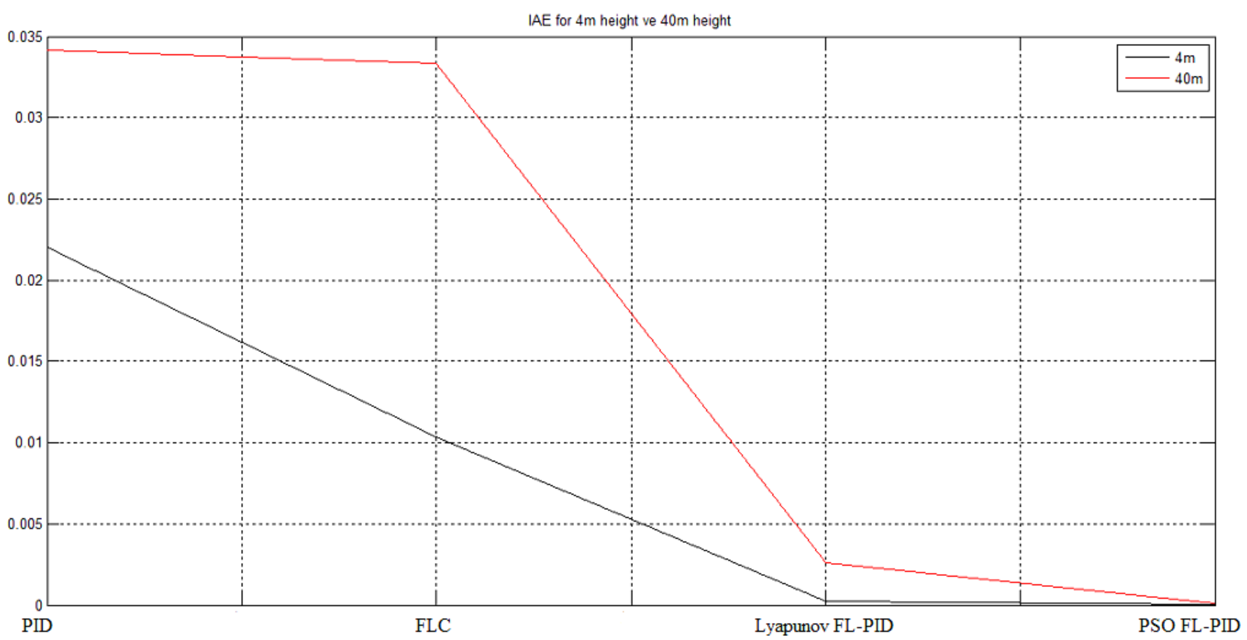

Figure 43. Integral absolute of error for 4m and 40m height

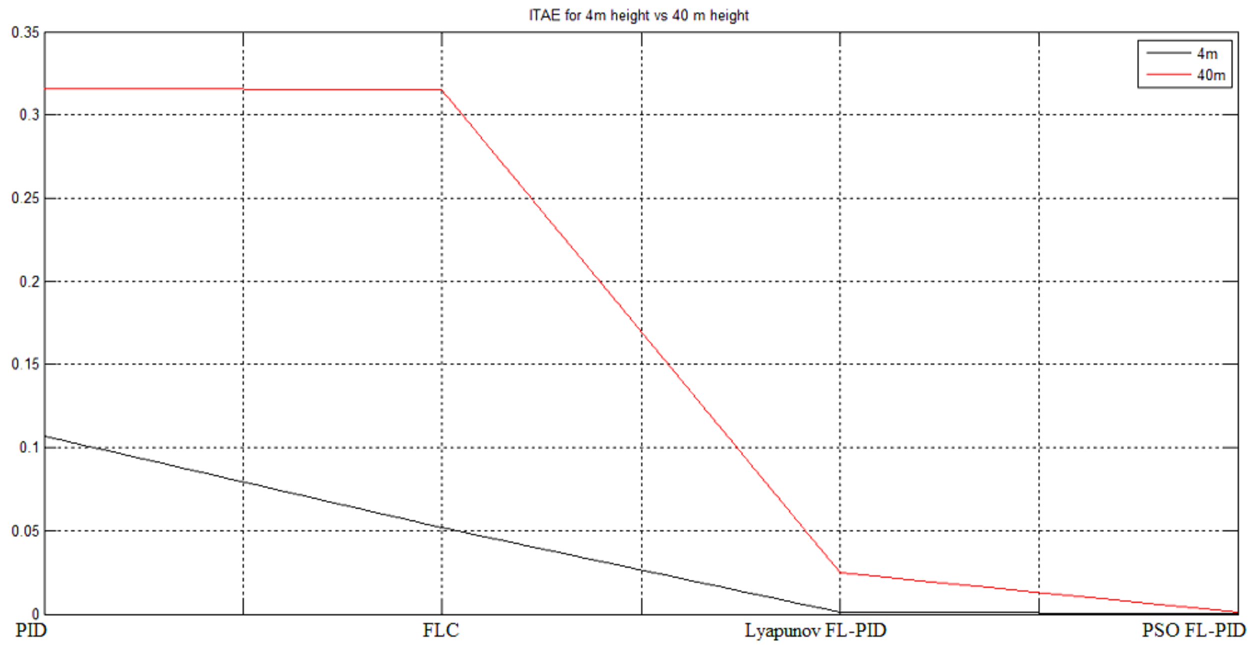

Figure 44. Integral time absolute of error for 4m and 40m height

5.2 Conclusions

A novel methodology for Fuzzy Logic controller design is proposed. The method shows how to generate the output outcomes in a Fuzzy Logic controller rule base without human experience intervention, as a first optimal screening. Also, the method optimizes the number of linguistic variables that are used in the fuzzification and output stages in order to simplify the fuzzy rules generated from the first optimal screening, as a second optimal screening. The method utilizes a binary search optimization algorithm for the optimal screening that is based on Particle Swarm Optimization.

A FL- PID controller was devised and successfully tested, verified and validated over a gearless traction elevator system via simulations. The proposed technique has shown promising results in auto-generation of the rules for the FL-PID controller. The results then compared with optimized classical PID controller and classical FL controller. The optimized FL-PID proposed in this work showed superiority in controlling an elevator system based on control standards.

Also, the proposed method generated rules were further compared with Lyapunov based Fuzzy Logic controller design methodology and the proposed method showed superiority based on control standards.

Finally, different scenarios of the elevator application problem all of it proved the FL-PID controller ability to track position and speed profile yet maintaining minimal amount of acceleration and jerk.

