DOKUZ EYLUL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DC TO DC STEP-UP CHOPPER WITH FUZZY LOGIC CONTROLLER

by

Evin TOPTAŞ

November, 2005

İZMİR

DC TO DC STEP-UP CHOPPER WITH FUZZY LOGIC CONTROLLER

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylul University
In Partial Fulfillment of the Requirements for the Degree of Master of Science
in
Electrical-Electronically Engineering, Applied Electronically Program

by
Evin TOPTAŞ

November, 2005
İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled DC TO DC STEP-UP CHOPPER WITH FUZZY LOGIC CONTROLLER completed by EVİN TOPTAŞ under supervision of PROF. DR. EYÜP AKPINAR and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Eyüp AKPINAR

Supervisor

Yrd. Doç. Dr. Hacer ÖZTURA

(Jury Member)

Prof. Dr. Erol UYAR

(Jury Member)

Prof.Dr. Cahit HELVACI
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENTS

I would like to thank my supervisor Prof. Dr. Eyüp AKPINAR for his valuable guidance and support during the course of this thesis. Asst. Tolga SÜRGEVİL for his helpful advice. Finally, I thank my parents and my friend Deniz BUDAKÇI for their understanding and never ending support throughout my life.

Evin TOPTAŞ

DC TO DC STEP-UP CHOPPER WITH FUZZY LOGIC CONTROLLER

ABSTRACT

In this thesis, a computer simulation program and an implemented circuit are used for a step-up chopper with fuzzy logic and the results are compared.

The, structures of dc-dc step-up chopper are explained. And also the values of components of the step-up chopper are calculated. The advantages, disadvantages and main principle of the fuzzy logic application in this drive circuit are explained. Also this method is compared with other methods.

In simulation program, the circuit of step-up chopper with fuzzy logic are establish in Matlab Simulink and the results of this circuit are examined.

The step-up chopper is controlled with microcontroller. For this reason, PIC 16F877 is used. The software of PIC 16F877 is written by Assembly machine language. Also, in this thesis the features and structures of PIC 16F877 are explained briefly.

In implemented circuit, a writer circuit of PIC 16F877 is established in order to load Assembly machine program into the PIC 16F877 test circuit.

Closed loop control is applied to the computer simulation program. Fuzzy logic control method is used for speed control. PWM is only applied to MOSFETs , which is used for switching.

Keywords: Step-up Chopper, Fuzzy Logic Control, PIC 16F877

GERİLİM ARTTIRICI DC KONVERTÖRÜN BULANIK MANTIK İLE KONTROLÜ

ÖZ

Bu tezde gerilim arttırıcı dc konvertör bulanık mantık ile kontrol edilmiştir. Bu amaçla bilgisayar ortamında simülasyon ve laboratuar ortamında deneysel çalışmalar yapılmıştır ve bunların sonuçları karşılaştırılmıştır.

Gerilim arttırıcı dc konvertörün özellikleri, yapısı anlatılmıştır. Ayrıca konvertörün elemanlarının hesaplamaları da anlatılmıştır. Bulanık mantığın temel ilkelerinin bu sürücü devresine uygulanması, avantajları ve dezavantajları anlatılmıştır. Ayrıca diğer kontrol yöntemleri ile kıyaslaması anlatılmıştır.

Matlab Simulink, simülasyon programında devre kurulmuş ve sonuçlar ilkönce bilgisayar ortamında incelenmiştir.

Gerilim arttırıcı dc konvertörün kontrolü mikro kontrol ile yapılmaktadır. Bunun için PIC 16F877 kullanılmıştır. PIC 16F877'nin yazılımı Assembly makine dili ile yapılmıştır. Tezde PIC 16F877'nin yapısı ile ilgili de bilgiler verilmiştir.

Assembly programlama dilini PIC 16F877 deneme kartına aktarmak için laboratuar ortamında bir yazıcı devresi yapılmıştır.

Simülasyon programında kapalı çevrim kontrol uygulanmıştır. Hız kontrolü için bulanık mantık kontrol yöntemi uygulanmıştır. PWM, anahtarlama elemanı olarak kullanılan MOSFET' e uygulanmıştır.

Anahtar Sözcükler: Gerilim Arttırıcı DC Konvertör, Bulanık Mantık, PIC 16F877

CONTENTS

Page

CHAPTER ONE
INTRODUCTION

1. Introduction

In the last years, the application of fuzzy logic controllers has been extended gradually in process control area and successful results have been taken. They have been applied to many fields Kosko, B., (1992) as economy, administration, medical science and process control systems since introduced by A. Lütfi Zadeh in 1965 Zadeh, L.A., (1965). Fuzzy logic controller is formed in the light of the knowledge and experiences of an expert. Thus, a flexible control method can be developed through the commonly used daily language words as; increase a lot, increase a little, a lot, a little that are known as linguistic determinatives. Fuzzy logic controller was based on such verbal statements and the logical relationships between them Hellendorn, H., & Thomas, C., (1993). The Fuzzy logic is much closer in spirit to human thinking and natural language than the traditional logic systems. The Fuzzy logic controller does the control process using linguistic expressions based on expert knowledge Zadeh, L.A., (1965). It is easy to develop a fuzzy logic controller, as mathematical expressions are not required. Therefore, the fuzzy logic controller can be applied to the complex systems that are non linear and cannot be modeled Lee, C.C., (1990), Mamdami, E.H., (1977). The fuzzy logic controller has three main qualities as transforming numerical values into linguistic expressions, linguistic outcome for resolution system and transforming linguistic conclusions into numerical conclusions for numerical control information Akcayol, M.A., & Çetin, A., & Elmas, C., (2002).

In this work, the fuzzy logic control was used to generate voltages above the supply voltage. The dynamics of DC-DC converters is non-linear, and practical converter operation deviates from theoretical prediction because of problems associated with parasitic resistances, stray capacitances, and leakage inductances of the components. All this make the design of an optimal compensation circuit for closed loop operation of the converter difficult So, W.C., & Tse, C.K., & Lee, Y.S., (1994).

The literature has been giving a particular interest to integrated step-up DC-DC converters that are able to convert a power supply voltage into a higher one. As well known, the conversion exploits the energy exchange between an inductor and a capacitor to boost up the input voltage. Although this architecture is quite simple, a control law is required for properly driving the switching activity, which affects the energy exchange Criscione, M., (2001).

The content of this thesis can be summarized briefly as follows:

Chapter 2 focuses on the step-up converter. Firstly, general structure and working principle of DC-DC converter is explained. Then, structure and working principle of step-up converter is explained. Also, the parasitic elements in a step-up converter are explained.

The chapter 3 of this thesis presents the fuzzy logic control. The explanation of fuzzy logic control is given theoretically. At the same time, comparison with the other control systems, advantages and disadvantages are explained. Also choosing of membership function in a fuzzy logic and constitution of rule table are explained as given examples. The information about fuzzy logic controller types is given. The comparison between Mamdami and Sugeno Method is explained. Finally, the information about fuzzy logic control applications in power electronics and its drivers are given.

Chapter 4 focuses on design of step-up converter. In this chapter, firstly, the procedures of the converter design have been explained. Then the choosing of the switching components, which will be used, has been given and the calculation of the inductance and capacitance values, which will be used have been given. Finally, the choosing of the diode, which is convenient with the other design components, has been given. MOSFET IRF Z44N has been used as a switching component, and then fast diode, which has 0.5ns switching speed, has been used as a diode.

Chapter 5 focuses on simulation results and experimental results. Also the comparison of the simulation results between experimental results is given. And selection of fuzzy logic is described.

Chapter 6 is concerned with the conclusions.

CHAPTER TWO
STEP-UP (BOOST) CONVERTERS

2. Control of DC-DC Converters

In dc-dc converters, the average dc output voltage must be controlled to equal a desired level, though the input voltage and the output load may fluctuate. Switch-mode dc-dc converters utilize one or more switches to transform dc from one level to another. In a dc-dc converter with a given input voltage, the average output voltage is controlled by controlling the switch on and off durations (t_{on} and t_{off}). To illustrate the switch-mode conversion concept, consider a basic dc-dc converter shown in Figure 2.1a. The average value V_o of the output voltage v_o in Figure 2.1b depends on t_{on} and t_{off}. One of the methods for controlling the output voltage employs switching at a constant frequency (hence, a constant switching time period T_s=t_{on} + t_{off}) and adjusting the on duration of the switch the control the average output voltage. In this method, called pulse-width modulation (PWM) switching, the switch duty ratio D, which is defined as the ratio of the on duration to the switching time period, is varied.

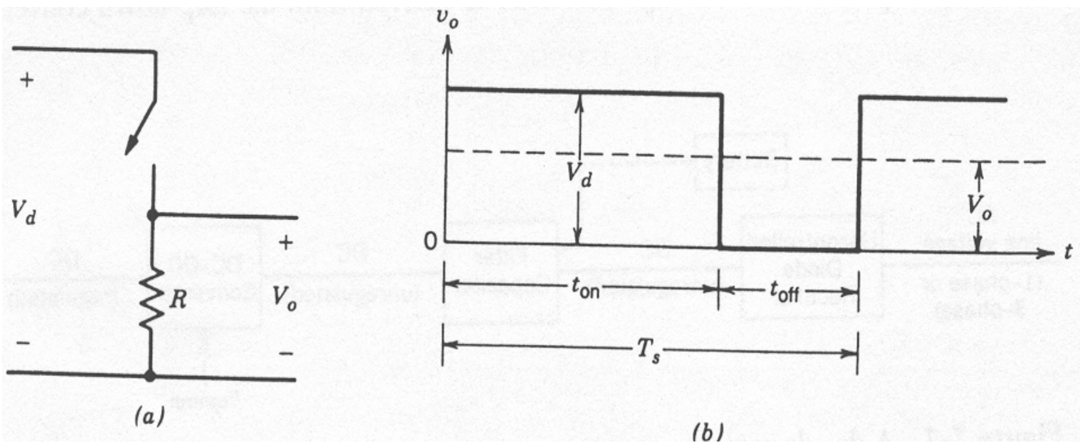


Figure 2.1 Switch-mode dc-dc conversion

In the PWM switching at a constant switching frequency, the switch control signal, which controls the state (on or off) of the switch, is generated by comparing a signal-level control voltage $v_{control}$ with a repetitive waveform as shown in Figure 2.2a and 2b. The control voltage signal generally is obtained by amplifying the error, or the difference between the actual output voltage and its desired value. The frequency of the repetitive waveform with a constant peak, which is shown to be a sawtooth, establishes the switching frequency. This frequency is kept constant in a PWM control and is chosen to be in a few kilohertz to a few hundred kilohertz range. When the amplified error signal, which varies very slowly with time relative to the switching frequency, is greater than the sawtooth waveform, the switch control signal becomes high, causing the switch to turn on. Otherwise, the switch is off. In terms of $v_{control}$ and the peak of the sawtooth waveform $\hat{V}_{st}$ in Figure 2.2, the switch duty ratio can be expressed as

$$D = \frac{t_{on}}{T_s} = \frac{v_{control}}{\hat{V}_{st}} \tag{2-1}$$

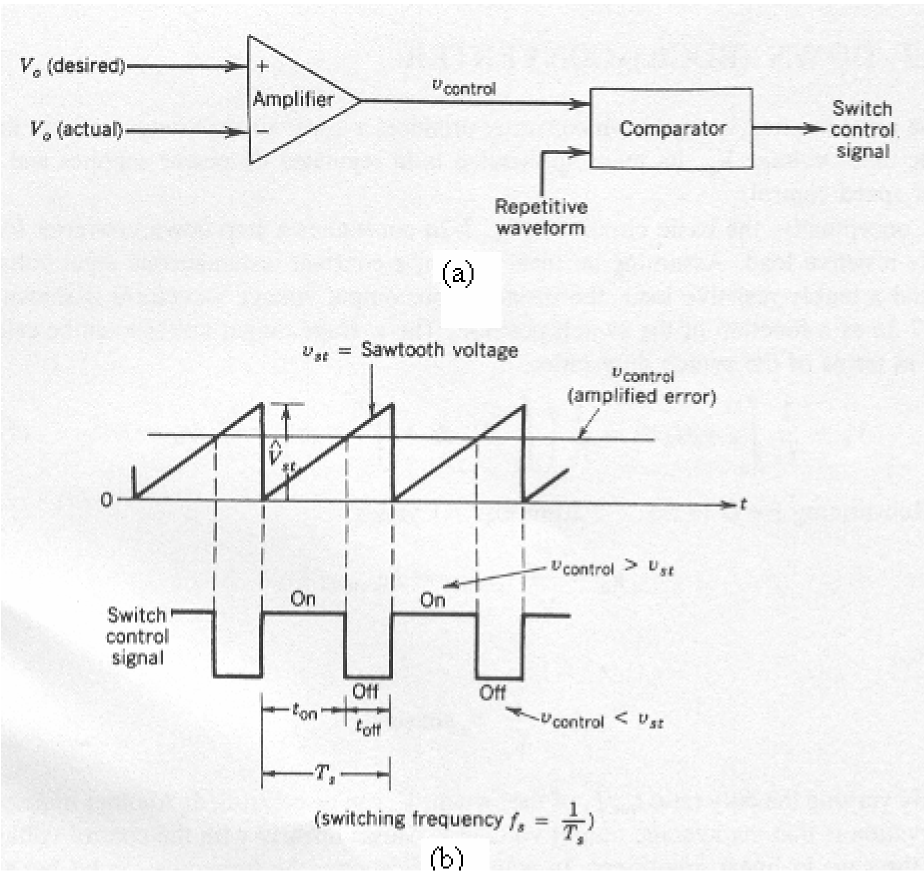


Figure 2.2 Pulse-width modulator: (a) block diagram; (b) comparator signals

The dc-dc converters can have two distinct modes of operation: (1) continuous current conduction and (2) discontinuous current conduction. In practice, a converter may operate in both modes, which have significantly different characteristics. Therefore, a converter and its control should be designed based on both modes of operation.

2.1. Step-Up (Boost) Converter

Figure 2.3 shows a step-up converter. Its main application is in regulated dc power supplies. As the name implies, the output voltage is always greater than the input voltage. When the switch is on, the diode is reversed biased, thus isolating the output stage. The input supplies energy to the inductor. When the switch is off, the output stage receives energy from the inductor as well as from the input. In the steady-state analysis presented here, the output filter capacitor is assumed to be very large to ensure a constant output voltage $v_o(t) \approx V_o$.

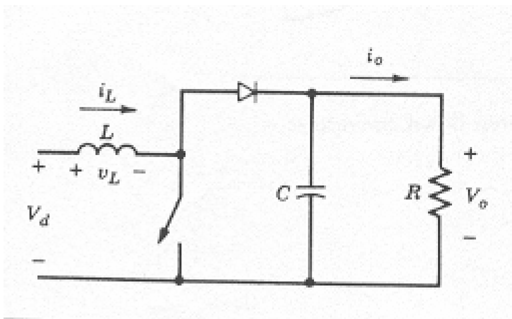


Figure 2.3 Step-up dc-dc converter

2.1.1. Continuous-Conduction Mode

Figure 2.4 shows the steady-state waveforms for this mode of conduction where the inductor current flows continuously $\left[i_L(t)\rangle\, 0\right]$.

Since in steady state the time integral of the inductor voltage over one time period must be zero,

$$V_d t_{on} + (V_d - V_o) t_{off} = 0 \tag{2-2}$$

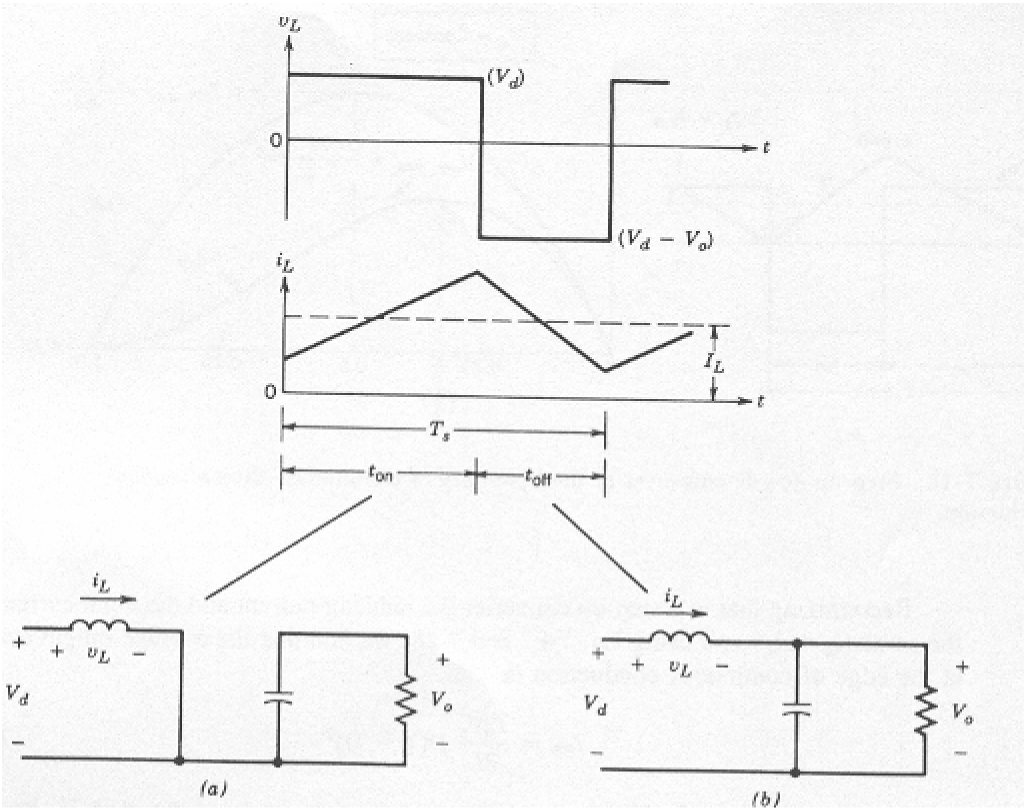


Figure 2.4 Continuous-conduction mode: (a) switch on; (b) switch off

Dividing both sides by T_s and rearranging terms yield

$$\frac{V_o}{V_d} = \frac{T_s}{t_{off}} = \frac{1}{1-D} \tag{2-3}$$

Assuming a lossless circuit, P_d=P_o,

$$\therefore V_d I_d = V_o I_o \tag{2-4}$$

and

$$\frac{I_o}{I_d} = (1 - D) \qquad (2\text{-}5)$$

2.1.2. Boundary Between Continuous and Discontinuous Conduction

Figure 2.5a shows the waveforms at the edge of continuous conduction. By definition, in this mode i_L goes to zero at the end of the off interval. The average value of the inductor current at its boundary is

$$\begin{aligned} I_{LB} &= \frac{1}{2} i_{Lpeak} \\ &= \frac{1}{2} \frac{V_d}{L} t_{on} \\ &= \frac{T_s V_o}{2L} D(1 - D) \qquad (2\text{-}6) \end{aligned}$$

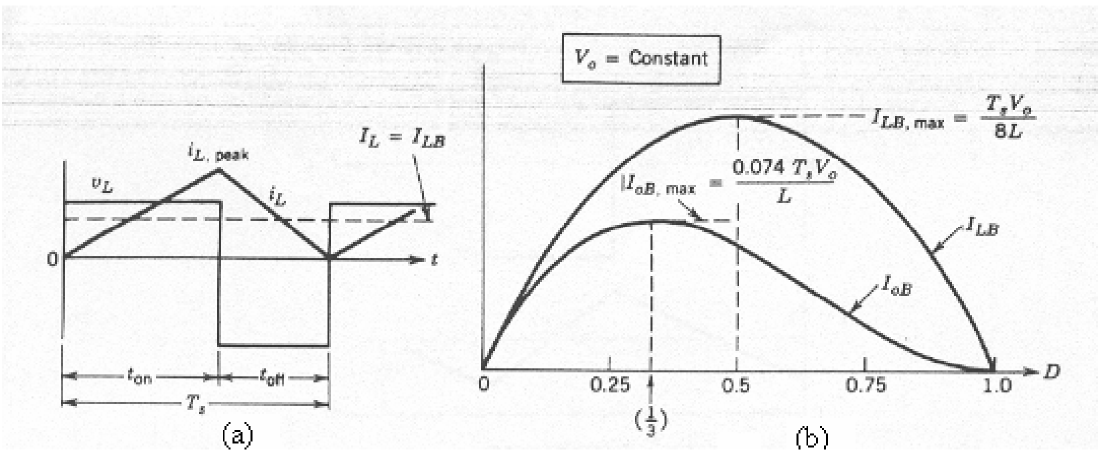


Figure 2.5 Step-up dc-dc converter at the boundary of continuous-discontinuous conduction

Recognizing that in step-up converter the inductor current are the same $(i_d = i_L)$ and using Eq.2-1 and 2-2, we find that the average output current at the edge of continuous conduction is

$$I_{oB} = \frac{T_s V_o}{2L} D(1-D) \tag{2-7}$$

Most applications in which a step-up converter is used require that V_o be kept constant. Therefore, with V_o constant, I_{oB} are plotted in Figure 2.5b as a function of duty ratio D. Keeping V_o constant and varying the duty ratio imply that the input voltage is varying.