Furthermore practical testing beyond simulation of this design methodology is recommended. Hence, a good future work may be to implement such a designed controller to control an elevator prototype in a laboratory environment. Also, other types of applications may take the advantage of this design methodology because it is not application dependent. On the other hand, stability analysis of the proposed controller based on Lyapunov direct method is a good future work too.

References

Abdalla, M. (2009), Fuzzy Logic. **Lecture Notes in Advanced Fuzzy Logic and Neural Nets, University of Jordan**.

Allaoua, B., Abderrahmani, A., Gasbaoui, B. and Nasri, A. (2008), The Efficiency of Particle Swarm Optimization Applied on Fuzzy Logic DC Motor Speed Control. **Serbian Journal of Electrical Engineering**, (5), 247-262.

Bandyopadhyay, R., Chakraborty, U.K., Patranabis, D. (2001), Autotuning a PID Controller: A Fuzzy-Genetic Approach. **Journal of Systems Architecture**, (47), 663-673.

Bingul, Z., Karahan, O. (2010), A Fuzzy Logic Controller Tuned with PSO for a 2 DOF Robot Trajectory Control. **Expert System Application**, 1017-1031.

Boutler, B. T. (2000), Elevator modeling and DC drive speed controller design. **Applied Industrial Control Solutions ApIC S LLC.**

Buhar, C. and Leephakpreeda, T. (1994), Stability of Fuzzy Control Systems, **IEEE International Conference on Neural Networks**, (4), 2605-2610.

Carvajal, J. Chen, G., Ogmen, H. (2000), Fuzzy PID Controller: Design, Performance Evaluation, and Stability Analysis, **Information Science**, (123), 249-270.

Chen, C., (2001), Design of Stable Fuzzy Control Systems Using Lyapunov's Method in Fuzzy Hypercubes. **Fuzzy Sets and Systems**, (139), 95-110.