Figure 2.5b shows that I_{LB} reaches a maximum value at D= 0.5:

$$I_{LB,\max} = \frac{T_s V_o}{8L} \tag{2-8}$$

Also, I_{oB} has its maximum at D=$\frac{1}{3}$ = 0.333:

$$I_{oB,\max} = \frac{2}{27}\frac{T_s V_o}{L} = 0.074\frac{T_s V_o}{L} \tag{2-9}$$

In terms of their maximum values, I_{LB} and I_{oB} can be expressed as

$$I_{LB} = 4D(1-D)I_{LB,\max} \tag{2-10}$$

and

$$I_{oB} = \frac{27}{4}D(1-D)^2 I_{oB,\max} \tag{2-11}$$

Figure 2.5b shows that for a given D, with constant V_o, if the average load current drops below I_{oB} (and, hence, the average inductor current below I_{LB}), the current conduction will become discontinuous.

2.1.3. Discontinuous-Conduction Mode

To understand the discontinuous-current-conduction mode, we would assume that as the output load power decreases, V_d and D remain constant (even thought, in practice, D would vary in order to keep V_o constant). Figure 2.6 compares the waveforms at the boundary of continuous conduction and discontinuous conduction, assuming that V_d and D are constant.

In Figure 2.6b, the discontinuous current conduction occurs due to decreased $P_o (= P_d)$ and, hence, a lower $I_L (= I_d)$, since V_d is constant. Since $i_{L,peak}$ is the same in both modes in Figure 2.6, a lower value of I_L (and, hence a discontinuous i_L) is possible only if V_o goes up in Figure 2.6.

If we equate the integral of the inductor voltage over one time period to zero,

$$V_d DT_s + (V_d - V_o)\Delta_1 T_s = 0$$

$$\therefore \frac{V_o}{V_d} = \frac{\Delta_1 + D}{\Delta_1} \tag{2-12}$$

and

$$\frac{I_o}{I_d} = \frac{\Delta_1}{\Delta_1 + D} \text{ (since } P_d = P_o\text{)} \tag{2-13}$$

From Figure 2.6, the average input current, which is also equal to the inductor current, is

$$I_d = \frac{V_d}{2L} DT_s (D + \Delta_1) \tag{2-14}$$

Using Eq.2-13 in the foregoing equation yields

$$I_o = \left(\frac{T_s V_o}{2L} \right) D\Delta_1 \tag{2-15}$$

In practice, since V_o is held constant and D varies in response to the variation in V_d, it is more useful to obtain the required duty ratio D as a function of load current for various values of $\frac{V_o}{V_d}$. By using Eqs. 2-12, 2-15, and 2-9, we determine that

$$D = \left[\frac{4}{27} \frac{V_o}{V_d} \left(\frac{V_o}{V_d} - 1 \right) \frac{I_o}{I_{oB,max}} \right]^{\frac{1}{2}} \tag{2-16}$$

In Figure 2.7, D is plotted as a function of $I_o / I_{oB,max}$ for various values of V_d / V_o. The boundary between continuous and discontinuous conduction is shown by the dashed curve.

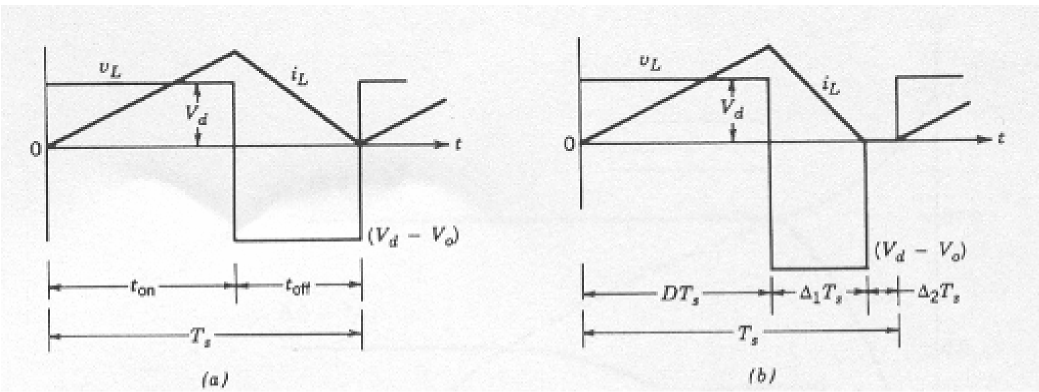


Figure 2.6 Step-up converter waveforms: (a) at the boundary of continuous -discontinuous conduction; (b) at discontinuous conduction

In the discontinuous mode, if V_o is not controlled during each switching time period, at least

$$\frac{L}{2} I^2_{L,peak} = \frac{(V_d D T_s)^2}{2L} \quad \text{W-s} \tag{2-17}$$

are transferred from the input to the output capacitor and to the load. If the load is not able to absorb this energy, the capacitor voltage V_o would increase until an energy balance is established. If the load becomes very light, the increase in V_o may cause a capacitor breakdown or a dangerously high voltage to occur.

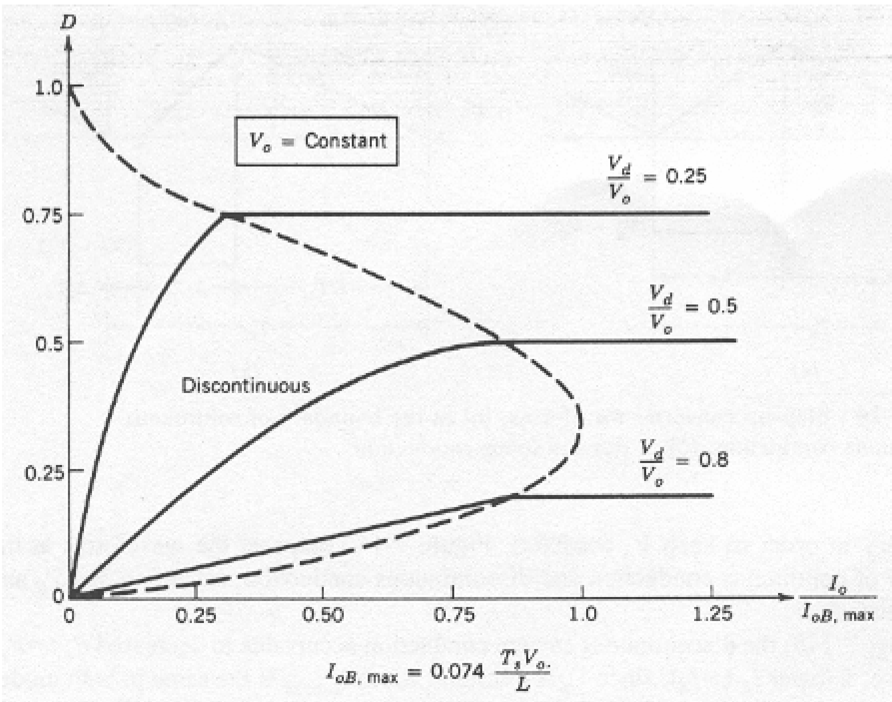


Figure 2.7 Step-up converter characteristics keeping V_o constant

2.1.4. Effect of Parasitic Elements

The parasitic elements in a step-up converter are due to the losses associated with the inductor, the capacitor, the switch, and the diode. Figure 2.8 qualitatively shows the effect of these parasitic on the voltage transfer ratio. Unlike the ideal characteristic, in practice, V_o/V_d declines as the duty ratio approaches unity. Because of very poor switch utilization at high values of duty ratio, the curves in this range are shown as dashed. These parasitic elements have been ignored in the simplified analysis presented here; however, these can be incorporated into circuit simulation programs on computers for designing such converters.

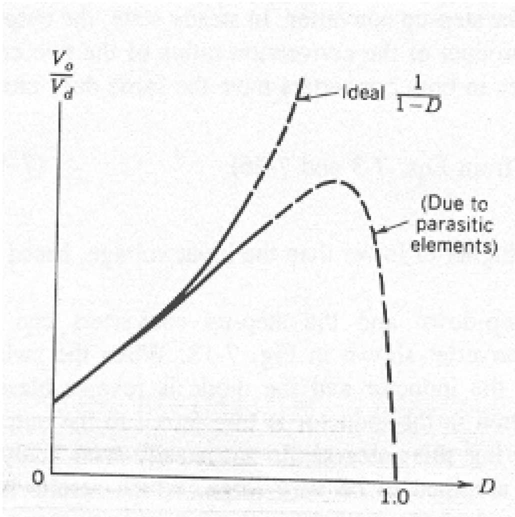


Figure 2.8 Effect of parasitic elements on voltage conversion ratio (step-up converter)

2.1.5. Output Voltage Ripple for Continuous Mode

The peak-to-peak ripple in the output voltage can be calculated by considering the waveforms shown in Figure 2.9 for a continuous mode of operation. Assuming that all the ripple current component of the diode current i_D flows through the capacitor and its average value flows through the load resistor, the shaded area in Figure 2.9 represents charge ΔQ. Therefore, the peak-peak voltage ripple is given by

$$\Delta V_o = \frac{\Delta Q}{C} = \frac{I_o D T_s}{C} \quad \text{(Assuming a constant output current)}$$

$$= \frac{V_o}{R}\frac{DT_s}{C} \tag{2-18}$$

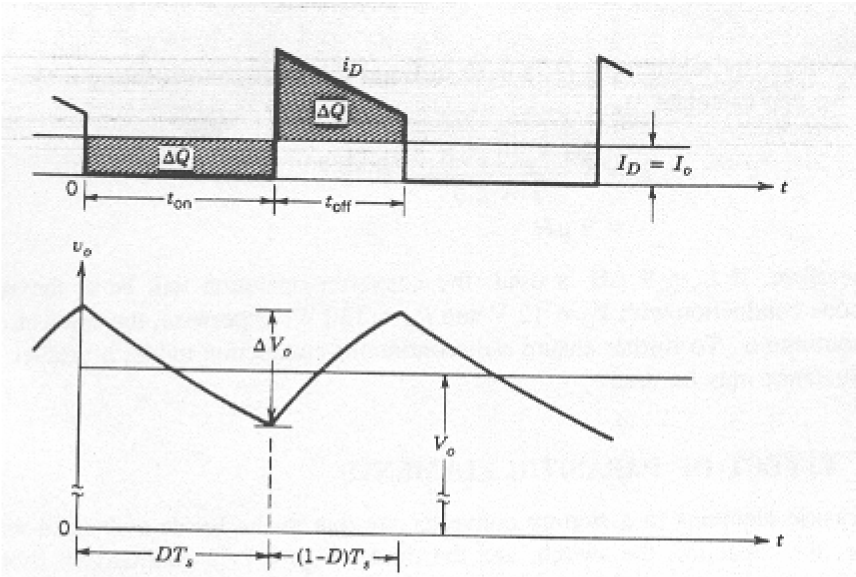


Figure 2.9 Step-up converter output voltage ripple

$$\therefore \frac{\Delta V_o}{V_o} = \frac{DT_s}{RC}$$

$$= D\frac{T_s}{\tau} \quad \text{(where } \tau = \text{RC time constant)} \qquad (2\text{-}19)$$

CHAPTER THREE
FUZZY LOGIC CONTROL

3. Theoretical Aspects of the Fuzzy Control

3.1. Introduction

The real world is complex; complexity in the world generally arises from uncertainty in the form ambiguity. Problems featuring complexity and ambiguity have been addressed subconsciously by humans since they could think; these ubiquitous features pervade most social, technical, and economic problems faced by the human race. Humans have the capacity to reason approximately, a capability that computers currently do not have. In reasoning about a complex system, humans reason approximately about its behavior, thereby maintaining only a generic understanding about the problem. Fortunately, this generality and ambiguity are sufficient for human comprehension of complex systems. As the quote above from DR. Zadeh's principle of incompatibility suggests, complexity and ambiguity (imprecision) are correlated: "The closer one looks at a real-world problem, the fuzzier becomes its solution" Zadeh,L.A., (1973).

Fuzzy systems approximate functions. They are universal approximators if they use enough fuzzy rules. In this sense fuzzy systems can model any continuous function or system. Those systems can just as well come from physics or sociology as from control theory or signal processing.

The quality of the fuzzy approximation depends on the quality of the rules. In practice experts guess at the fuzzy rules. Or neural schemes learn the rules from data and tune the rules with new data. The result always approximates some unknown nonlinear function that can change in time.

A fuzzy system can help model or control a system when we do not have a math model of how the system's output depends on its input. The fuzzy system uses commonsense rules in place of the math model or the so-called plant model.

The fuzzy system itself is a function or mapping. It is a set of fuzzy if-then rules that maps inputs to outputs. It converters stimuli to responses or sensor measurements to control actions. The rules might have the verbal form "If the wash water is very dirty then add much more detergent" or "If the error is small and positive then turn the wheel a little to the left" or "If the air is cool then set the motor speed to slow." The inputs and outputs can be numbers or vectors of numbers. These rule-based systems can in theory model any system.

3.1.1. The Benefits of Fuzzy Control

Considering the existing applications of the fuzzy control, which range from very small, micro-controller based systems in home applications to large-scale process control systems, the advantages of the fuzzy control using usually fall into one of following five categories.

1. Less dependence on mathematical models

The fundamental advantages of the fuzzy control over conventional methods are a less dependence on mathematical models. Distinctively from traditional approaches that require an accurate knowledge of the different block-boxes forming the system, fuzzy control needs only some knowledge (a priori, experimental or expert) on the overall system. The fuzzy logic approach in this way considers as a behavior approach.

2. Implementing expert knowledge

In many cases of industrial process control the degree of automation is quite low. There is variety of basic, conventional control loops, but a human operator is needed during starting or closing phase, for parametrizing the controllers, or switching between different control modules. The knowledge of this operator is usually based on experience and does not lend itself to being expired in differential equations. It is often rather of the type "if the situation is such and such, I should do the following." In this case, fuzzy control offers a method for representing and implementing the expert's knowledge because the base of fuzzy control is IF… THEN… rules.

3. Non-linear control

In real life we deal basically with non-linear systems. Fuzzy control is non-linear control method in nature, so that give possibility to control by real systems more effective.

4. Robust control

Consider the following problem: robot arm with several links has to move object with different masses along predefined path. Since there are good exact models of this system available, it is not too big a problem to realize a PID-controller that works pretty well for known masses within a narrow range. Substantial parameter changes, however, or major external disturbances, lead to a sharp decrease in performance. In the presence of such disturbance, PID-systems usually are faced with a trade-off between fast re-action with significant overshoot or smooth but slow reactions, or they even run into problems in stabilizing the system at all. In this case, fuzzy control offers ways to implement simple but robust solution that covers a wide range of system parameters and that can cope with major disturbances.

5. New possibility for further automatization

There are many complex engineering tasks that deeply involve experienced human thinking, for example, design of induction machines. Here if even every design step is founded on well-known basic electromagnetic relations, yet the human experience is hardly convertible into a crisp automatic procedure. But fuzzy logic greatly helps to manage the experience transfer from human to integrated automatic procedures because a consequence of performance analysis is made by skilled designer is based on set of rules If {result of comparison} then {modification to carry out}.

However, set of rules as that is a base of fuzzy control so the fuzzy logic inference mechanism substituting the human being in proposing the design modifications close the design loop, that is therefore made design process completely automatic.

3.1.2. Disadvantages of Fuzzy Control

Fuzzy logic control systems are essentially non-linear systems. For this reason it is difficult to obtain general results on the analysis and design of fuzzy logic controller. Also stability analysis is of extreme importance, and the lack of satisfactory formal techniques for studying the stability of process control systems involving fuzzy logic controller has been considered a major drawback of fuzzy logic controller. Characteristic feature of fuzzy logic method is the expert knowledge that is required. Expert knowledge relies on any knowledge, experimental or theoretical, on the system. It is obvious that for some cases it is very difficult, may be impossible, to have enough experimental knowledge. On the other hand, it is possible to have some theoretical knowledge, even by rough modelization or simulation. In this case, fuzzy logic approach may be considered since a rigorous mathematical model is not requisite. Furthermore as the fuzzy logic approach is a behavior one, some information's lack or some disturbances may be tolerated.

3.2. The Fuzzy Control Mathematics

3.2.1. Membership Functions

Let see what lies in the base of fuzzy control.

In fuzzy set theory, 'normal' sets are called crisp sets, in order to distinguish them from fuzzy sets. Let C be a crisp set defined on universe U, then for any element u of

U, either u $\in$ or u $\notin$ C. In fuzzy set theory this property is generalized, therefore in a fuzzy set F, it is not necessary that either u $\in$ F or u$\notin$ F Driankov, D., (1993).

The generalization is performed as follows. For any crisp set C it is possible to define a characteristic function $\mu_C : U \Rightarrow \{0, 1\}$ as

$$\mu_A(x) = \begin{cases} 1\_\_when\_x \in A, \\ 0\_\_when\_x \notin A. \end{cases}$$

In fuzzy set theory, the characteristic function is generalized to a membership function that assigns to every u $\in$ U a value from unit interval [0, 1] instead from two-element set {0, 1}. The set that is defined on the basis of such an extended membership function is called a fuzzy set.

So, the main point of fuzzy set theory is a membership function that is defined as: The membership function μ_F of a fuzzy set F is a function

$$\mu_F : U \Rightarrow [0, 1].$$

So, every element u from U has a membership degree $\mu_F(u) \in [0, 1]$. F is completely determined by the set of tuples

$$F = \{(u, \mu_F(u)) \mid u \in U\}.$$

For example, suppose someone wants to describe the class of cars having the property of being expensive, by considering cars such as BMW, Buick, Ferrari, Fiat, Lada, Mercedes and Rolls Royce. Some cars, like Ferrari or Rolls Royce belong to this class, while other cars, like Fiat or Lada do not belong to it. But there is a third group of cars, where it is difficult to state whether they are expensive cars is e.g.

{(Ferrari, 1), (Rolls Royce, 1), (Mercedes, 0.8), (BMW, 0.7), (Buick, 0.4)}, i.e., Mercedes belong to degree 0.8, BMW to 0.7, Buick to 0.4 to the class of expensive cars.

Or we can define the set of natural numbers ‘close to 10’ as

$\tilde{1}\tilde{0}$ = (5, 0.1)+(8, 0.6)+(10, 1)+(12, 0.6)+(15, 0.1).

Early, without definition of membership function we could not answer on the question, for example, close number 5 or number 6 to 10 or not. Now we can do that and can answer how 5 or 6 close to number 10.

Thus, fuzzy logic, unlike Boolean or crisp logic, deals with problem that have vagueness, uncertainly, or impression, and uses membership functions with values varying between 0 and 1. Fuzzy logic tends to mimic human thinking that is often fuzzy in nature. In conventional set theory based on Boolean logic, particular object or variable is either a member of given set or it is not. On the other hand, in fuzzy set theory based on fuzzy logic, a particular object has a degree of membership in given set that may be anywhere in range of 0 (completely not in the set) to 1 (completely in the set). This property allows fuzzy logic to deal with uncertain situations in fairly natural way.

In fuzzy control, there are usually four kinds of fuzzy sets that are dealt with, namely ‘increasing’, ‘decreasing’, and two kinds of ‘approximating’ notions. And for each kind of fuzzy sets there is own shape that represents set.

The increasing membership function with straight lines is called Γ-functions, because of the similarity of these functions with this character. This function is a function of one variable and two parameters defined as follows Figure 3.1.

$$\Gamma(u,A,B)=\begin{cases}0 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ u\langle A \\ (u-A)/(B-A) \_\_ A\le u\le B \\ 1 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ u\rangle B\end{cases}$$

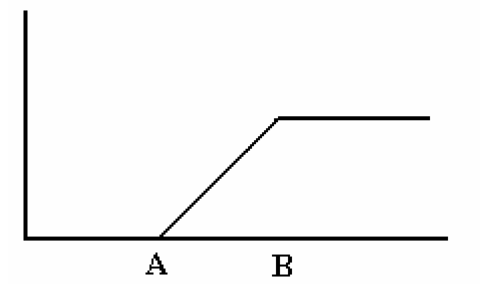


Figure 3.1 An example of a Γ - function

The decreasing membership function with straight lines is called L-function; it is defined as follows Figure 3.2:

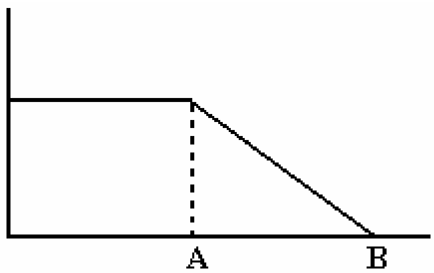


Figure 3.2 An example of a L- function

Bell-shaped membership function with straight lines is called Λ-function or triangular function; it is defined as follows Figure 3.3:

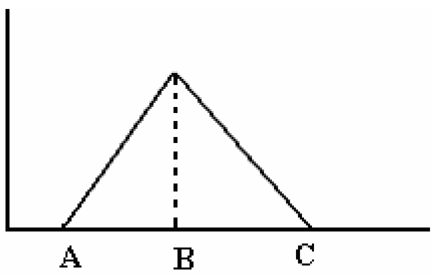


Figure 3.3 An example of a Λ - functio

Approximating membership function with straight lines, where the top is not one point but an interval, is called Π -function; it is defined as follows Figure 3.4:

$$\Pi(u, A; B; C; D) = \begin{cases} 0\_\_\_\_\_\_\_\_\_\_\_\_\_u \langle A \\ (u - A)/(B - A)\_\_A \le u \le B \\ 1\_\_\_\_\_\_\_\_\_\_\_\_\_B \le u \le C \\ (C - u)/(D - C)\_C \le u \le D \\ 0\_\_\_\_\_\_\_\_\_\_\_\_\_u \rangle D \end{cases}$$

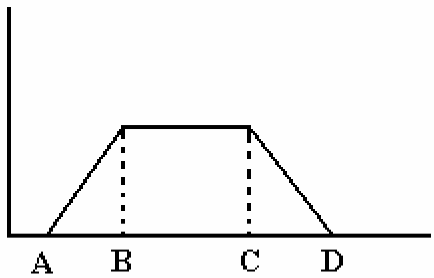


Figure 3.4 An example of a Π - function

Usually, in fuzzy logic are used smooth membership function, for example, Zadeh's bell-shaped Π -function or S-function but in fuzzy control we use membership functions with straight lines because it is faster from computational point of view and gives appropriate results.

3.2.2. Operations on Fuzzy Sets

Basic operations that are used in fuzzy control are union, intersection and complement. Let define these operations.

In classical set theory the union, intersection and complement of sets are simple operations that are unambiguously defined. The unambiguity follows from the fact that the logical operations and, or, not have a well defined semantics based on propositional logic. For example, "A and B" in propositional logic is true if and only if both expressions A and B are true. In fuzzy set theory their interpretation is not so simple, because graded characteristic functions are used. Zadeh proposed:

$$\forall x \in X : \mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$$
$$\forall x \in X : \mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$$
$$\forall x \in X : \mu_{A'}(x) = 1 - \mu_A(x)$$

3.2.3. Fuzzy Relations

Fuzzy relations are very important operations in fuzzy control because they can describe interaction between variables. This is particularly interesting in if-then rules.

A relation can be considered as a set of tuples, where a tuple is an ordered pair. A fuzzy relation is a fuzzy set of tuples, i.e. each tuple has a membership degree between 0 and 1.

Or more strictly, let U and V be uncountable (continues) universes, and $\mu_R : U \times V \to [0,1]$, then

$$R = \int_{U.V} \mu_R(u,v)/(u,v)$$

is a binary fuzzy relation on $U \times V$. If U and V are countable (discrete) universes, then

$$R = \sum_{U.V} \mu_R(u,v)/(u,v)$$

The symbol denotes the set of all tuples $\mu_R(u,v)/(u,v)$ on $U \times V$.

For example, each rule of fuzzy logic controller: if E is PB and ΔE is PS, then ΔU is NM, where PB, PS and NM are fuzzy sets, defined on universe of discourse E, ΔE and ΔU respectively, we can represent by ternary fuzzy relation R defined as:

$$R = \int_{E\times\Delta E\times\Delta U} \min\left(\mu_{PB}(E),\mu_{PS}(\Delta E),\mu_{NM}(\Delta U)\right)/\left(E,\Delta E,\Delta U\right)$$

i.e., each triple $(E,\Delta E,\Delta U)$ has a membership degree equal to the minimum of three membership functions.

The rule base of fuzzy logic controller is fuzzy relation, so it is necessary to know that operations there are on fuzzy relations.

In fuzzy control we use fuzzy sets that are defined on discrete universe because our fuzzy logic controller is designed on microprocessors or DSP-based controllers that is why I'll give definitions of fuzzy relation operations in binary case.

There are two most important operations on fuzzy sets and fuzzy relations. These are projection and cylindrical extension. The projection operation brings a ternary relation back to a binary relation, or a binary relation to a fuzzy set, or a fuzzy set to a single crisp value.

Let R be defined on $X\times Y$, then

$$\text{proj\_R\_on\_Y} = \int_{Y} \sup \mu_R(x,y)/y$$

For example, consider the relation R as

	y_1	y_2	y_3	y_4
x_1	0.3	0.7	0.8	0.3
x_2	0.1	1.0	0.6	0.5
x_3	0.5	0.7	0.8	0.8

When the projection on X means that x_1 is assigned the highest membership degree from the tuples (x_1, y_1), (x_1, y_2), (x_1, y_3), and (x_1, y_4), i.e., 0.8 (the maximum of the first row). Analogously we define the highest membership degrees in case x_2 and x_3.

So one obtains the fuzzy set

$$\text{proj}\, R \text{ on } X = 0.8/x_1 + 1/x_2 + 0.8/x_3$$

The projection operation is almost always used in combination with the cylindrical extension. It extends fuzzy sets to binary relations, fuzzy binary relations to fuzzy ternary relations, etc. It mainly serves the following goal: let A be a fuzzy set defined on X, and let R be a fuzzy relation defined on $X \times Y$, then, of course, it is not possible to take the intersection of A and R, but when A is extended to $X \times Y$, this is possible.

Let F be a fuzzy set defined on Y, in binary case the cylindrical extension of F on $X \times Y$ is the set of all tuples $(x, y) \in X \times Y$ with membership degree equal to $\mu_F(y)$, i.e.,

$$ce(F) = \int_{x \bullet y} \mu_F(y)/(x, y)$$

Consider the fuzzy set (from previous example)

$$A = \text{proj}\, R \text{ on } X = 0.8/x_1 + 1/x_2 + 0.8/x_3$$

Its cylindrical extension on the domain $X \times Y$ is given

	y_1	y_2	y_3	y_4
x_1	0.8	0.8	0.8	0.8
x_2	1.0	1.0	1.0	1.0
x_3	0.8	0.8	0.8	0.8

The combination of fuzzy sets and fuzzy relation with aid of cylindrical extension and projection is called composition. It is denoted by.

Let A be a fuzzy set defined on X and R be a fuzzy relation defined on $X \times Y$. Then the composition of A and R resulting in a fuzzy set B defined on Y is given by

$$B = A \circ R = \text{proj}(\text{ce}(A) \cap R) \text{ on } Y$$

or, if intersection is performed with the maximum operation and projection with minimum,

$$\mu_B(y) = \max_x \min(\mu_A(x), \mu_R(x, y)).$$

This is called max-min composition. If intersection is performed with maximum and projection with product, we have

$$\mu_B(y) = \max_x \mu_A(x) \bullet \mu_R(x, y).$$

This called max-dot or max product composition.

The composition is very important operation for fuzzy control. Using this operator we define output of fuzzy system on certain input. For example, let after fuzzification our input is fuzzy set $A = 1/x_1 + 0.6/x_2 + 0.5/x_3$ and relationship between input fuzzy set and output fuzzy set carry out through relation

	y_1	y_2	y_3
x_1	0.1	0.4	0.7
x_2	1.0	0.8	0.6
x_3	0.7	0.2	0.1

In this case, output fuzzy set is following $B = A \circ R = 0.6/y_1 + 0.6/y_2 + 0.7/y_3$.

3.2.4. Approximate Reasoning

The fundamental knowledge representation unit in appropriate reasoning is the notion of a linguistic variable. Linguistic variable is a variable whose value is words or sentences in a natural or artificial language Driankov, D., (1993).

It is usual in approximate reasoning to have the following framework associated with the notion of linguistic variable: (A, LA, X, Mx)

Here A denotes the symbolic name of a linguistic variable, i.e., error or change-of-error. LA is the set of linguistic values that X can take. In case of linguistic variable error E we have LE = {NB, NS, Z, PS, PB}. LE is so called the term-set of E. X is the actual physical domain over which the linguistic variable A takes its quantitative (crisp) values. In the case of error and change-of-error, though there is an actual physical domain, one often uses a 'normalized domain' [-1, 1]. X can be discrete or continuous. Mx is a function which takes a symbol as its argument, e.g., error, and returns the 'meaning' or the error-symbol in terms of fuzzy set.

The base of fuzzy controller, representing our knowledge about plant is fuzzy If-then statements.

A fuzzy conditional or fuzzy if-then production rule is symbolically expressed as If [fuzzy proposition] then [fuzzy proposition] for example, if E and ΔE are process state variables and ΔU is the control output variable then

IF E is NB AND ΔE is PB THEN ΔU is PS

is a symbolic expression of the following causal relationship stated in natural language form:

if it is the case that the current value of E is negative big and the current value of ΔE is positive big then this is a cause for a small increase in previous value of the control output.

For using if-then statements in our mathematics we need operation on these statements. This operation is implication that expresses meaning of if X is A then Y is B. There are some different forms of implication, for example, Zadeh implication, Lucasiewicz implication, Sharp implication but with respect to fuzzy control the most important implication is Mamdami implication. Its definition is based on the intersection operation, i.e.,

$$p \to q \equiv p \cap q .$$

The relation R_c (c from conjunction) is defined as

$$\mu_{Rc}(x, y) = \min(\mu_A(x), \mu_B(y)).$$

In a system of n rules each rule is symbolically represented as

IF E is LE (k) THEN U is LU (k), k=1,….,n.

Considering a set of n rules and the Mamdami type of individual-rule based inference, one obtains the clipped fuzzy sets CLU(1),…CLU(n) that have to be combined to obtain one output fuzzy set U,

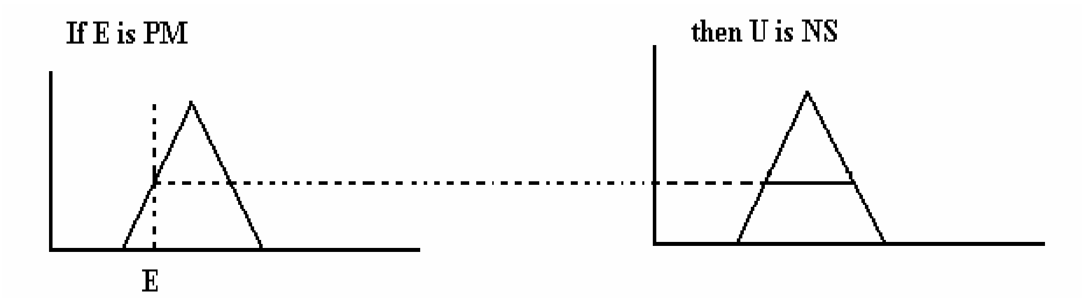


Figure 3.5 The graphical representation of the firing of the rule if E is PM, then U is NS. The thick lines in NS denote the clipped fuzzy set CNS

$$U = \bigcup_{k=1}^{n} CLU^{(K)}$$

Thus, to drive output fuzzy set there are two type of inference:

1. Compositional based inference. One first combines all rules into R_M and then 'fires' with fuzzy input μ via operation composition.

$$U = \mu(E) \circ R_M$$

2. Individual-rule based inference. One first 'fires' rules individually with crisp input E and thus obtain n clipped fuzzy sets that are combined into one overall fuzzy set.

$$U = \bigcup_{k=1}^{n} CLU^{(K)} .$$

3.3. Fuzzy Logic Controller Design Parameters

3.3.1. The structure of a Fuzzy Logic Controller

The principal structure of fuzzy logic controller (FLC), as illustrated in Figure 3.6 consists of the following components Driankov, D., (1993).

Fuzzification Module

The fuzzification module (FM) performs the following function:

- FM-F1: Performs a scale transformation (i.e., an input normalization) which maps the physical values of the current process state variables into a normalized universe of discourse (normalized domain). When a non-normalized domain is used then there is no need for FM-F1.
- FM-F2: Performs the so-called fuzzification which converts a point-wise (crisp), current value of a process state variable into a fuzzy set, in order to make it compatible with the fuzzy set representation of the process state variable in the rule-antecedent.

The design parameter of the fuzzification is "Choice of fuzzification strategy".

The above choice is determined by the type of the inference engine or rule firing employed in the particular application of FLC. Thus there are only two choices available: (1) fuzzification in the case when inference is composition based and (2) fuzzification in the case when inference is individual-rule-firing based.

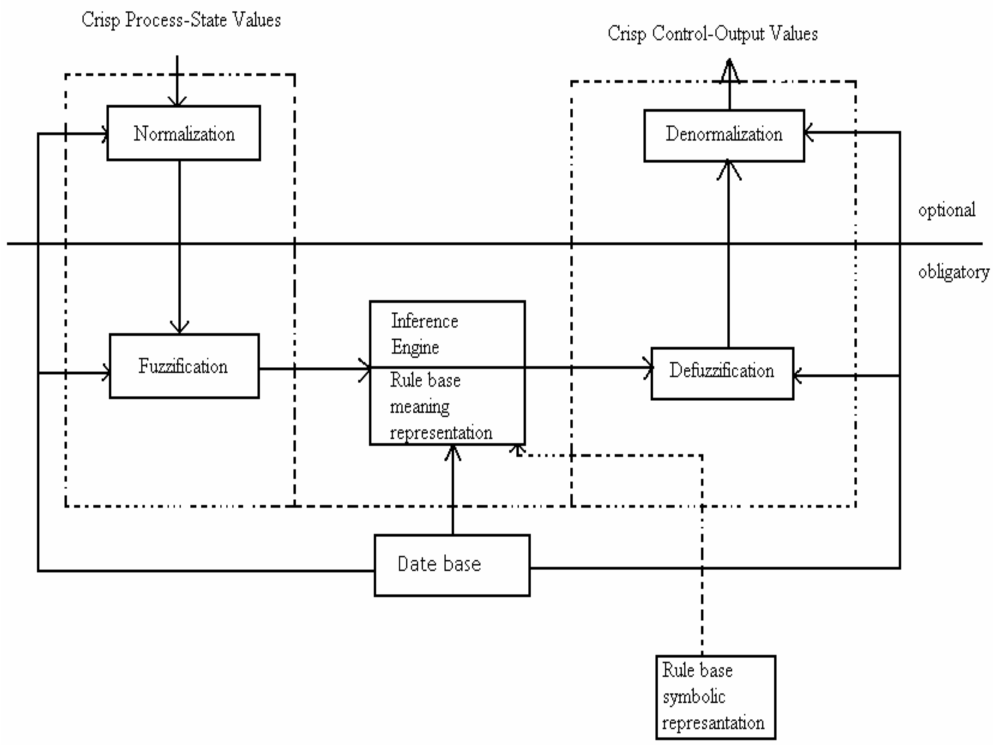


Figure 3.6 The structure of FLC

Knowledge Base

The knowledge base of a FLC consists of a data base and a rule base.

The basic function of the data base is to provide the necessary information for the proper functioning of the fuzzification module, the rule base, and the defuzzification module. This information includes:

- Fuzzy sets (membership functions) representing the meaning of the linguistic values of the process state and control output variables.
- Physical domains and their normalized counterparts together with the normalization/ de-normalization (scaling) factors.

If the continuous domains of the process state and control output variables have been discretized then the data base also contains information concerning the quantizated look-up tables defining the discretizational policy. The design parameter of data base is

- ❖ Choice membership functions.

The basic function of the rule base is to represent in a structured way the control policy of an experienced process operator and/or control engineer in the form of a set of production rules such as

if (process state) then (control output).

These proposition state the linguistic values which the control output variables take whenever the current process state matches (at least to a certain degree) the process state description in the rule-antecedent. The design parameters involved in construction of the rule base include:

- ❖ Choice of process state and control output variables,
- ❖ Choice of the contents of the rule-antecedent and the rule-consequent,
- ❖ Choice of term-sets (ranges of linguistic values) for the process state and control output variables,
- ❖ Derivation of the set of rules.

Inference Engine

There are two basic types of approaches employed in the design of the inference engine of fuzzy logic controller:

1. Composition based inference
2. Individual based inference

So the design parameter for the inference engine is

- ❖ Choice inference engine
- ❖ Testing the set of rules for consistency and completeness.

Defuzzification Module

The function of the defuzzification module is as follows:

- DM-F1: Performs the so-called defuzzification which converts the set of modified control output values into a single point-wise value.
- DM-F2: Performs an output de-normalization which maps the point-wise value of the control output onto its physical domain. DM-F2 is not needed if non-normalized domains are used.

The design parameter of defuzzification module is

- ❖ Choice of defuzzification operators.

3.3.2. Choice of Design Parameters

In this chapter the available choices for design parameters of a FLC are presented and also the relevance of some of these choices for the performance of the controller.

Choice of Variable and Contents of Rules

If one has made the choice of designing a P-, PD-, PI-, OR PID-like FLC this already implies the choice of process state and control output variables, as well as the content of the rule-antecedent and the rule-consequent for each of the rules. The process state variables representing the contents of the rule-antecedent (if-part of a rule) are selected among

- error, denoted by

- change-of-error, denoted by ΔE,
- sum-of-error, denoted by δE.

The control output (process input) variables representing the contents of the rule-consequent (then-part of the rule) are selected among

- change-of-control output, denoted by ΔU,
- control output, denoted by U.

Furthermore, by analogy with a conventional controller, we have that

- $E(k) = Y_{sp} - Y(k)$,
- $\Delta E(k) = E(k) - E(k-1)$,
- $\Delta U(k) = U(k) - U(k-1)$.

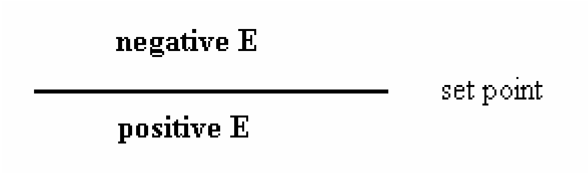
In the above expressions, Y_{sp} stays for desired process output or set-point: Y is the process output variable (control variable); k is the sampling time.

Choice of a Term Set

The linguistic value, members of the term set, are expressed as tuples of the form (value sign, value magnitude), e.g., (positive big), (negative small), etc. The value sign component of such a tuple takes on either one of the following two values: positive and negative. The value magnitude component can take on any number of linguistically expressed magnitudes, e.g., (zero, small, medium, big, very big), etc. The meaning of a tuple, in the case of a PI-like FLC, can be summarized as follows:

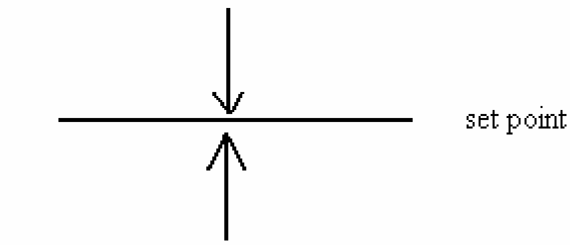
- Linguistic values of E with a negative sign mean that the current process output has a value above the set-point Y_{sp} since $E(k) = Y_{sp} - Y(k) \langle 0$. The magnitude of a negative value describes the magnitude of the difference $Y_{sp} - Y$. Analogously in the case of linguistic values E with positive sign.

Thus,



- Linguistic values of ΔE with a negative sign mean that the current process output $Y(k)$ has increased when compared with its previous values $Y(k-1)$ since $\Delta E(k) = E(k) - E(k-1) = -(Y(k) - Y(k-1)) < 0$. The magnitude of such a negative value is given by the magnitude of this increase. Linguistic values of $\Delta E(k)$ with a compared to $Y(k-1)$. The magnitude of a value is the magnitude of the decrease.
- A linguistic value of “zero” for E means that the current process output is at set-point. A “zero” for ΔE means that the current process output has not changed from its previous value, i.e., $-(Y(k) - Y(k-1)) = 0$.
- A linguistic values of $\Delta U(k)$ with a positive sign mean that the value of the control output $U(k-1)$. The reason for this is that $U(k) = U(k-1) + \Delta U(k)$. The magnitude of a value is the magnitude of increase/decrease of the value of $U(k-1)$.