Chen, C. W., Chiang, W. L., Tsai C. H., et al. (2006), Fuzzy Lyapunov Method for Stability Conditions of Non-linear Systems. **International Journal on Artificial Intelligence Tools**, (15), 163–171.

Cui, X., Hardin, T., Ragade, R. K. and Elmaghraby, A.S. (2004), A Swarm-based Fuzzy Logic Control Mobile Sensor Network for Hazardous Contaminants Localization. **IEEE International Conference on Mobile Ad-hoc and Sensor Systems**.

Eberhart, R. C. and Y. Shi. (2001),Tracking and optimizing dynamic systems with particle swarms, 94-97.

Elwer, A. S., Wahsh, S. A. (2009), Improved Performance of Permanent Magnet Synchronous Motor by Using Particle Swarm Optimization Techniques. **Journal of Power Electronics**, (9), 207-214.

Erenoglu, I., Eksin, I., Yesil E., and Guzelkaya, M. (2006), An Intelligent Hybrid Fuzzy PID Controller. **Proceedings of 20th European Conference on Modeling and Simulation**.

Ermanu, Hakim, A., Adi S., Mauridhi H.P. (2007), PSS Design Based on PD and PI Fuzzy Controller by Particle Swarm Optimization. **Proceeding ICEEI**, 17-19.

Feng, G. (2003), Controller Synthesis of Fuzzy Dynamic Systems Based on Piecewise Lyapunov Functions. **IEEE Transactions on Fuzzy Systems**, (11), 605-612.

Gao, F., Tong, H.Q. (2006), A novel Optimal PID Tuning and on-line Tuning Based on Particle Swarm Optimization. **Proceeding of the International Conference on Sensing, Computing and Automation**.

Georg, K. I., Hu, B. G. and Gosine R.G. (2001), Two-Level Tuning of Fuzzy PID Controllers. **IEEE Trans. on System, Man, and Cybernetics**, (31), 263-269.

Guzelkaya, M., Eksin, I. and Yeil E. (2003), Self-tuning of PID-type Fuzzy Logic controller coefficients via relative rate observer. **Engineering Applications of Artificial Intelligence**, (16), 227-236.

Hu, X. and Eberhart, R. C. (2001), Tracking Dynamic Systems with PSO: Where's the Cheese?. **Proceeding of the Workshop on PSO**.

Huayong, Y., Jian, Y., Bing, X. (2004), Computational Simulation and Experimental Research on Speed Control of VVVF Hydraulic Elevator. **Control Engineering Practice**, 563-568.

Jantzen, J. (1998), Design of Fuzzy Controllers. **Technical University of Denmark: Department of Automation**, Technical Report no. 98-E-864.

Kang, J. K. and Sul, S. K. (2000), Vertical-Vibration Control of Elevator Using Estimated Car Acceleration Feedback Compensation. **IEEE Transactions on Industrial Electronics**, (47), 91-99.

Kao, C. C., Chuang, C. W., Fung, R.F. (2006), The Self-tuning PID Control in a Slider-crank Mechanism System by Applying Particle Swarm Optimization Approach. **Mechatronics**, (16), 513-522.

Kazemian, H. B. (2005), Developments of Fuzzy PID controllers. Expert Systems: **The Int. Journal of Knowledge Engineering and Neural Networks**. (22), 254-264.

Kennedy, J., R. Eberhart (1995), Particle Swarm Optimization. **Proceedings of IEEE International Conference on Evolutionary Computation**, 1942-1948.

Kim, C. S., Hong, K. S., Kim, M. K. (2004), Nonlinear Robust Control of a Hydraulic Elevator: Experiment-based Modeling and Two-stage Lyapunov Redesign. **Control Engineering Practice**, (13), 789-803.

Korba, P., Babuska, R., Verbruggen, H. and Frank, P. (2003), Fuzzy Gain Scheduling: Controller and Observer Design Based on Lyapunov Method and Convex Optimization. **IEEE Transactions on Fuzzy Systems**, (1), 285-298.

Lee.C. (1990), Fuzzy Logic in Control Systems: Fuzzy Logic Controller, part 1. **IEEE Transaction Systems**, (20), 404-418.