Now using these meanings we can construct rule base. In control process necessary that output of system aspires to set-point



For example, if output of system below set-point and at same time ΔE is positive, i.e. , Y is moving away from the set-point then a positive change $\Delta U(k)$ in the previous control output $u(k-1)$ is intended to reverse this trend and make Y, instead of moving away from the set-point, to start moving towards it, so rule will be

if E is PB and ΔE is PB then ΔU is PB.

Or if $y(k)$ is much above the set-point $(E(k) \text{is NB})$ and it is moving toward the set-point with a small step $(\Delta E(k) \text{is PS})$ then the magnitude of this step has to be significantly increased $(\Delta U(k) \text{is NM})$.

According to these reasons we construct rule base that in tabular form are illustrated in Table 3.1.

Table 3.1 The rule base of a PI-like FLC tabular form

ce e	NB	NM	NS	Z	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	Z
NM	NB	NB	NB	NM	NS	Z	PS
NS	NB	NB	NM	NS	Z	PS	PM
Z	NB	NM	NS	Z	PS	PM	PB
PS	NM	NS	Z	PS	PM	PB	PB
PM	NS	Z	PS	PM	PB	PB	PB
PB	Z	PS	PM	PB	PB	PB	PB

It has to be noticed that the size of the term set determines the granularity of the control action of the FLC. If one desires better control resolution around the set-point one can consider a larger range of linguistic values. However, one should be aware of the fact that the use of term sets with larger size leads to the an increase in number of rules.

Above was supposed so-called a linear fuzzy control rules matrix Hirota, (1995). The linear fuzzy control rules matrix is designed to give linear fuzzy control actions with fuzzy linguistic input variables. The switching surface of this matrix is shown in Figure 3.7.

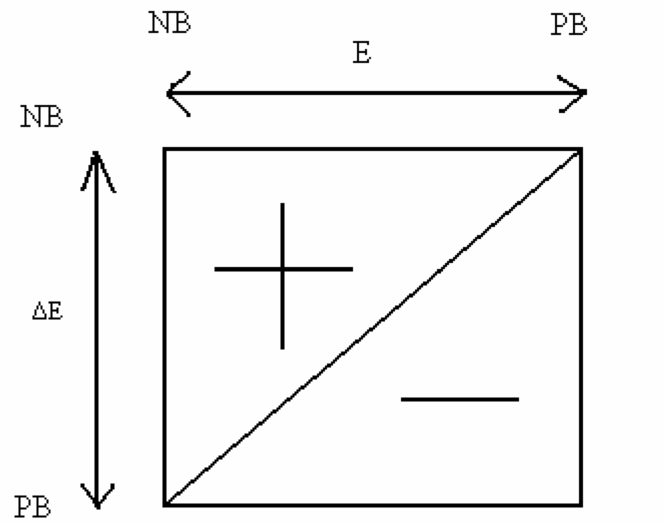


Figure 3.7 Switching surface of the linear fuzzy control rules matrix

If fuzzy control rule matrix is designed to give non-linear fuzzy control actions with respect to different control regions than it is called the aggressive fuzzy control action rule matrix. The regions with large gain control actions are designed to reach the set point faster and the regions with small control actions are to provide damping to the system and to minimize the external disturbances injected into the system.

The aggressive fuzzy control rules are presented in following table

Table 3.2 The aggressive fuzzy rules

ce / e	NB	NM	NS	Z	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	Z
NM	NB	NB	NB	NM	NS	Z	PS
NS	NB	NB	NM	NS	Z	PS	PM
Z	NB	NM	NS	Z	PS	PM	PB
PS	NM	NS	Z	PS	PM	PB	PB
PM	NS	Z	PS	PM	PB	PB	PB
PB	Z	PS	PM	PB	PB	PB	PB

The switching surface of aggressive fuzzy control action rules matrix is shown in Figure 3.8.

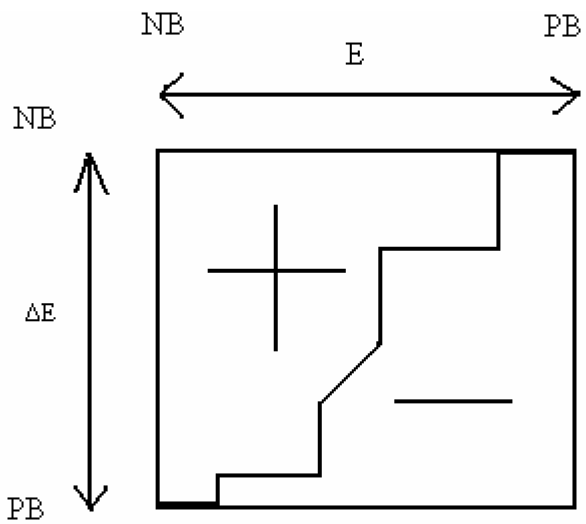


Figure 3.8 Switching surface of the aggressive fuzzy control rules matrix

Choice of Membership Function

Shape of membership function.

For the needs of a quantitative description of the close-loop system behavior, involving the computation of the quantitative control output, we need a quantitative interpretation of the meaning of the linguistic values. For this purpose we use membership functions.

For computational efficiency, efficient use of memory, and performance analysis needs, a uniform representation of the membership functions for input and output is required. This uniform representation can be achieved by employing membership functions with uniform shape and parametric, functional definition.

The most popular choices for the shape of the membership functions include triangular-, trapezoidal-, and bell-shaped function. However, the triangular membership function is mostly used. It can be explained by the simplicity with which a parametric, functional description of the membership function can be obtained, stored with minimal use of memory, and manipulated efficiently, in terms of real-time requirements, by the inference engine.

Cross-point

There is such parameter of membership functions as cross-point. A cross-point between μ_{LX1} and μ_{LX2} is that value x_{cross} in X such that $\mu_{LX1}(x_{cross}) = \mu_{LX2}(x_{cross})$.

Respectively cross-point necessary to say:

First, membership functions have to be such that the cross-point level for every two membership function is greater than zero. This means that every crisp value of input belongs to at least one membership function with degree of membership strictly greater than zero. If this is not the case then none of the rules will fire and consequently, no value for the control output will be computed.

Second, for linear systems up to third order and for symmetrical membership functions there are some “optimal” values for the cross-point level and ratio. If each two adjacent membership functions have cross-point level of 0.5 and cross-point ratio of 1, then this provides for significantly less overshoot, faster rise-time, and less undershoot. The shape of membership functions does not play a significant role, but trapezoidal functions are responsible for a slower rise time.

Left and Right Width

The left (right) width of membership function is the length of the interval from the peak value to that value in X (domain) which is situated to the left (right) of the peak value and whose degree of membership is zero. In process of choosing of membership function are recommended observe the condition width:

The left-width of one membership function is equal to the right-width of adjacent membership function and they are both equal to the length of the interval between the peak values of these two function.

In this case when input changes smoothly from one peak value to another, and after inference and consequent application of the Center-of-Gravity method, output also changes smoothly from one peak value to another.

Choice of Scaling Factors

The use of normalized domains requires a scale transformation which maps the physical values of the process state variables into normalized domain. This is called input normalization. Furthermore, output de-normalization maps the normalized value of the control output variables into their respective physical domain such scale transformations are required both for discrete and continuous domains.

The scaling factor which describes the particular input normalization and output de-normalization play a role similar to that of the gain coefficients in a conventional controller. In other words, they are most important with respect of controller performance and stability related issues, i.e., they are the source of possible instabilities, oscillation problems.

For example, a PI-like FLC can be represented as

$$N_u \cdot \Delta U(k) = F(N_E \cdot E(k), N_{\Delta E} \cdot \Delta E(k)),$$

where N_E, $N_{\Delta E}$, and N_u are the scaling factors for $E, \Delta E, \Delta U$ respectively, and F is a non-linear function representing FLC.

There are basically two major approaches to the determination of the scaling factors: (1) heuristic, and (2) analytic. The first approach has a real-and-error nature. By changing the scaling factors for each one of E and ΔE we are in fact changing the weights given to this particular process state variable. For example, if the process response is slower than the desired one, then we need to increase the effect of error on the process, hence N_E is increased. The iteration is continued until the desired performance values are obtained. The second approach to the derivation of the scaling factors aims at establishing an analytic relationship between the values of the scaling factors and the closed-loop behavior of the controlled process.

Choice of Defuzzification Procedure

There are many defuzzification methods but in practice are used only two of them. These are:

- Center-of-Area/Gravity method
- Height defuzzification.

The Center-of-Gravity method in discrete case (U= $\{U_1,\ldots,U_i\}$) results in

$$U^{\bullet}=\frac{\sum_{i=1}^{l}U_i\cdot\mu_U(U_i)}{\sum_{i=1}^{l}\mu_U(U_i)}=\frac{\sum_{i=1}^{l}U_i\cdot\max_k\mu_{CLU^{(K)}}(U_i)}{\sum_{i=1}^{l}\max_k\mu_{CLU^{(K)}}(U_i)}$$

So this method determines the center of the area below the combined membership function. This operation is computationally rather complex and, therefore, results in quite slow inference system but this method more exactly reflect fuzzy system and gives smooth changing of fuzzy system output in the case of altering of input parameters of the fuzzy system. Also this defuzzification method allows to introduce more significant and more direct able nonlinearity if compare with another defuzzification procedure.

Height defuzzification is a method which instead of using common output membership function uses the individual clipped control outputs. This method takes the peak value of each $CLU^{(K)}$ and builds the weighted (with respect to the height f_K of $CLU^{(K)}$) sum of these peak values. Thus neither the support or shape of $CLU^{(K)}$ play a role in the computational of $U^{\bullet}$. The Height method is both a very simple and very quick method. Let $c^{(k)}$ be the peak value of LU, and f_K is the height of $CLU^{(K)}$. Then the Height defuzzification method in a system of m rules is formally given by

$$U^{\bullet}=\frac{\sum_{k=1}^{m}c^{(k)}\cdot f_k}{\sum_{k=1}^{m}f_k}$$

3.4. Fuzzy Controller Types

There are two basic types of fuzzy logic controller. These are Mamdami's fuzzy controller and Sugeno's fuzzy controller. A brief explanation has been given below for these methods.

3.4.1. Mamdami's Fuzzy Inference Method

Mamdami's fuzzy inference method is the most commonly seen fuzzy methodology. Mamdami's method was among the first control systems built using fuzzy set theory. It was proposed in 1975 by Ebrahim Mamdami as an attempt to control a stream engine and boiler combination by synthesizing a set of linguistic control rules obtained from experienced human operator. Description of fuzzy logic controller in previous part was based on Mamdami's type controller so here I just represent the whole scheme of working of Mamdami's fuzzy logic controller.

Figure 3.9 illustrates the fuzzy control, for example, of a DC motor drive where the two rules are derived from the observed behavior of the plant.

Rule1 states that if the speed loop error (E) is zero (ZE) and the rate of change of speed (CE) is negative small (NS), then the control signal increment (DU) is negative small (NS). The linguistic variables ZE, NS, and DU are defined by symmetrical membership functions, as shown. Graphically solving the problem, the control output Rule1 is DU1. In practice, more than one rule is fired at a time. If Rule2 is fired, it will give output DU2. The effective control output is given by the weighted average of DU1 and DU2. As indicated in Figure 3.10, the output membership function of each rule is given by MIN (minimum) operator whereas the combined fuzzy output is given by SUP (supreme or maximum) operator Bose, B., (1994).

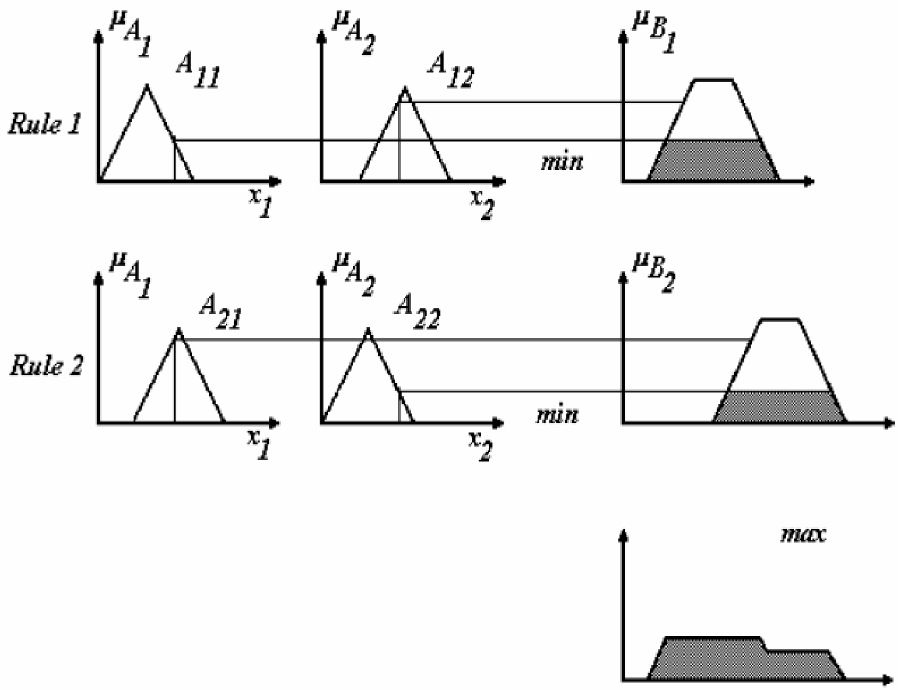


Figure 3.19 Mamdami method

3.4.2. Sugeno's Fuzzy Inference Method

It's possible, and in many function cases much more efficient, to use a single spike as the output membership functions rather than a distributed fuzzy set. This is sometimes known as a singleton output membership function, and it can be thought of as a pre-defuzzified fuzzy set. It enhances the efficiency of the defuzzification process because it greatly simplifies the computation required to find the centroid of a two-dimensional shape. Rather than integrating across a continuously varying two-dimensional shape to find the centroid, we can just find the weighted average of a few data points. Sugeno systems support this kind of behavior Roger Jang & Gulley, (1995).

So-called Sugeno or Takagi-Sugeno-Kang method of fuzzy inference first introduced in 1985. It is similar to the Mamdami method in many respects. In fact the two first parts of the fuzzy inference process, fuzzifying the inputs and applying the fuzzy operator, are exactly the same.

A typical fuzzy rule in a zero-order Sugeno fuzzy model has the form

If x is A and y is B then z=k

where A and B are fuzzy sets in the antecedent, while k is a crisply defined constant in the consequent. When the output of each rule is a constant like this, the similarity with Mamdami's method is striking. The only distinctions are the fact that all output membership functions are singleton skies, and the implication and aggregation methods are fixed and can not be edited. The implication method is simply multiplication, and the aggregation operator just includes all of the singletons.

The more general first-order Sugeno fuzzy model has rules of the form

if X is A and y is B then z= p.x+ q.y +r

where A and B are fuzzy sets in the antecedent, while p, q, and r are all constants. The easiest way to visualize the first-order system is to think of each rule as defining the location of a "moving singleton". That is, the singleton output spikes can walk around the output space, depending on what the input is. This also to make the system notation very compact and efficient. Higher order Sugeno fuzzy models are possible, but they introduce significant complexity with little obvious merit.

Let consider how Sugeno's controller transform input crisp values into output crisp value (Figure 3.10). As shown in Figure 3.10 the premise portion of the rules is identical with that in the rule-base approach, but the consequents are described by equations. Rule1 in the figure can be stated as

IF E is medium AND CE is medium THEN $DU = A_{01} + A_{11} \cdot E + A_{21} \cdot CE$ and Rule2 be stated as IF E is zero AND CE is medium THEN $DU = A_{02} + A_{12} \cdot E + A_{22} \cdot CE$

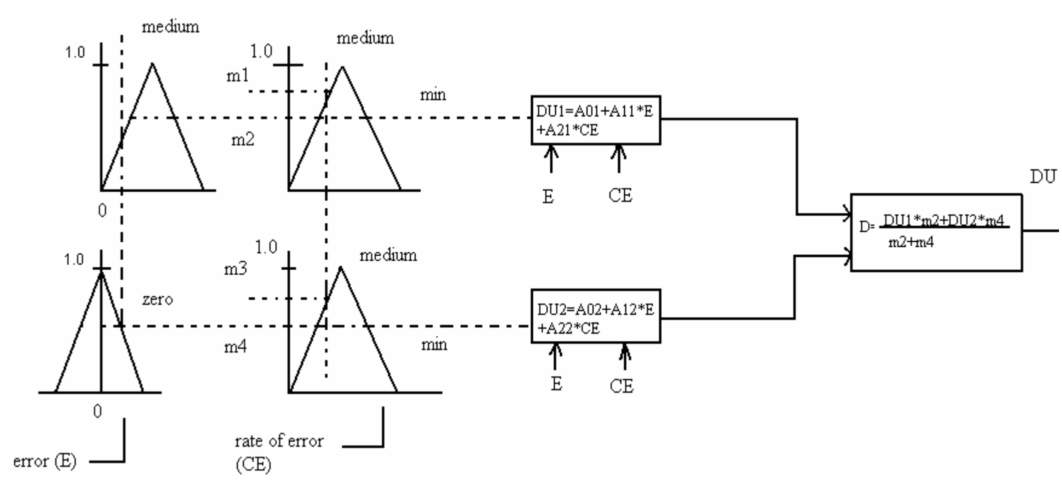


Figure 3.10 Sugeno method

The consequences are linear functions of C and CE and parameters A_{ij} are constant coefficients. The A_{ij} can be determined by multiregression linear analysis, and then fine-tuned by observation or simulation. The linear equation outputs are then defuzzified, i.e., weighted average of the consequence is evaluated by the respective membership values to determine the crisp output.

Because of the linear dependence of the each rule on the system's input variables, the Sugeno method is ideal for acting as an interpolative supervisor of multiple linear controllers that apply in different operating conditions of a dynamic nonlinear system. A Sugeno fuzzy inference system is extremely well suited to the task of smoothly interpolating linear gains across the input space; it's a natural and efficient gain scheduler. A Sugeno system is also suited for modeling nonlinear systems by interpolating multiple linear models.

Thus, Sugeno rule can be more expressive than a rule in a Mamdami system. Because it is more compact and computationally efficient representation than a Mamdami system, the Sugeno system lends itself to adaptive techniques.

3.4.3. Comparison of Mamdami and Sugeno Methods

From consideration of these two methods it is possible to make following conclusion.

Advantages of Mamdami's method:

- More intuitive
- Widespread acceptance
- Better suited to human input
- More nonlinear

Advantages of Sugeno's method:

- Computationally efficient
- Works well with linear techniques (e.g. PID control, etc.)
- Guaranteed continuity of the output surface
- Better suited to mathematical analysis

3.5. Applications of Fuzzy Control in Power Electronics and Drives

The fuzzy control has found wide application in field of power electronics, because of the nonlinear structure of the circuits.

From analysis of literature about fuzzy logic it is possible to distinguish following main directions of the fuzzy logic using:

- Approximation of continuous function
- Motion control
- The estimation of system parameters
- Optimization of system performance

3.5.1. Approximation of Continuous Function

In many situations fuzzy logic constitutes an efficient way to represent a non-analytic mapping, such that an output value y can be inferred from a given input x. In order to construct the fuzzy model, the input/output pattern must be known. Membership functions can then be assigned to ranges of values for both input and output variables, such that proper covering of their universe of discourse is obtained. A certain amount of overlap among the membership functions is desirable from a robustness point of view.

A set of fuzzy rules in the form of "IF x is A THEN y is B" (A and B are fuzzy sets in universe of discourse of x and y, respectively) is obtained form expert knowledge, or generated by neural or statistical procedures. It has been demonstrated Kosko, (1992) that fuzzy system can approximate any continuous function to any degree of accuracy. It can be seen from Figure 3.11 that the fuzzy system approximation the unknown function with patches.

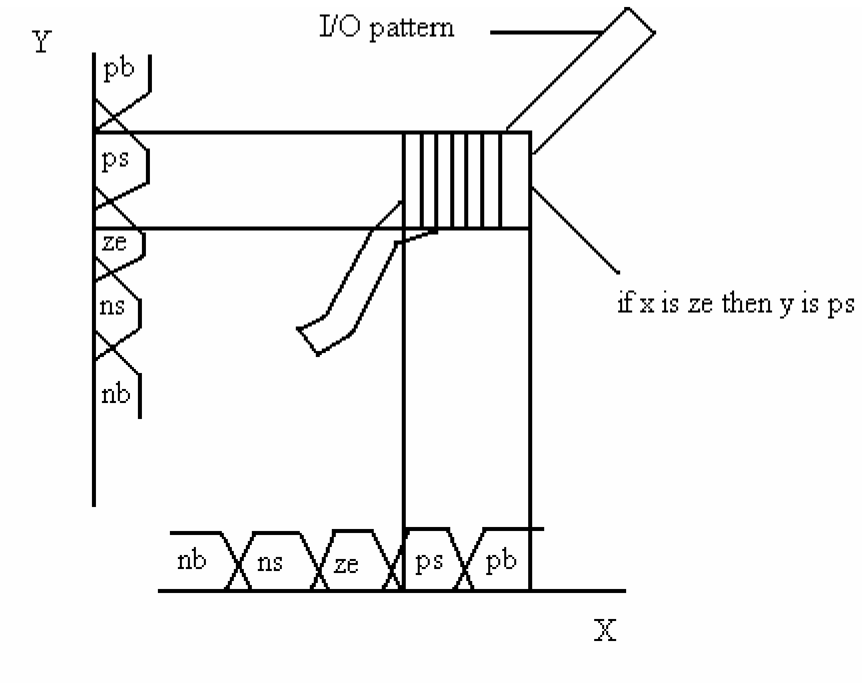


Figure 3.11 Fuzzy system representations

Higher accuracy can be obtained using a large number of membership functions but at expenses of increased computational cost.