Li, J. H., Li, T. S., Chen, C. Y. (2007), Design of Lyapunov Function Based Fuzzy Logic Controller for a Class of Discrete Time Systems. **International Journals for Fuzzy Systems**, (9), 1-7.

Li, K., Mannan M.A., Xu, M., Xiao, Z. (2001), Electro-hydraulic Proportional Control of Twin-cylinder Hydraulic Elevators. **Control Engineering Practice**, (9), 367–373.

Mamdani, E.H., Assilian, N.S. (1974), A Case Study on the Application of Fuzzy Set Theory to Automatic Control. **Proc. IFAC Stochastic Control Symp, Budapest**.

Margaliot, M., Langholz, G. (1999), Fuzzy Lyapunov-based Approach to the Design of Fuzzy Controllers. **Fuzzy Sets and systems**, 49-59.

Massad, E., Ortega, N. R. S., Barros, L. C. and Struchiner, C. J., **Fuzzy Logic in Action: Applications in Epidemiology and Beyond**. Studies in Fuzziness and Soft Computing, Springer, (232).

O'Dwyer, A. (2001), PI and PID Controller Tuning Rules for Time Delay Processes: Summary, Technical Report AOD-00-01, version 2. **Dublin Institute of Technology, Ireland.**

Ramsey, Sleeper (2007), **Architectural Graphic Standards**, eleventh edition. American Institute of Architects.

Rhee, B.J, Won, S. (2006), A new Fuzzy Lyapunov Function Approach for a Takagi-Sugeno Fuzzy Control System Design. **Fuzzy Sets System**, 1211-1228.

Sha, D., Bajic, V. B., Yang, H. (2002), New Model and Sliding Mode Control of Hydraulic Elevator Velocity Tracking System. **Simul Pract Theory**, 65-85.

Sharma, K.D., Chatterjee, A., Rakshit, A. (2009), Hybrid approach for design of stable adaptive controllers employing Lyapunov theory and particle swarm optimization. **IEEE Transactions on Fuzzy Systems**, (17), 329-342.

Strakosch, G. R. (1998), **The Vertical Transportation Handbook**, 3rd edition, John Wiley and Sons, New York.

Sugeno, M. Takagi, T. (1983), A new Approach to Design of Fuzzy Controllers, **Plenum Press, New York, Wang, p. p. edition**, 325-334.

Tanaka, K., Hori, T. and Wang, H. O. (2003), A multiple Lyapunov Function Approach to Stabilization of Fuzzy Control Systems. **IEEE Transactions on Fuzzy Systems**, (11), 582–589.

Tanaka, K., Ohtake, H. and Wang, H. O. (2007), A descriptor System Approach to Fuzzy Control System Design via Fuzzy Lyapunov Functions. **IEEE Transactions on Fuzzy Systems**, (15), 333–341.

Tang, K. S., Man, K. F., Chen, G. and Kwong, S. (2001), An Optimal Fuzzy PID Controller. **IEEE Transactions on Industrial Electronics**, (48), 757 – 765.

Visioli, A. (1999), Fuzzy Logic Based Set-Point Weight Tuning of PID Controllers. **IEEE Transactions on Systems Man and Cybernetics Part a-Systems and Humans**, (29), 587-592.

Visioli, A. (2001), Tuning of PID Controllers with Fuzzy Logic. **IEE Proceedings – Control Theory and Application**, (14), 1-8.

Wahsh, S., Elwer, A., Khalil, M. and Noureldin, A. (2005), Tuning PI-Controller Using Particle Swarm Optimization for Control of Permanent Magnet Synchronous Motor. **5th st. Petersburg Workshop in Simulation, RUSSIA**, 239-244.

Woo, Z.W, Chung H.Y., Lin J. (2000), A PID Type Fuzzy Controller with Self-tuning Scaling Factors. **Fuzzy Sets and Systems**, (115), 321 – 326.

Wu, H. N. and Zhang, H. Y. (2007), Reliable H-infinity Fuzzy Control for a Class of Discrete-time Non-linear Systems Using Multiple Fuzzy Lyapunov Functions. **IEEE Transactions on Circuits and Systems I**, (54), 357–361.