3.5.2. Motion Control

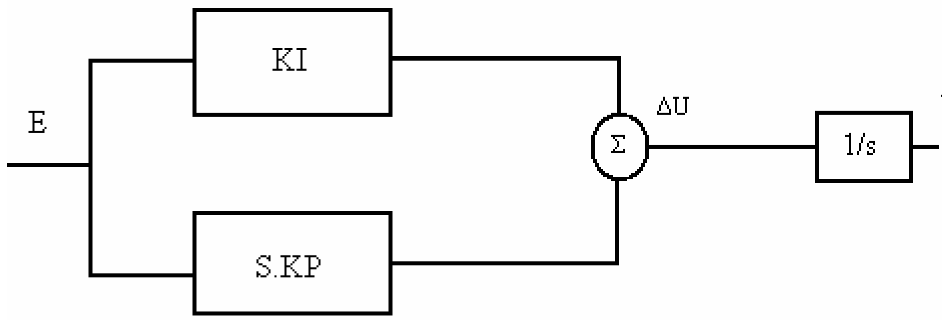
The large majority of applications of FLC lies in this category. Fuzzy logic controller is applied in closed-loop control of a drive system. With nonlinearity, parameter variation, and load disturbance effects, it can provide faster and more robust control then usual conventional PI controller.

The drive system may be based on a DC or Ac machine. Since vector-controlled AC drive has identical dynamics models, the same fuzzy control principle is valid in either case. So consider, for example, DC drive system. In a phase-controlled converter DC drive using a separately excited DC motor is following possibilities of using FLC:

- In speed loop and current loop instead conventional PI controller
- In position loop instead PD controller
- FLC in phase controlled converter for linearization characteristic at discontinuous condition.

If analyze literature about applications of FLC then it can be noticed that the common methodology, using by authors for design of FLC is the same. For example, we replace conventional PI controller in speed loop by fuzzy logic controller.

The first step is definition of linguistic variables. If we use such structure of PI controller as then input variables, similar usual conventional controller, are error (E) and change of error (CE) and output variable is control increment (DU). But in the case of PD controller input variables also error and change of error. Thus, PI- and PD-like fuzzy controllers have same structure and differ only by output variable (control increment and control action, respectively).



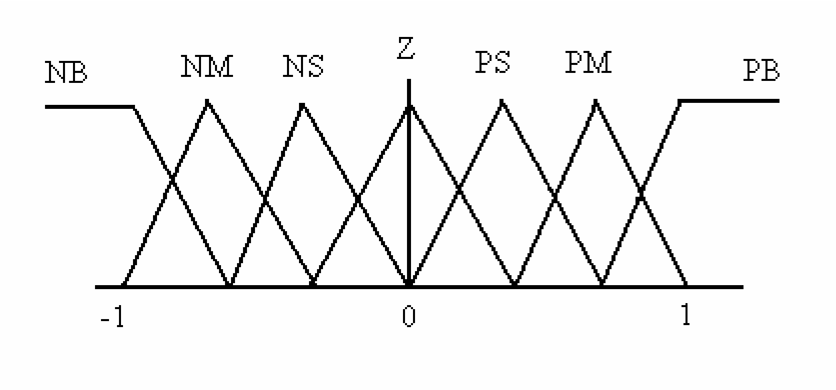
Second step is choice of term set and universe of discourse for each linguistic variable. Many authors propose to use five-seven fuzzy sets for input variables and seven fuzzy sets for output variable. These are:

NB-negative big,
NM-negative medium,
NS-negative small,
Z-zero,
PS-positive small,
PM-positive medium,
PB-positive big.

And these fuzzy sets are defined on normalized universe of discourse [-1, 1] that gives afterwards for us easy way to tune our system by changing scaling factor for each variable.

Third step is choice of membership function form. Are proposed to use symmetrical triangular membership functions with 50% overlap.

So for fuzzy speed controller on initial stage we get following fuzzy sets for each linguistic variable.



Fourth step is formulating of rule base. Usually this is most difficult step of fuzzy controller design because the step requires expert knowledge about system. But in the case of system in which bigger input value corresponds bigger output value and the purpose of the control is to eliminate steady-state error between desire and actual values of the system output we can formalize the procedure of rule base creating in table form.

- First, place fuzzy sets of variable E from NB to PB from up to down
- Second, place fuzzy sets of variable CE from NB to PB from left to right
- Now place fuzzy sets of control incremental by following principle

ce e	NB	NM	NS	Z	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	Z
NM	NB	NB	NB	NM	NS	Z	PS
NS	NB	NB	NM	NS	Z	PS	PM
Z	NB	NM	NS	Z	PS	PM	PB
PS	NM	NS	Z	PS	PM	PB	PB
PM	NS	Z	PS	PM	PB	PB	PB
PB	Z	PS	PM	PB	PB	PB	PB

Build "Z"-diagonal from left down corner to upper right corner. After that in each column place fuzzy sets below Z consequently from PS to PB and above Z consequently from NS to NB. And by such method rule base table will be created.

- And on forth step has chosen the defuzzification procedure finish the fuzzy logic design.

That fuzzy controller gives desired value of current for current loop. In current loop it is also possible use the same fuzzy controller like in speed loop. And if fuzzy controllers well-tuned results of simulations show that control with FLC gives faster response of system on the change of reference value without overshoot and oscillations if compare with conventional PI-controller. And also system becomes more robust with inertia variation and load torque disturbance Bolognani, (1992).

However, FLC cannot adapt themselves to changes in their environment or in operating conditions, so some mechanism of adaptation is required to maintain desired characteristic of the control when operating conditions change over a wide range.

Classical adaptive control systems are based on mathematical modeling so their design and implementation are usually complex because of the computationally intensive algorithms used. The adaptation of fuzzy logic control system can be achieved using the same schemes developed for classical adaptive systems. However, the design and implementation can be made simpler by using fuzzy reasoning in the adaptation mechanism Le-Huy & Viarouge, (1995).

Figure 3.12 shows block diagram illustrating the model adaptive principle applied to a fuzzy logic control.

Two important functional blocks (reference model and fuzzy logic adaptation mechanism) have been added to the conventional fuzzy logic controller to form a second control loop.

The reference model is used to specify the desired performance that satisfies design criteria such as rise time and overshoot. It can be any type of dynamic system but a first-or second-order model is usually used. The same reference input r is applied to both the reference model and the fuzzy controller. The system actual output is compared with the reference model output. The result error (em) and the change of error (cem) are applied to the fuzzy logic adaptation mechanism (FLAM) that will modify the FLC characteristics to force the system to behave like the reference model.

Two schemes can be used: parameter adaptation and signal adaptation.

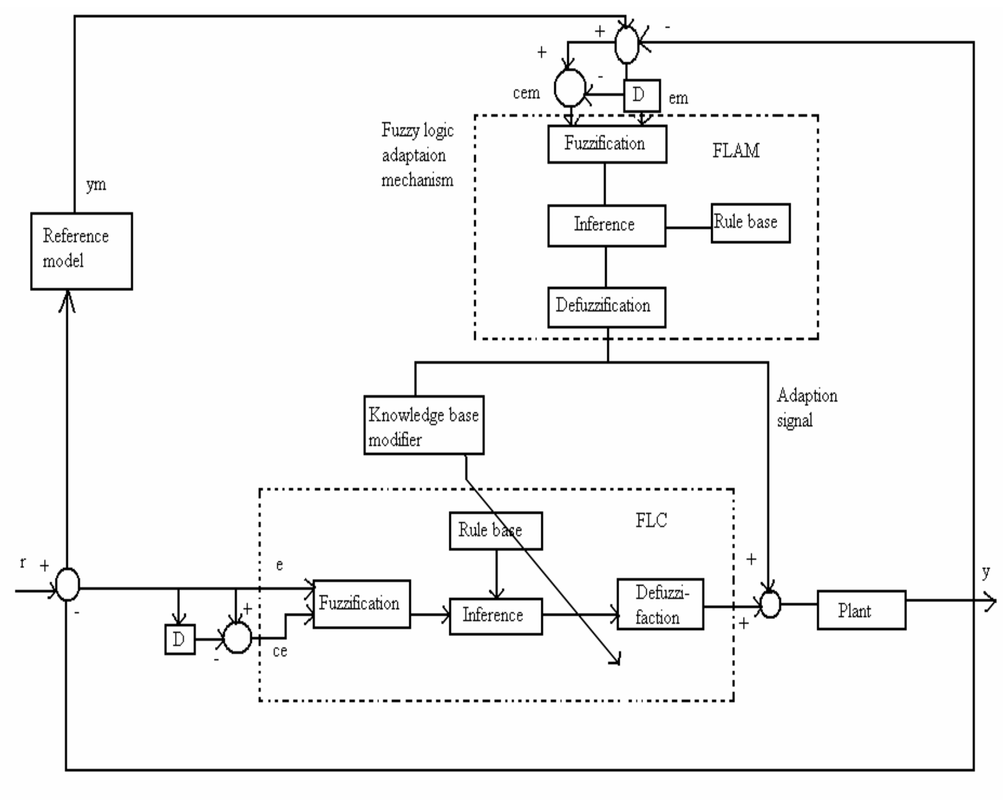


Figure 3.12 Model reference adaptive control system

In parameter adaptation scheme, the knowledge base of the fuzzy logic controller is modified by changing the membership functions, the characteristics of rules, or topology of fuzzy sets. The FLAM output signal is used as input signal for a knowledge base modifier, the function of which is to modify the knowledge base of

FLC to produce the required change in the plant input u. The knowledge base is updated at regular intervals the modifications can be stored in system memory. With this scheme the FLC becomes self-organizing, that is it can learn to work better with experience. However, its implementation may be complex because the manipulation of large quantities of information in the knowledge base is necessary.

In the signal adaptation scheme, the adaptation mechanism produces an auxiliary control signal that is added to the FLC output to compensate for the deviation of the performance due to changes in operating or load conditions. This adaptation approach does not provide learning capability but its implementation is much simpler since the knowledge base is not modified.

The internal structure of the FLAM is identical to that of direct FLC: fuzzification, rule execution, and defuzzification. As in the case of FLC, FLAM control rules are formulated based on the knowledge of the drive behavior and common sense.

The simulation results have confirmed the efficiency of the fuzzy adaptive scheme in the case of changing load moment of inertia and load torque in large range.

3.5.3. Estimation of the System Parameters

The ability of fuzzy logic for continuous approximation of human knowledge gives effective way for estimation of system's parameters to another method, for example, in the case of waveform estimation Bose, (1994).

Power electronic converters characteristic ally generate distorted voltage and current waveforms. Electronic instrumentation techniques are extensively used to process these waves and determine the quantities, such as total rms value, fundamental rms value, active power, reactive power, displacement factor, and power factor.

Often, mathematical model (if available) and look-up table's methods are also used for the estimation. The computation-intensive approach has the disadvantage that the response is slow because integration and averaging processes are involved. The look-up tables solve this demerit, but for good accuracy, the size of the table (one or multi-dimensional) should be large or interpolative calculation becomes necessary.

Fuzzy logic waveform estimation has the advantages of the fast response, multi outputs from a single premise of a rule, and immunity of noise that drift from the sensors.

For example, we can use rule-based estimation technique for line current of a three-phase diode rectifier Bose, (1994). The pattern of the current wave is characterized by the width and height parameters, and estimate is dependent on their values. As usually each linguistic variable define by fuzzy sets, moreover number of fuzzy sets for each variable is different and membership functions are asymmetrical and non-identical for each variable because each output (rms current, fundamental current, or displacement factor) is different and has different degree of nonlinearity. Using our knowledge about dependencies between height, width of current wave and current rms, fundamental rms, displacement factor we define rule base and form for membership function. The typical rule will be IF height is small AND width is big THEN current rms is not large, fundamental rms is big, and displacement factor is medium.

And after using of the fuzzy inference mechanism and defuzzification procedure we obtain estimated values of output parameters. For improved accuracy, the formulation of the rule base and membership functions and their iteration are based on simulations results.

3.5.4. The Optimization of System's Performance

Fuzzy logic can be used to enhance the performance of the system. For example, induction motor drives are normally operated at rated flux condition to give best transient response. However, at light-load condition, this gives excessive core loss, impairing efficiency of the drive. The flux can be programmed at light-load steady state in order to improve efficiency of the drive. Consider the motor operation initially at rated and steady state with the load, torque, and speed. The rotor flux is decremented in steps by reducing the magnetizing component of stator current i_{ds}. This result in an increase of the torque component of current i_{qs} (normally by speed loop), so that the developed torque remains the same. As the core loss decreases with the decrease of flux, the copper loss increases, but the system (converter and machine) loss decreases improving the overall efficiency. This is reflected in the decreases of dc link power. The search is continued until the system settles down at minimum input power.

For this purpose we can use a fuzzy logic approach Bose, (1995). The fuzzy logic controller has the advantage that it adaptively decrements the step size of the excitation current so that fast convergence is attained. The steps are again programmable and depend on the operating point of the torque-speed plane. The fuzzy efficiency controller detects the steady state condition when the speed loop error Δw_r approaches zero, and then it invokes the efficiency optimization control. A typical fuzzy rule can be given as IF the power increment (ΔP_d) is negative medium (NM) AND the last $i_{ds}(L - i_{ds})$ is negative (N) THEN the excitation increment (Δi_{ds}) is negative medium (NM).

This procedure continues until the input power will be minimum. Or another example, fuzzy controller can be successfully applied in phase-controlled converter for linearization of converter characteristic, under discontinuous condition.

For continuous current condition, the converter output voltage varies linearly with cosine of firing angle α. Linearity is lost when the converter enters the discontinuous condition region, where output voltage becomes a function of both α and armature current. Converter linearization can be reestablished by proper addition of a compensating $\Delta\alpha$.

While closed-form equation $\Delta\alpha$ does not exists, the fuzzy relation can be obtained from the numerical solution, for limited number of α and armature current values. The inherent interpolation capability of FLC assures proper computation of $\Delta\alpha$ for any given α and I_a inputs, thus rendering a linear overall converter characteristic. In this way, faster current loop response can be obtained for all modes of operation.

So, in the case of linearization of converter's characteristics under discontinuous condition we have not mathematical model for this purpose but fuzzy approach gives us convenient way to decide this problem.

Thus, as can be seen from above review fuzzy logic approach expands now very rapidly and number of fuzzy logic applications constantly grows.

3.6. Adaptive Fuzzy Control

The fuzzy logic controller is nonlinear and so they can be designed to cope with a certain amount of process nonlinearity. However, such design is difficult, especially if the controller must cope with nonlinearity over a significant portion of the operating range of the process. Also, the rules of the fuzzy logic controller do not, in general, contain a temporal component, so they cannot cope with process changes over time. So there is a need for adaptive FLC as well.

Adaptive controllers generally contain two extra components on top of the standard controller itself. The first is a "process monitor" that detects changes in the process characteristics. It is usually in one of two forms.

- a performance measure that assesses how well the controller is controlling, or
- a parameter estimator that constantly updates a model of the process.

The second component is the adaptation mechanism itself. It uses information passed to it by the process monitor to update the controller parameters and so adapts the controller to the changing process characteristics.

Fuzzy logic controller contains a number of parameter sets that can be altered to modify the controller performance. These are:

- the scaling factors for each variable,
- the fuzzy set representing the meaning of linguistic values,
- the if-then rules.

The adaptation mechanism must modify the controller parameters to improve the controller performance on the basis of the output from the process monitor. Adaptation mechanism for FLC can be classified according to which parameters are adjusted. Parameters that can be adjusted include the scaling factors with which controller input and output values are mapped onto the universe of discourse of the fuzzy set definitions.

The set definitions will often be defined on a normalized universe from, say, -1.0 to +1.0, as illustrated in Figure 3.13.

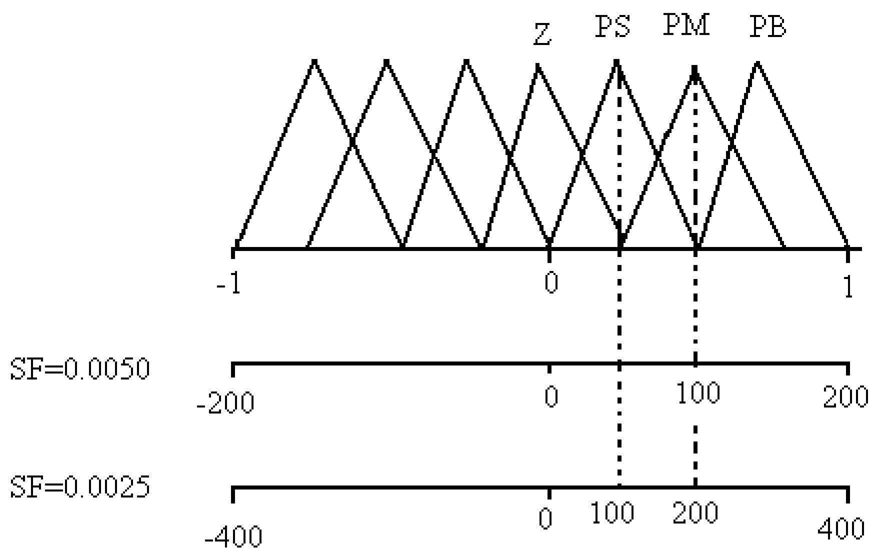


Figure 3.13 The effect of altering a scaling factor

The real values of the input variables may actually range from, say, -200 to +200, and so need to be scaled. If, for example, the input value is multiplied by a scaling factor of 0.005, the input is mapped to the universe of discourse as shown by the middle scale in Figure 3.13. In this case an input value of 100 is classified as positive medium. Altering the scaling factor changes the classification of an input value. For example, with a scaling factor of 0.0025 a value of 100 is now classified as positive small, as shown by the bottom scale of Figure 3.13. This reduces the sensitivity of the controller to the input, and so reduces the controller gain. In this way, altering scaling factors is similar to gain tuning in standard PID-controllers.

Another gain tuning mechanism is to alter the shapes of the fuzzy sets. An example, the sets are altered to increase the sensivity of the controller to small values of the input is shown in Figure 3.14.

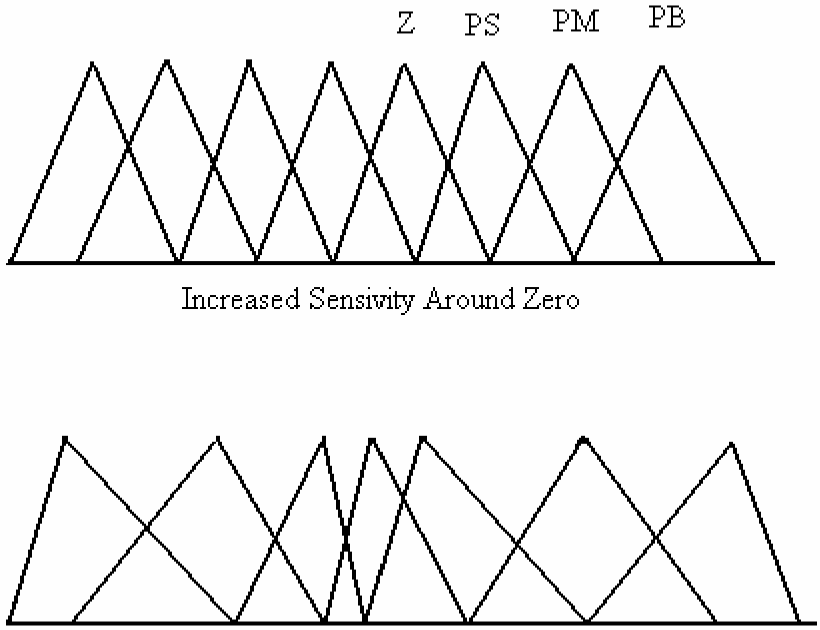


Figure 3.14 Adapting fuzzy set definition

Whereas altering the scaling factors alters the gain uniformly across the entire input universe, changing the shapes of specific fuzzy sets allows the gain to be modified within specific regions of the universe. Adaptive controllers that adjust the fuzzy set definitions or scaling factors are called self-turning controllers.

A third set of parameters that can be altered are the if-then rules themselves. For example, a rule may be changed from

IF error is positive small AND changes of error positive medium THEN control increment is positive medium.

IF error is positive small AND change of error positive medium THEN control increment is positive big.

to adapt changing operating conditions. Adaptive controllers that alter the rules are called self-organizing controllers.

At present with the knowledge that was considered above in this chapter it is possible to design and research fuzzy logic controller for any kind application.

CHAPTER FOUR
STEP-UP CONVERTER DESIGN

4. DC-DC Converter Design

4.1. Converter Design Procedure

The manual implementation of an integrated design procedure can be tedious even for simple systems due to the number of variables involved, non-linearity of modules, discrete and continuous nature of the design variables, and so on. Therefore, a computer algorithm is developed to automate the procedure of integrated design, and is referred to as the design program in this paper Sridhar,K., & Lang, J.H., & Umans,D., (1995).

4.1.1. Power Switch Design

BJT, MOSFET, or IGBT can be used as a switching components in boost converter design. However, this switching components have some advantages and disadvantages as to each other. In below, some features are given and these parameters are given Figure 4.1.

- BJTs- greater capacity, low ON state loss
- MOSFETs- fast switching, voltage driven
- IGBTs- combined modules, powerful and expensive

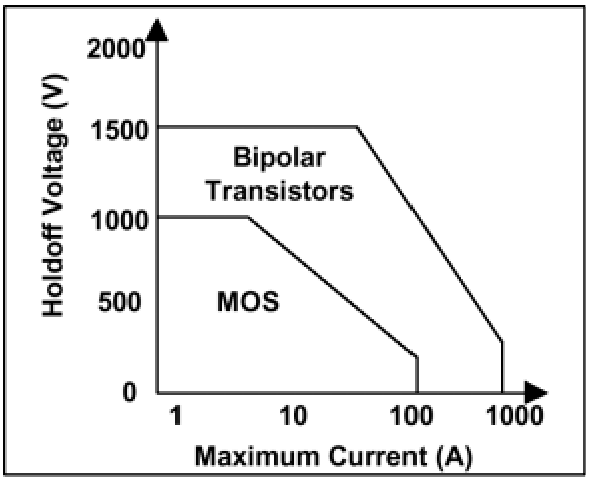


Figure 4.1 The graphic of switching voltage and current

Illustrate the design procedures with a design example:

Design requirement:

A 240-watt DC/DC boost converter with $V_{in} = 24V$ and $V_{out} = 48V$.

1. Current and voltage rating requirements

Peak transistor current equals to

$$I_{in} = P/V_{in} = 240W/24V = 10A$$

Voltage rating requirements

$$V_{Tmax} = V_{out} + V_{F(diode)} = 48 + 0.7V = 48.7V$$

2. Device selection based on the requirements

Design:

Candidate I: BJT 2N6547

$I_C = 15A \rangle 10A$ and $V_{CE} = 400V \rangle 48V$

Candidate II: power MOSFET IRF2Z4N

$I_D = 15A \rangle 10A$ and $V_{DS} = 55V \rangle 48V$

3. Selection

Power MOSFET IRF2Z4N

4. Switching loss

$$P_{SW\_loss} = \frac{W_{loss}}{T} = \frac{1}{T}\left(W_{loss\_ON} + W_{loss\_OFF}\right) = \frac{V_{ds} I_d}{2T}\left(t_{ON} + t_{OFF}\right)$$

$$= \frac{48 \times 10}{2 \times 1/20 \times 10^3}(60 + 100) \times 10^9 = 0.768W$$

5. ON state loss

* ON state time

$$t_1 = \frac{1}{f}\left(\frac{V_{out} + V_F - V_{in}}{V_{out}}\right) = \frac{1}{20 \times 10^3}\left(\frac{48 + 0.7 - 24}{48}\right)\sec = 25.73\mu\ s$$

* ON state loss

$$P_{ON\_loss} = \frac{W_{ON\_loss}}{T} = \frac{1}{T}\left(I_D{}^2 r_{DS(ON)} t_1\right)$$

$$= \frac{1}{1/20 \times 10^3}\left(15^2 \times 0.075 \times 25.73 \times 10^{-6}\right) = 8.684W$$

6. Overall loss

$$P_{loss} = P_{SW\_loss} + P_{ON\_loss} = 9.452W \langle P_D = 45W$$

Calculation of junction to sink temperature difference

$$\Delta T_{JC}(t) = P_{loss} Z_{\theta JC(50\%)}(t) R_{\theta JC}$$

$$\therefore \Delta T_{js} = R_{\theta js} \times P_{loss} = 3.3^{\circ} C/W \times 9.452W = 31.2^{\circ} C \langle T_{jmax} = 175^{\circ}$$

Figure 4.2 is showed junction to sink temperature with MOSFET IRFZ44N.

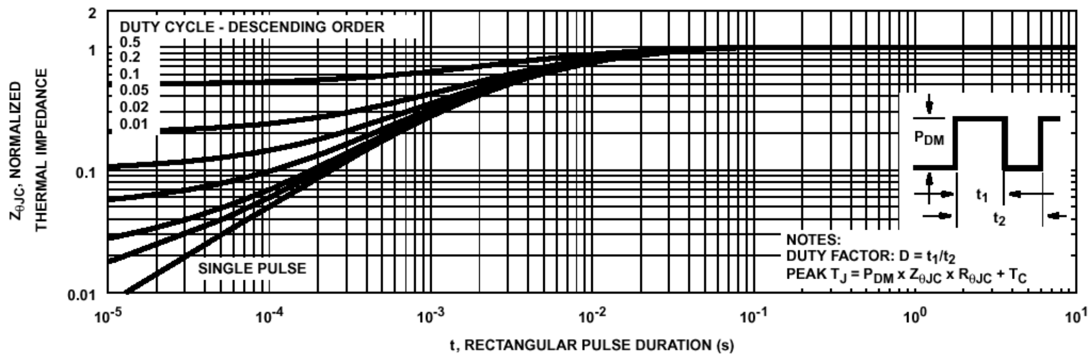


Figure 4.2 The graphic of junction to sink temperature

4.1.2. Calculation of Inductance

In the previous section choosing of switching component was done, in this section, inductance calculation of boost converter will be done.

For continuous mode $(\Delta I_L \langle 2I_{in})$

$$t_1 = \left(\frac{1}{f}\right) \cdot \left(\frac{V_{out} + V_F - V_{in}}{V_{out}}\right)$$

$\Delta I_L = \frac{1}{L} \cdot V_{in} \cdot t_1$ and

$$I_{max} = I_{in} + \frac{1}{2}\Delta I_L$$

For discontinuous mode $(\Delta I_L \rangle 2I_{in})$

$$t_1 = \sqrt{2I_{out} \cdot L \cdot \left(\frac{V_{out} + V_F - V_{in}}{f \cdot V_{in}^{\ 2}}\right)}$$

$$t_2 = t_1 \cdot \left(\frac{V_{out} + V_F}{V_{out} + V_F - V_{in}}\right) \text{ and}$$

$$I_{max} = \frac{1}{L} \cdot V_{in} \cdot t_1$$

Assuming 15% current ripple:

$$\Delta I_L = 0.15 I_L = 0.15 I_{in} = 1.5\text{A}$$

$$L = \frac{1}{f}\left(V_{out} + V_F - V_{in}\right)\left(\frac{V_{out} + V_F}{V_{in}}\right)\frac{1}{\Delta I_L}$$

$$= \frac{1}{20\times10^3} \times (48 + 0.7 - 24) \times \frac{48 + 0.7}{24} \times \frac{1}{0.15\times10}\text{H}$$

$$= 406\,\mu\text{H}$$

Peak current:

$$I_{max} = I_{in} + \frac{1}{2}\Delta I_L = I_{in} + \frac{1}{2}\left(\frac{1}{f}\right)\left(V_{out} + V_F - V_{in}\right)\left(\frac{V_{in}}{V_{out} + V_F}\right)\frac{1}{L}$$

$$= 10 + \frac{1}{2}\frac{1}{20\times10^3}(48 + 0.7 - 24)\left(\frac{24}{48 + 0.7}\right)\frac{1}{650\times10^{-6}}\text{A}$$

$$= 10 + 0.468\text{A} = 10.468\text{A}$$

$$10.468\text{A} \langle 15\text{A} = I_{Dmax}$$

After inductance value was calculated, copper wire, which has 2.8mm^2 diameter, is wrapped up to the toroid by hand, in order to claim calculated inductance value. And this inductance value is measured with the specific mechanism. This measurement result are given in the below Figure 4.3.

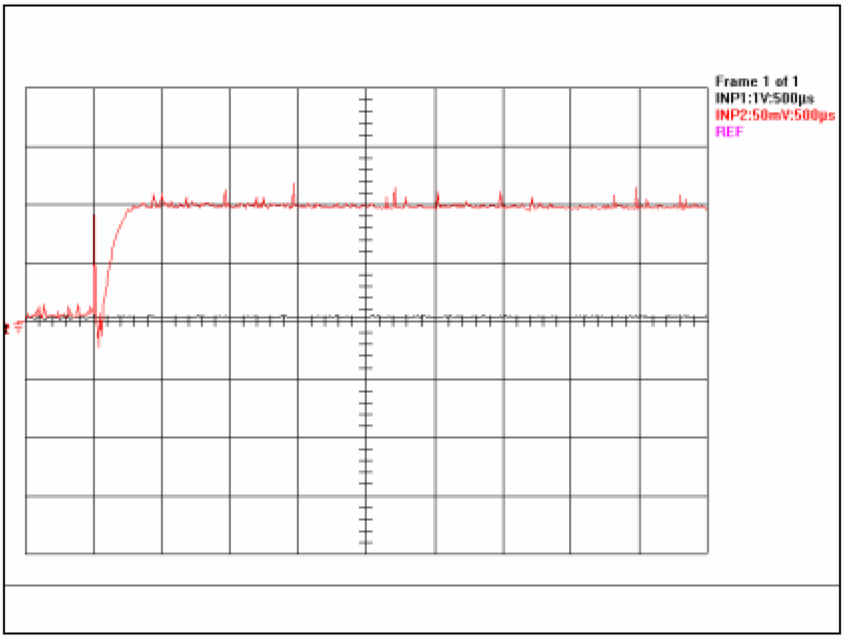


Figure 4.3 The graphic of inductance measurement

4.1.3. Calculation of Capacitance

In this section capacitance value is calculated. Peak current of diode, rms value of diode current, and rms value of capacitance current is taken into consideration in the calculation of capacitance. Figure 4.4 is showed a conventional boost converter circuit.

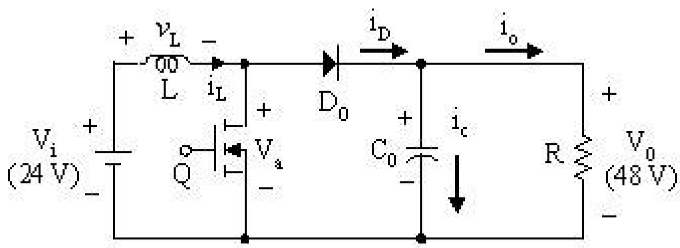


Figure 4.4 A conventional boost converter

Assume the diode current i_D is a square wave form is given below Figure 4.5.

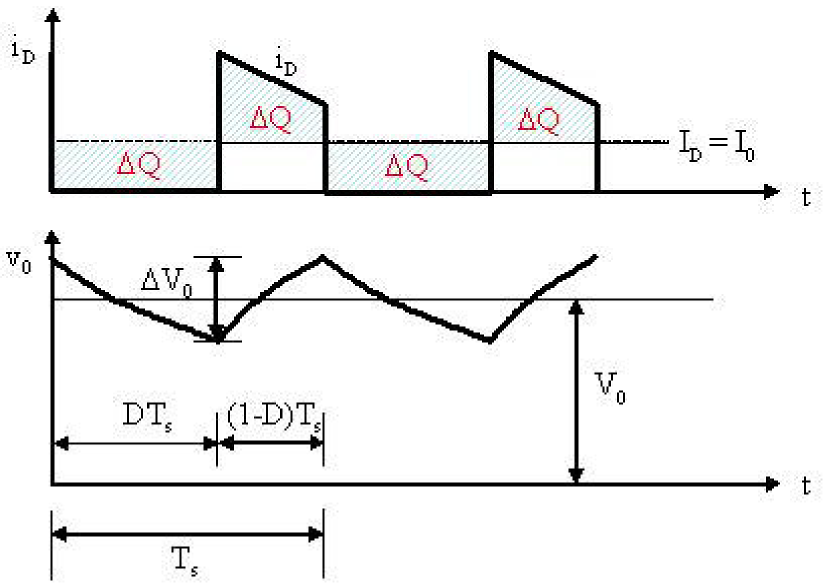


Figure 4.5 Output voltage ripple

- Peak diode current:

$$I_{D,peak} = \frac{I_0}{D} = \frac{I_0}{0.5} = 10A$$

$$\left(\text{Where } D = 0.5, I_0 = P/V_0 = 240/48 = 5A\right)$$

- RMS diode current:

$$I_{D,rms} = I_{D,peak} \cdot \sqrt{D} = 10 \cdot \sqrt{0.5} = 7.07A$$

- RMS capacitor current:

$$I_{C,rms} = \sqrt{{I_{D,rms}}^2 - {I_0}^2} = \sqrt{7.07^2 - 5^2} = 5A$$

- Output voltage ripple:

$$\Delta V_0 = \frac{\Delta Q}{C} = \frac{I_{C,rms} D T_s}{C}$$

- Capacitance:

$$\therefore C = \frac{\Delta Q}{\Delta V_0} = \frac{I_{C,rms} D T_s}{\Delta V_0} = \frac{5 \cdot 0.5}{48 \times 10^{-3} \times 20 \times 10^3} = 2600 \mu F$$

$$\text{Where, } D = 0.5,\ T_s = 1/\left(20 \times 10^3\right)\sec,\ I_{C,rms} = 5A,\ \Delta V_0 = 48mV$$

A finally, fast diode, which has 0.5ns switching time, is used in circuit design.

CHAPTER FIVE
SIMULATION RESULTS AND EXPERIMENTAL RESULT

5. Design of Fuzzy Logic Controller

In this chapter, in simulation program step-up converter, which is controlled by fuzzy logic, has been explained. Control of step-up converter, which is made by Matlab simulation program, is described and results are got. The block diagram of the system is given in Figure 5.1.

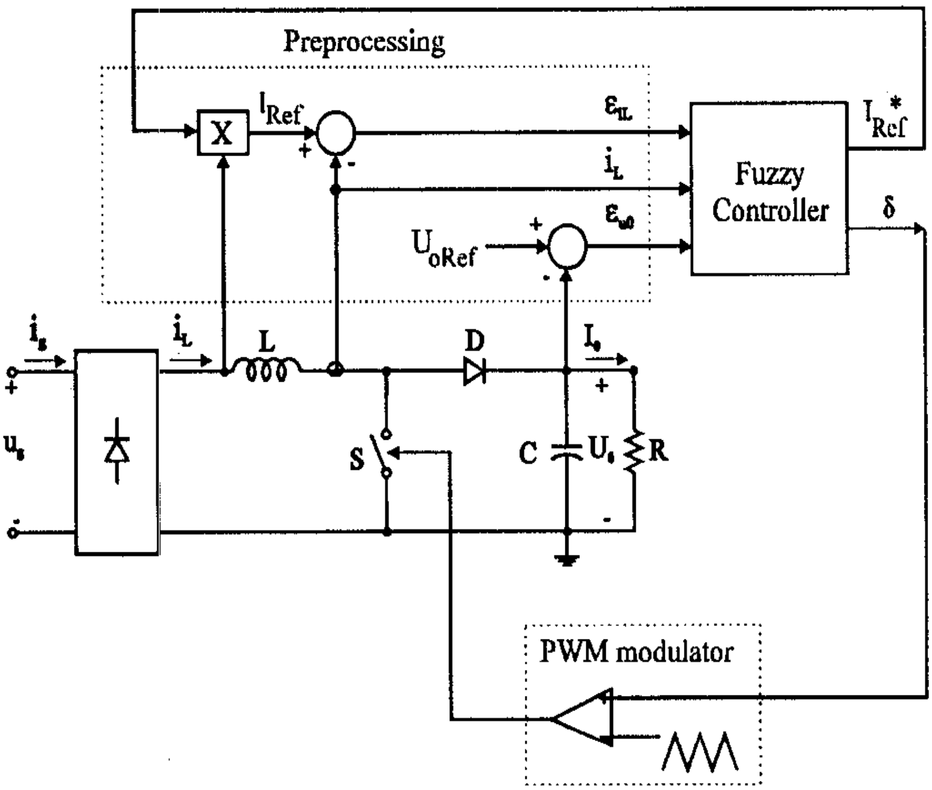


Figure 5.1 Block diagram of step-up converter with fuzzy control

The derivation of the fuzzy control rules is heuristic in nature and based on the following criteria:

1. When the output of the converter is far from the set point, the change of duty cycle must be large so as to bring the output to the set point quickly.
2. When the output of the converter is approaching the set point, a small change of duty cycle is necessary.
3. When the output of the converter is near the set point and is approaching it rapidly, the duty cycle must be kept constant so as to prevent overshoot.
4. When the set point is reached and the output is still changing, the duty cycle must be changed a little bit to prevent the output from moving away.
5. When the set point is reached and the output is steady, the duty cycle remains unchanged.
6. When the output is above the set point, the sign of the change of duty cycle must be negative, and vice versa.

As already pointed out, the current reference in this work has been attained by a fuzzy PI regulator. The general guidelines to design such a regulator can be found in (11). As far as the scaling factors (k_{eu} and k_{ceu} for the output voltage and its error and $k_{\Delta Iref}$ for the increment of the current reference) are concerned a procedure similar to the previously described for the other fuzzy regulator can be employed.

5.1. Simulation Results of Step-up Converter

The system, which is given above, is analyzed by using Matlab program. In this system, a PWM modulator, PI controller and Fuzzy control look-up table are used. This simulation program is given in the below.

In our system, the PWM switching signal is 20 kHz. The component, which has been used, has been described in chapter four. The values, which has been found, of components have been is replaced in the simulation program. The load of converter is only resistive load. The circuit is increases DC voltage from 24 V to 48V.

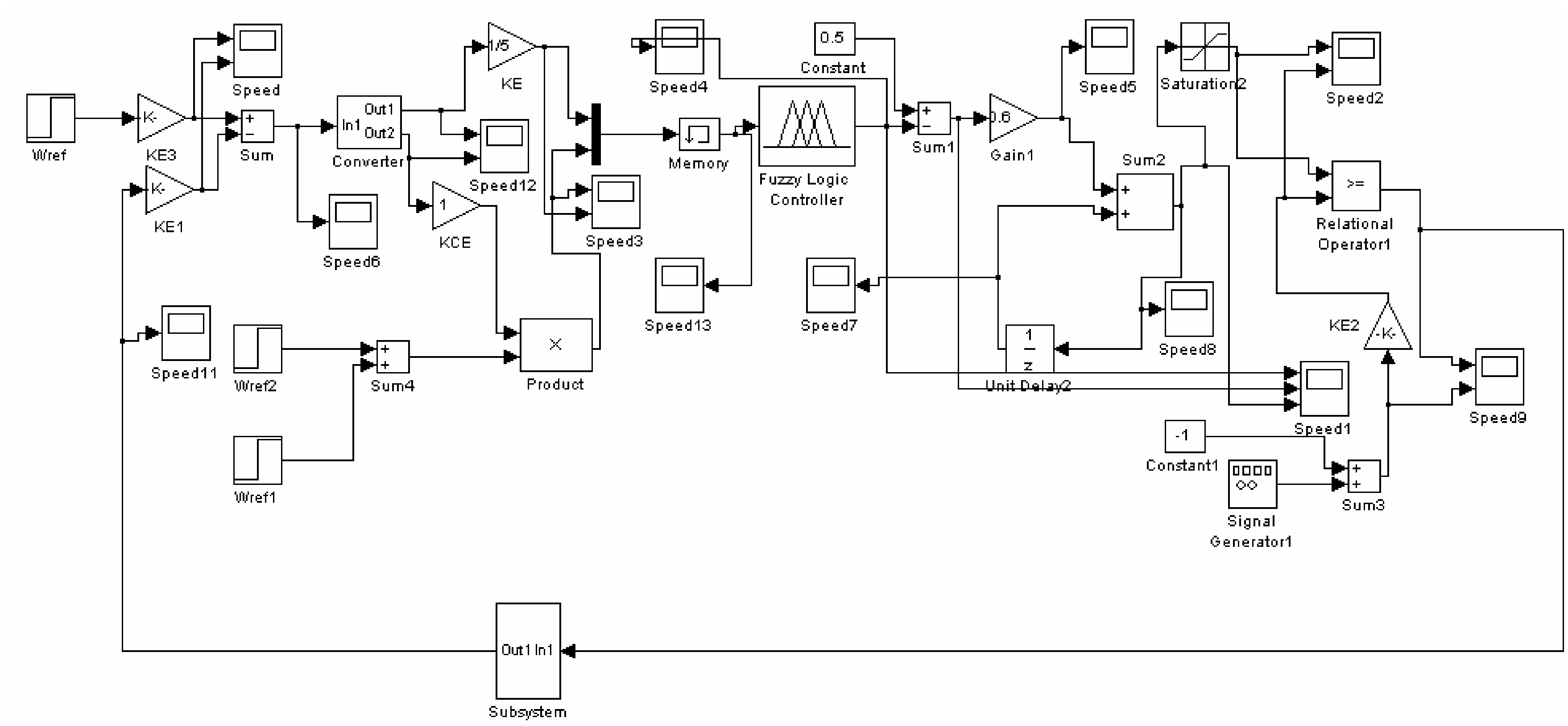


Figure 5.2 The Matlab simulation program scheme

In our system, we have used MOSFET as a switching component. The PWM signal, which has been applied to MOSFET, is shown in the below.