Yeh, K. Chen, C. Y. and Chen C. W. (2008), Robustness Design of Time-delay Fuzzy Systems Using Fuzzy Lyapunov Method. **Applied Mathematics and Computation**, (205), 568–577.

Zadeh, L.A. (1956), Fuzzy Sets. **Information and Control**, (8), 338-353.

Zadeh, L.A. (1972), A Rationale for Fuzzy Control. **J. Dynamic System Meas. Control**, (94), 3-4.

Zadeh, L.A. (1973), Outline of a New Approach to the Analysis of Control Systems and Decision Processes. **IEEE transaction SMC-3**, 28-44.

Zadeh, L.A. (1975), The Concept of a Linguistic Variable and its Application to Approximate Reasoning-I. **Information Sciences**, (8), 199-249.

Zadeh, L.A. (1975), The Concept of a Linguistic Variable and its Application to Approximate Reasoning-II. **Information Sciences**, (8), 301-357.

Zadeh, L.A. (1975), The Concept of a Linguistic Variable and its Application to Approximate Reasoning-III. **Information Sciences**, (9), 43-80.

Zadeh, L. (1996), Fuzzy Logic = computing with words. **IEEE Transactions on Fuzzy Systems**, (4), 103-111.

Zhang, T.J, Feng G. and Lu, J.H. (2007), Fuzzy Constrained Min-max Predictive Control Based on Piecewise Lyapunov Functions. **IEEE Transactions on Fuzzy Systems**, (15), 686-698.

Zhou, C. (2002), Fuzzy-arithmetic-based Lyapunov Synthesis in the Design of Stable Fuzzy Controllers: a Computing-with-words Approach. **International J. Applied Mathematics Computer Science**, (12), 411-421.

Zhou, L., Dong, Z., Wang, S. and Qi, Z. (2008), Design and Analysis of a Hybrid Backup Power System for a High-Rise and High-Speed Elevator. **Proceedings of the 2008 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications**, 12-15.

المتحكم النسبي التكاملي الاشتقاقي الضبابي الأمثل

إعداد
تغريد عاطف الجراح

المشرف
الدكتور موسى حسن عبد الله

ملخـــــــــــــــص

واحدة من القضايا الصعبة في تصميم النظام الضبابي هو توليد القواعد، الذي هو أساسا استراتيجية التحكم للنظام. تقليديا، بناء قواعد المتحكم الضبابي كان يعتمد بشكل أساسي على خبرة مشغلي التحكم أو على تصرفاتهم. لسوء الحظ، الحصول على القواعد من الخبراء ليس مهمة سهلة، علاوة على ذلك من الصعب جدا لمهندس المعارف استخراج القواعد من بيانات ثابتة، من ناحية أخرى، تحديد مجموعة القواعد الأهم للمتحكم الضبابي من قواعد معطاه قضية مهمة في تصميم المتحكم الضبابي.

في هذا العمل، عرضت منهجية تصميم المتحكم الضبابي بشكل كامل. تستخدم الطريقة خوارزمية بحث أمثلية سرب الجسيمات ثنائية البحث لتوليد نتائج المخرجات في قاعدة المتحكم الضبابي بدون تدخل الخبرة البشرية، كفرز أولي. بالأضافة إلى ذلك، الطريقة المطروحة تستخدم منهجية بحث أمثلية سرب الجسيمات لتبسيط قواعد المتحكم الضبابي، كفرز ثاني.

التقنية المعروضة قورنت مع تصميم المتحكم الضبابي المبني بالاعتماد على ليابونوف في توليد القواعد. وقد أظهرت الطريقة المعروضة تفوقا عليها في نتائج مخرجات النظام مع قائمة قواعد أقصر.

أخيرا، فاعلية المتحكم المطروح و أداؤه تم فحصها، التحقق منها و إقرارها باستخدام المصعد الكهربائي كتطبيق للمتحكم ثم تم مقارنة نتائج المتحكم المطروح مع المتحكم الضبابي الكلاسيكي و المتحكم النسبي التكاملي الاشتقاقي الكلاسيكي، و أظهر المتحكم المطروح تفوقه بالتحكم في المصعد بالاعتماد على معايير تحكم مختلفة.