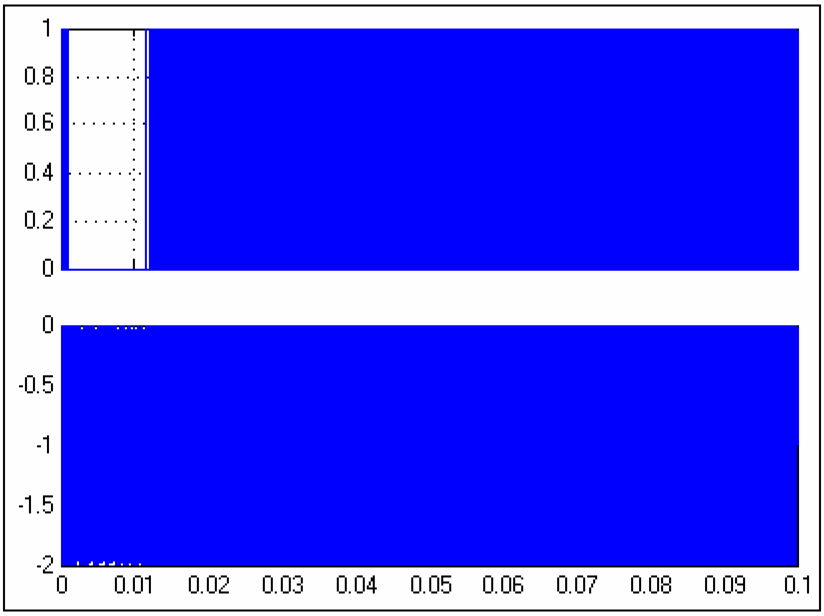


Figure 5.3 The signal of PWM

The output values of A/D converter are given in the below graphics. The error and change of error signals of fuzzy logic controller are got as using this output values. These are shown in the below.

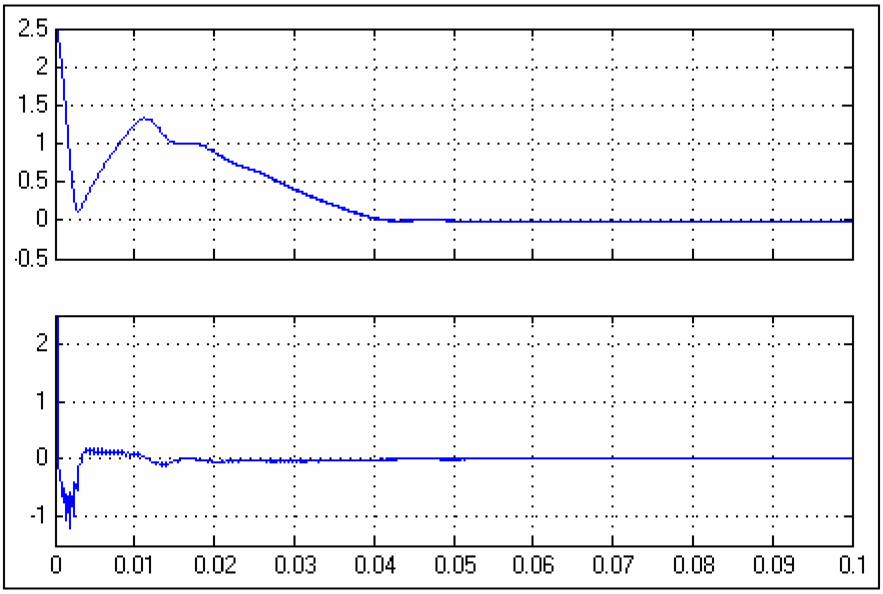


Figure 5.4 The output of A/D converter

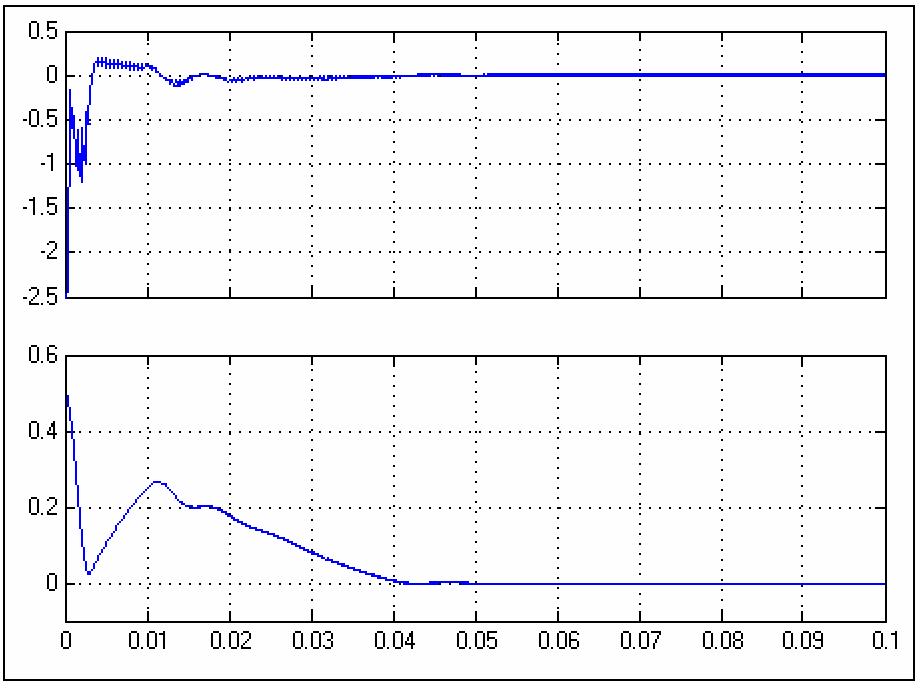


Figure 5.5 The graphics of change of error and error signal

Finally, the results, which taken from outputs of the step-up converter, has been seen. The initial value of step-up converter is 24 V and the duty cycle is %50. Finally, we take 48V from outputs of the system. At the same time, we see the current graphics of converter in the below.

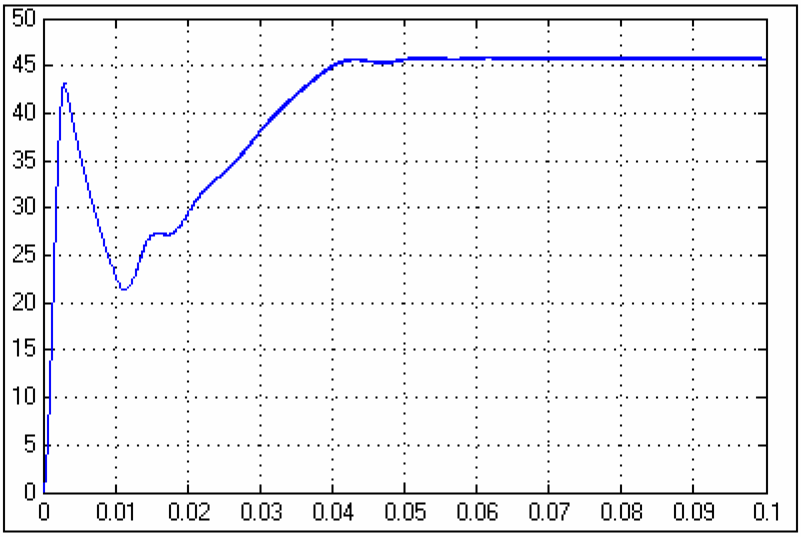


Figure 5.6 The output voltage of step-up converter

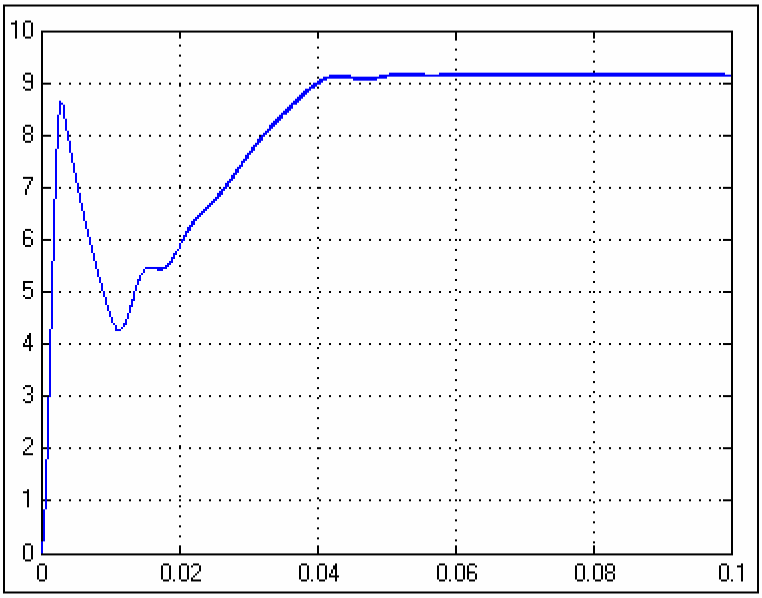


Figure 5.7 The output current of step-up converter

5.2 Simulation Results of Step-Up Boost Converter for Open-Loop Control

The system, which is given above, is analyzed by using Matlab program. The system controlled by open-loop. And the PWM signal is given signal generator. And the duty ratio is given %40.

In our system, the PWM switching signal is 20 kHz. The component, which has been used, has been described in chapter four. The values, which has been found, of components have been is replaced in the simulation program. The load of converter is only resistive load. The circuit is increases DC voltage from 24 V to 40V.

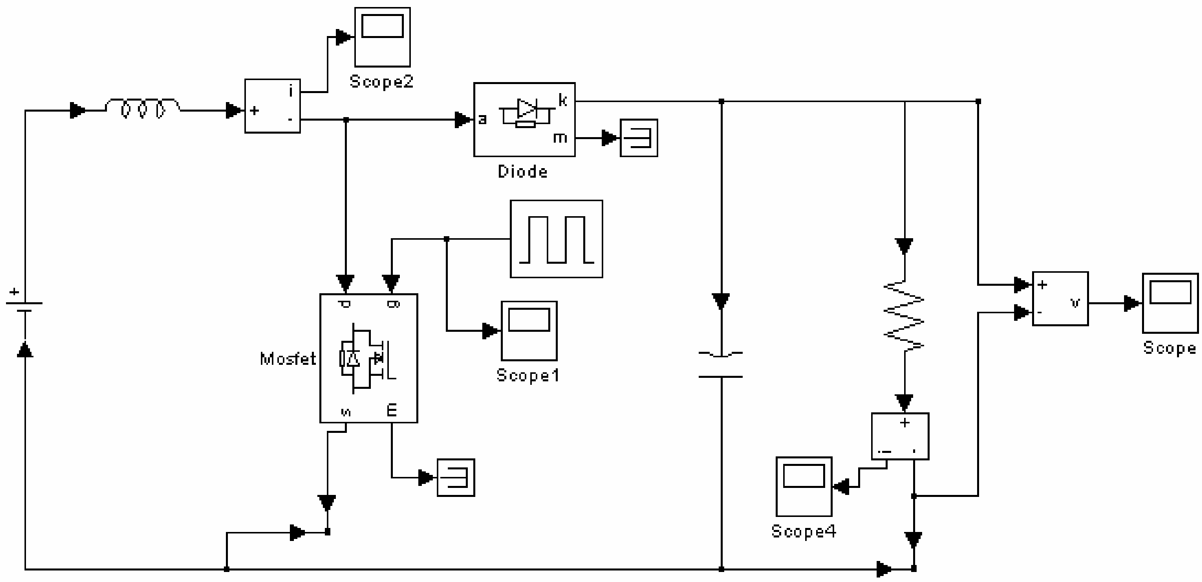


Figure 5.8 The Matlab simulation scheme for open-loop control

In our system, we have used MOSFET as a switching component. The PWM signal, which has been applied to MOSFET, is shown in the below.

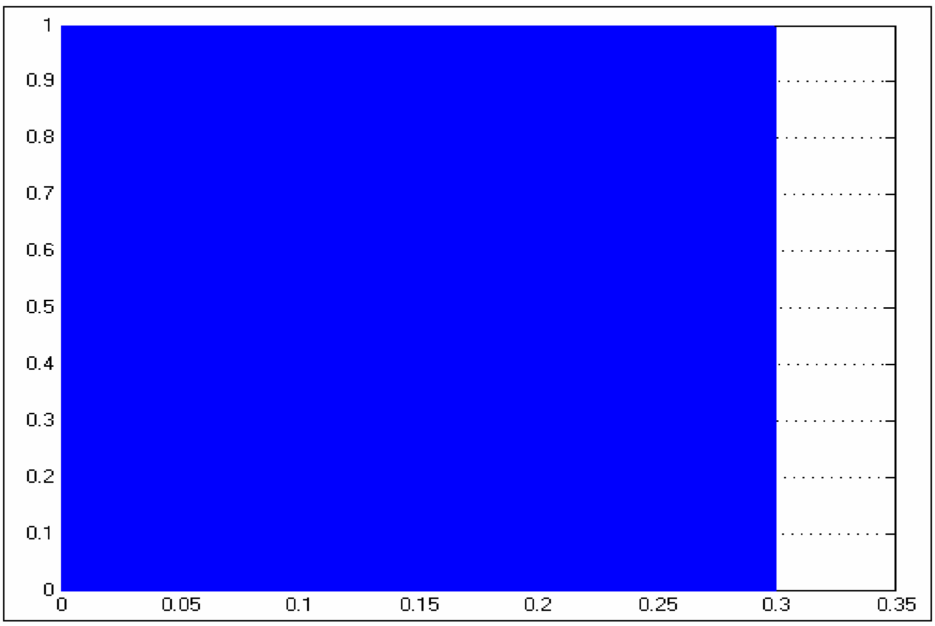


Figure 5.9 The signal of PWM for open-loop control

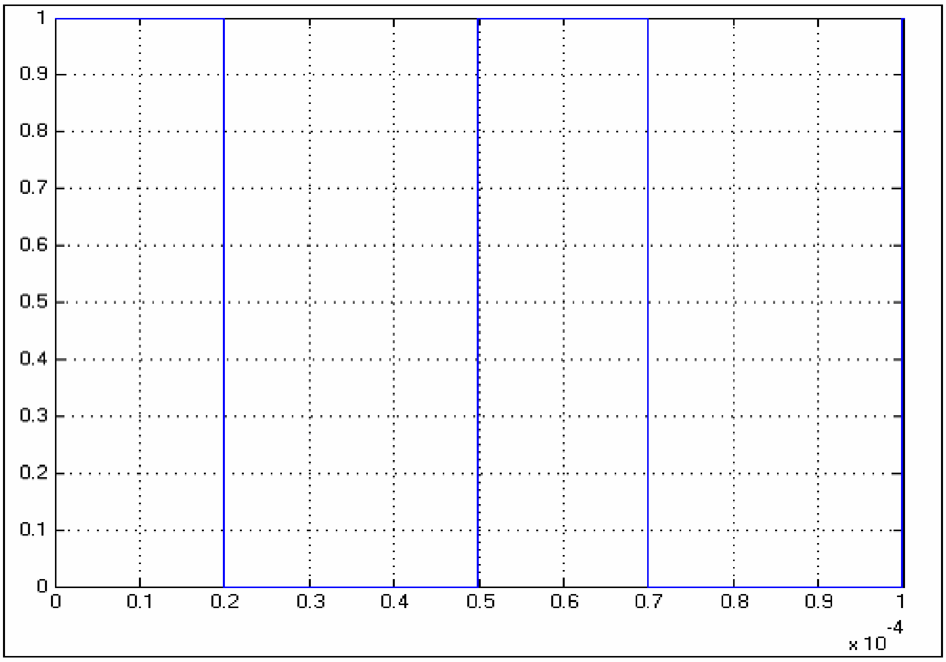


Figure 5.10 The signal of PWM (T= 50µs)

The results, which taken from outputs of the step-up converter, has been seen. The initial value of step-up converter is 24 V and the duty cycle is %40. Finally, we take 40V from outputs of the system. At the same time, we see the current graphics of converter in the below.

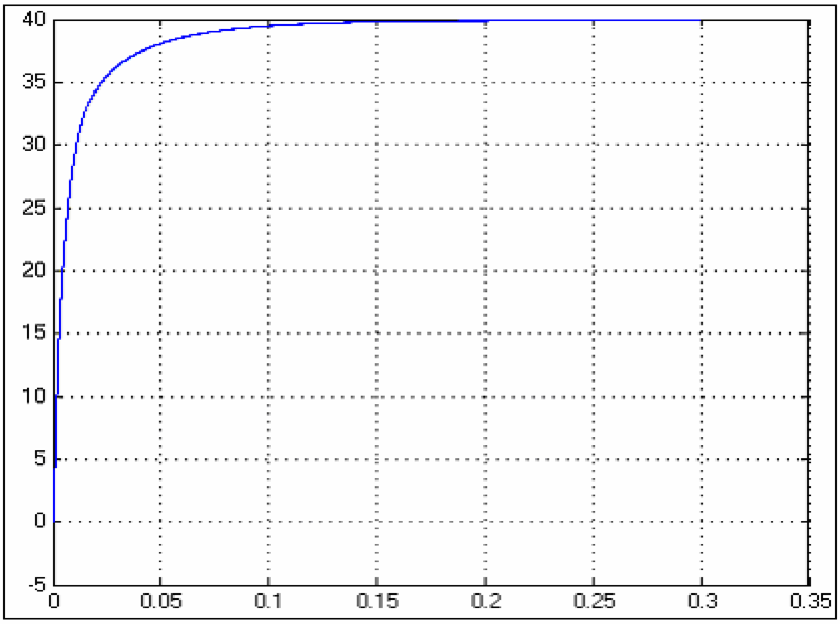


Figure 5.11 The signal of output voltage

The results, which taken from outputs of inductance current, has been seen. Figure 5.12 is showed inductance current.

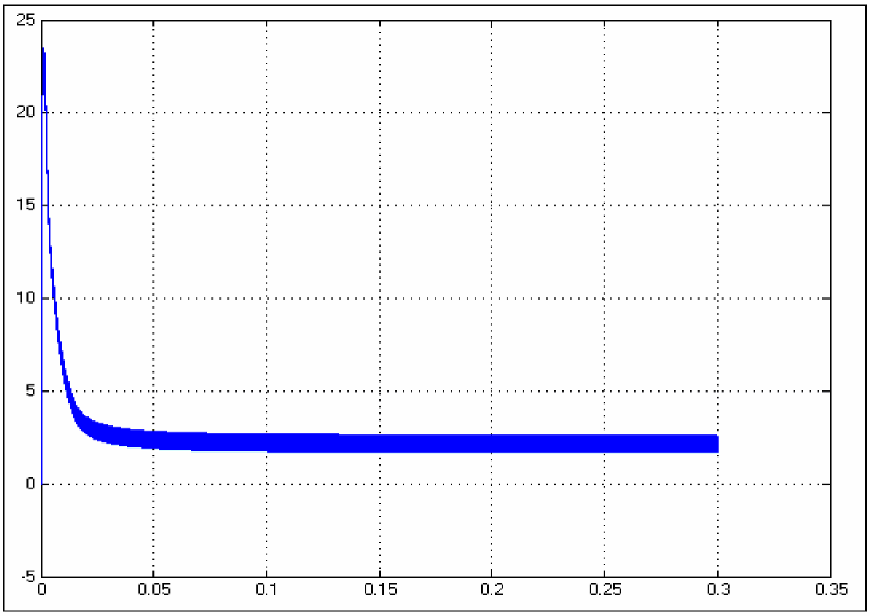


Figure 5.12 The signal of inductance current

5.3 Experimental Results of Step-up Converter for Open-Loop Control

The boost converter is controlled by a PWM signal provided by a microcontroller. Boost converter is the fact that this microcontroller measures the power going into the load and adjusts the duty cycle of the PWM in order to supply the most power possible to the load. The microcontroller used for this task was the PIC16F877 by Microchip. A brief description of the microcontroller follows.

The PIC16F877 from Microchip is a 40-pin microcontroller which has a number of useful features. These include an internal timer, analogue to digital modules and a PWM module, all of which are used by our application. The other technical information about PIC16F877 is given in Appendix B.

Figure 5.13 is showed experimental work. We use this works AN589 programmer, PIC control circuit and Boost Converter circuit.

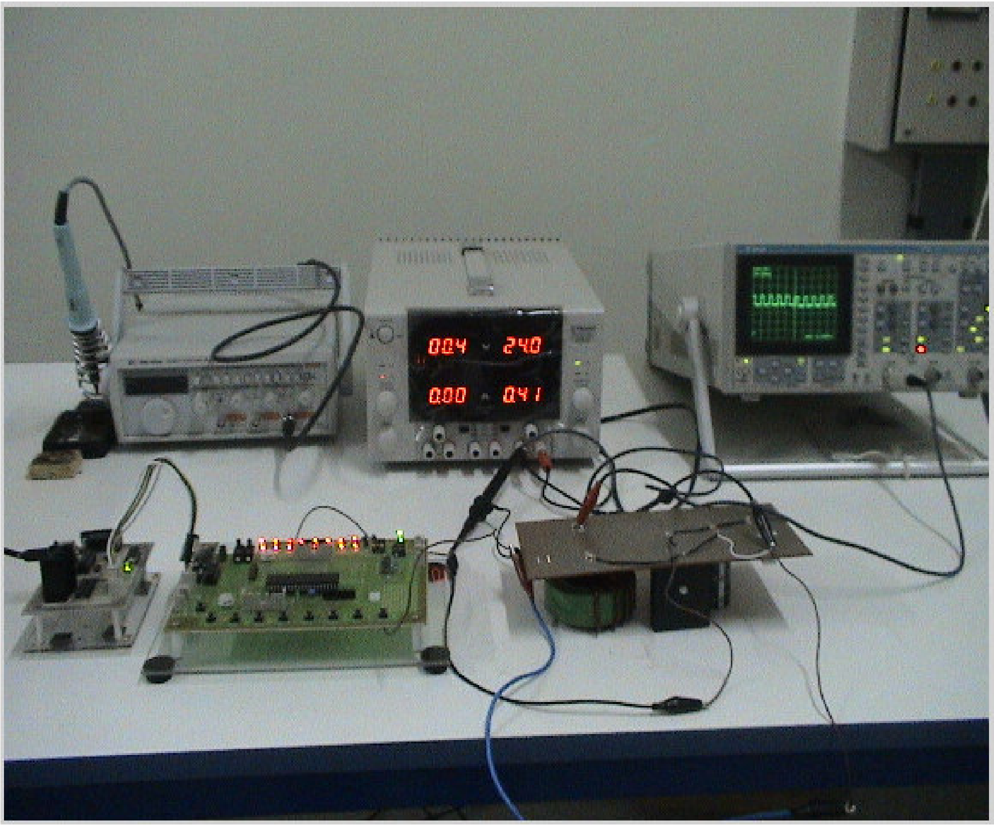


Figure 5.13 The photograph of experimental circuit

The step-up chopper is controlled with microcontroller. For this reason, PIC 16F877 is used. The software of PIC 16F877 is written by Assembly machine language. Also, in this thesis the features and structures of PIC 16F877 are explained briefly.

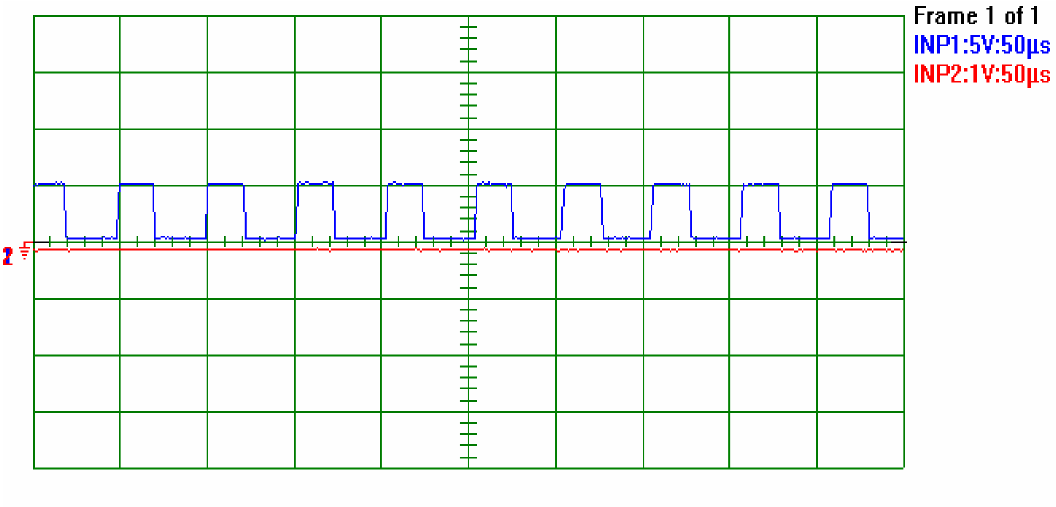


Figure 5.14 The graphic of PWM signal of experimental result

Figure 5.14 is given in above and this figure is showed the PWM signal of 20 kHZ of experimental results.

CHAPTER SIX
CONCLUSIONS

6. Conclusions

In this work, the design and simulation of a step-up dc/dc converter is performed. This converter is analyzed by using Matlab. The fuzzy logic is implemented to specify the duty ratio. The membership functions and the rule table of the developed fuzzy logic controller were formed by trail and error method using the set resemblance programmer.

Implementation of circuit is carried out in the laboratory. The converter is operated as open loop. The control program was written in Assembly machine language and transferred into PIC 16F877 micro-controller.

PWM drive signal is obtained using the micro-controller and the control sign was renewed with a 20 kHz frequency. The adjustment of output voltage by the fuzzy logic controller was done according to the reference voltage input without any deviation and no corrugation was seen after reaching the set point. The results of simulation program and the circuit designed are given in this study.

REFERENCES

Akçayol, M.A. (2004). Bulanık Mantık Denetimli Katodik Koruma Devresi. *G.U. Journal of Science, 17 (1): 111-127.* Retrieved 2004, from Gazi Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü tez arşivi.

Altınbaşak, O. (2000). *Mikrodenetleyiciler ve PIC Programlama.* NY: Altaş Basım Yayın Dağıtım.

Bose, B.K. (1992). *Modern Power Electronics: Evaluation, Technology and Applications.* NY: IEEE Press.

Criscione, M., Giustolui, G., Lionetto, A., Muscara, M., Palumbo, G. (2000). A Fuzzy Controller For Step-up DC-DC Converters. *IEEE Transaction on Power Electronics.* Retrieved December, 2000, from IEEE database.

Elmas, Ç. (2003). *Bulanık Mantık Denetleyiciler (Kuram, Uygulama, Sinirsel Bulanık Mantık).* NY: Seçkin Yayıncılık .

Gupta, T., Boudreaux, R.R., Nelms, R.M., Hung, J.Y. (1997). Implementation Of A Fuzzy Controller For DC-DC Converters Using An Inexpensive 8-b Microcontroller. *IEEE Transaction on Industrial Electronics, Vol: 44, No: 5.* Retrieved October, 1997, from IEEE database.

Kosko, B. (1997). *Fuzzy Engineering*. NY: Prentice-Hall International, Inc.

Mattavelli, P., Rosetto, L., Spiazzi, G., Tenti, P.P. (1997). General-Purpose Fuzzy Controller for DC-DC Converters. *IEEE Transaction on Power Electronics, Vol: 12, No: 1, 79-86.* Retrieved January, 1997, from IEEE database.

Middlebrook, R.D., Cuk, S. (1976). A General Unified Approach to Modeling

Switching Converter Power Stages. *IEEE Power Electronics Specialist Conf. 18-31*. Retrieved 1976, from IEEE database.

Mohan, N., & Undeland, T. M., & Robbins, W.P. (1989). *Power Electronics: Converters, Applications and Design.* (2nd ed.). NY: John Willey & Sons, Inc.

Rashid, M.H. (1993). *Power Electronics: Circuits, Devices and Applications.* (2nd ed.). NY: Prentice-Hall, Inc.

Ross, T.J. (1995). *Fuzzy Logic With Engineering Applications*. NY: McGraw-Hill, Inc.

So, W.C., Tse, C.K., Lee, Y.S. (1996). A Fuzzy Controller For DC-DC Converter. *IEEE Transaction of Power Electronics.* Retrieved 1996, from IEEE database.

Şen, Z. (2001). *Bulanık Mantık Ve Modelleme İlkeleri*. NY: Bilge Kültür Sanat.

PIC16F87X Data Sheet. (2003). 28/40 Pin 8-bit CMOS FLASH Microcontrollers. *Microchip Technology Inc, USA.* Retrieved March 5, 2003, from http://www.microchip.com/download/lit/pline/picmicro/families/16F87X/30292.pdf

APPENDIX A

```
    LIST    P=16F877,R=DEC
      INCLUDE "P16F877.INC"   ;file include

CYCLE       EQU   H'10'
TM1   EQU   H'11'
TM2   EQU   H'12'
MDATA       EQU   H'13'
COUNT1      EQU   H'14'
COUNT2      EQU   H'15'
RODAT       EQU   H'16'

;initialize

INIT  BCF   STATUS,RP1
      BSF   STATUS,RP0 ;Change page1
      CLRF  TRISA
      CLRF  TRISB
      CLRF  TRISD
      CLRF  TRISE
      MOVLW       B'11111111'   ;PORTA DATA
      MOVWF       TRISC         ;PORTA all input
      BCF   STATUS,RP0 ;Change page0
      CLRF  PORTA               ;PORTB  clear
      CLRF  PORTB
      CLRF  PORTD
      CLRF  PORTE

MAIN BTFSC        PORTC,7
```

```
        GOTO MAIN

        BTFSSPORTC,7
        GOTO MAIN

LOOP
        MOVLW       D'250'          ;PWM cycle = 250 >> W reg
        MOVWF       CYCLE                   ;W reg >> CYCLE
        MOVLW       D'25'           ;initial value = 25(PWM on time) >> Wreg
        MOVWF       MDATA                   ;Wreg >> MDATA

;**********************************************************
;   Switch routine
;**********************************************************

SW
        BTFSSPORTC,7                ;PORTA0 SW OFF(1) >> skip SW on(0) >>
next
        GOTO STOP           ;PORTA0 SW on then goto STOP
;       BTFSSPORTC,4                ;PORTA1 SW off(1) >> skip  SW on(0) >> next
;       GOTO UPSP           ;PORTA1 SW on then goto UPSP(up speed)
;       BTFSSPORTC,3                ;PORTA2 SW off(1) >> skip  SW on(0) >> next
;       GOTO DOSP           ;PORTA2 SW on then goto DOSP(down speed)

PWM     MOVFMDATA,W                 ;MDATA >> Wreg(PWM on time)
        MOVWF       TM2             ;Wreg >> TM2
        SUBWF       CYCLE,W                 ;PWM cycle(CYCLE) - PWM ON
time(Wreg) = Wreg(PWM off time)
        MOVWF       TM1             ;Wreg >> TM1(PWM off time)

;**********************************************************
```

```
;*******************************************************
        MOVLW       B'11111111'
        MOVWF       PORTD               ;
PON     NOP         ;call 0.03ms timer
        DECFSZ      TM2,F       ;TM2 - 1 >> TM2 = 0? if TM2=0 then skip
        GOTO PON            ;goto PON

;*******************************************************
;  PWM pulse off time is 250*0.03 - ontime (ms)
;*******************************************************
        MOVLW       B'00000000'
        MOVWF       PORTD               ;
POFF                ;call 0.03ms timer
        DECFSZ      TM1,1       ;TM1 - 1 >> TM1 = 0? if TM1=0 then skip
        GOTO        POFF        ;goto POFF

        GOTO SW             ;goto SW(SWTCH)

;*************************************************
; if porta1 switch is pushed processing this routine
;*************************************************

UPSP    INCF  MDATA,F           ;MDATA + 1 >> MDATA(max = 250)
        MOVLW       D'250'      ;CYCLE DATA(250) >> Wreg
        SUBWF       MDATA,W             ;MDATA - Wreg >> Wreg
        BTFSSSTATUS,Z   ;Wreg = 0 then skip(MDATA = 250 is max value)
        GOTO PWM            ;goto PWM
        DECF  MDATA,F           ;MDATA(250) - 1 = MDATA(249) >> It is
possible to continue push swtch
        GOTO PWM            ;goto PWM
```

```
;**************************************************
; if porta2 switch is pushed processing this routine
;**************************************************

DOSP  DECF  MDATA,F         ;MDATA - 1 >> MDATA(min -0)
      BTFSSSTATUS,Z    ;Wreg = 0 then skip(MDATA = 0 is minimum value)
      GOTO PWM         ;goto PWM
      INCF  MDATA,F         ;MDATA(0) + 1 = MDATA(1) >> It is possible
to continue push swtch
      GOTO PWM         ;goto PWM


STOP  GOTO STOP

;**************************************************
; timer subroutine
;**************************************************


      END
```

APPENDIX B

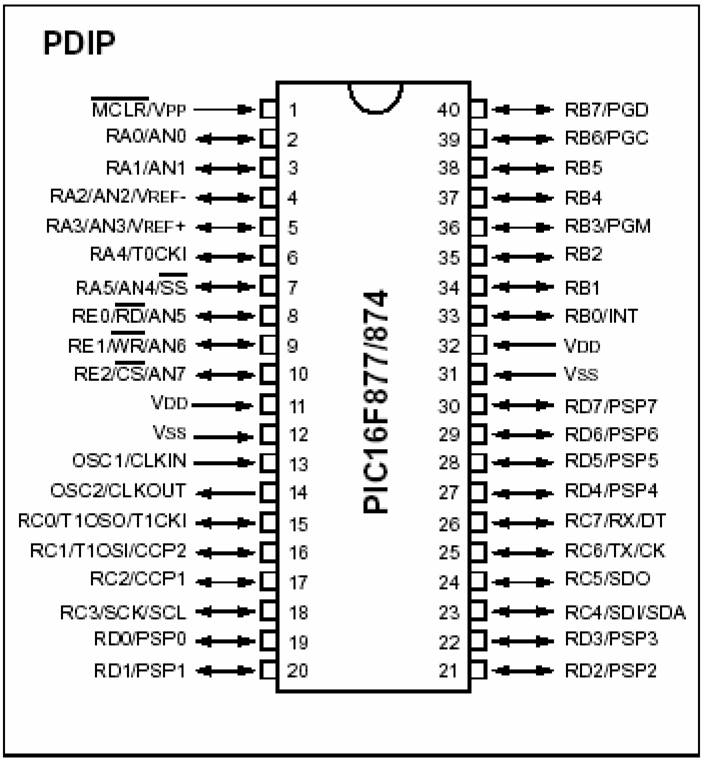
40 Pin 8- Bit CMOS FLASH Microcontrollers

Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC- 20 MHz clock input, DC-200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory, Up to 368 x 8 bytes of Data Memory (RAM), Up to 256 x 8 bytes of EEPROM Data Memory
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/ EEPROM technology
- Fully static design
- In-Circuit Serial Programming (ICSP) via two pins
- Single 5 V ICSP capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0 V to 5.5 V
- High Sink/Source Current: 25 mA

- Commercial, Industrial and Extended temperature ranges
- Low-power consumption:
  - → < 0.6 mA typical @ 3V, 4 MHz
  - → 20 µA typical @ 3V, 32 kHz
  - → < 1 µA typical standby current

Pin Diagram:



Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during SLEEP via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler

- Two Capture, Compare, PWM modules

  → Capture is 16-bit, max. resolution is 12.5 ns
  → Compare is 16-bit, max. resolution is 200 ns
  → PWM max. resolution is 10-bit

- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI (Master mode) I^2C (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) 8-bits wide, with external RD, WR and Cs controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

Device	Program FLASH	Data Memory	Data EEPROM
PIC16F874	4K	192 Bytes	128 Bytes
PIC16F877	8K	368 Bytes	256 Bytes

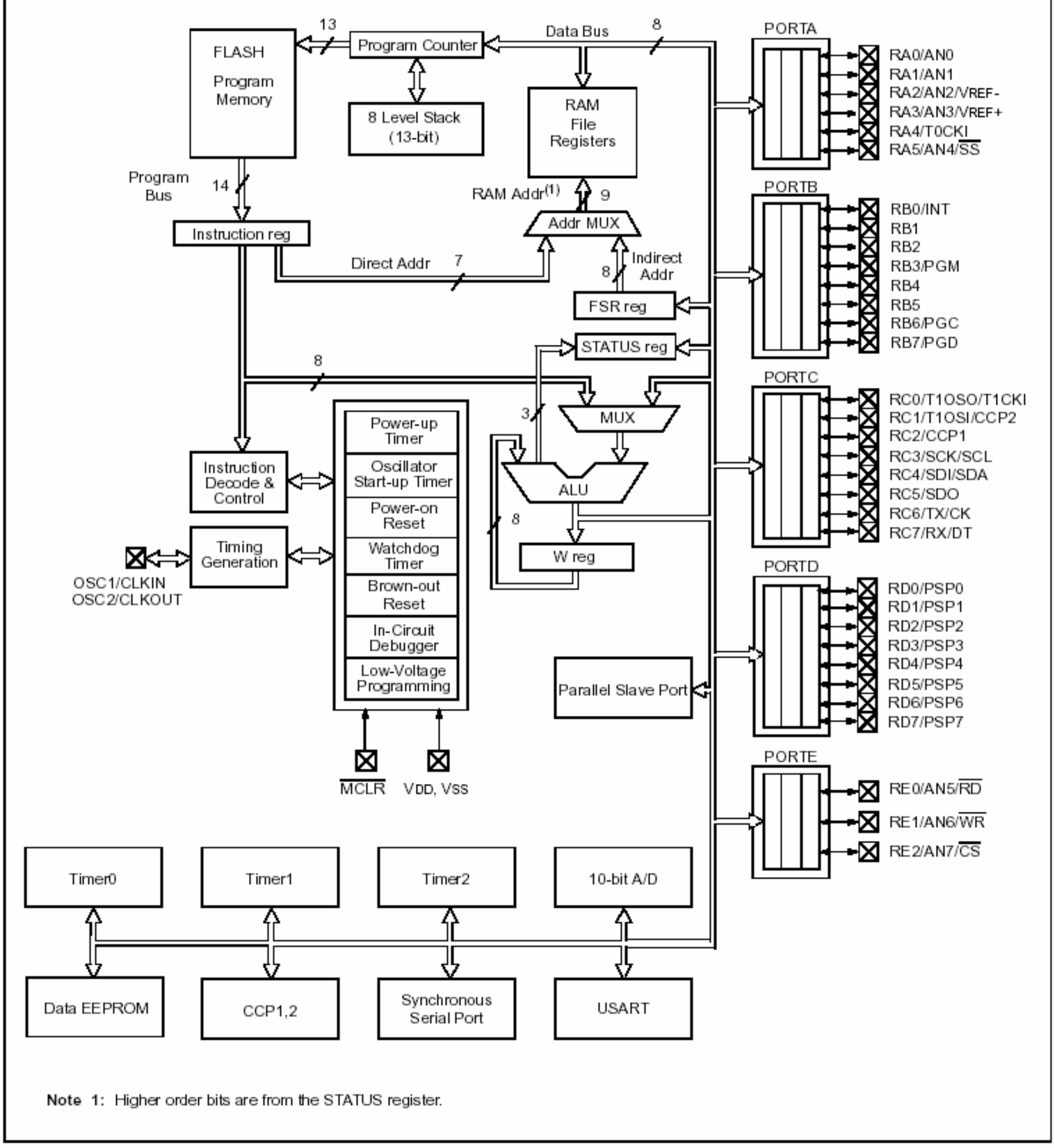


Figure 1. PIC 16F877 Block Diagram

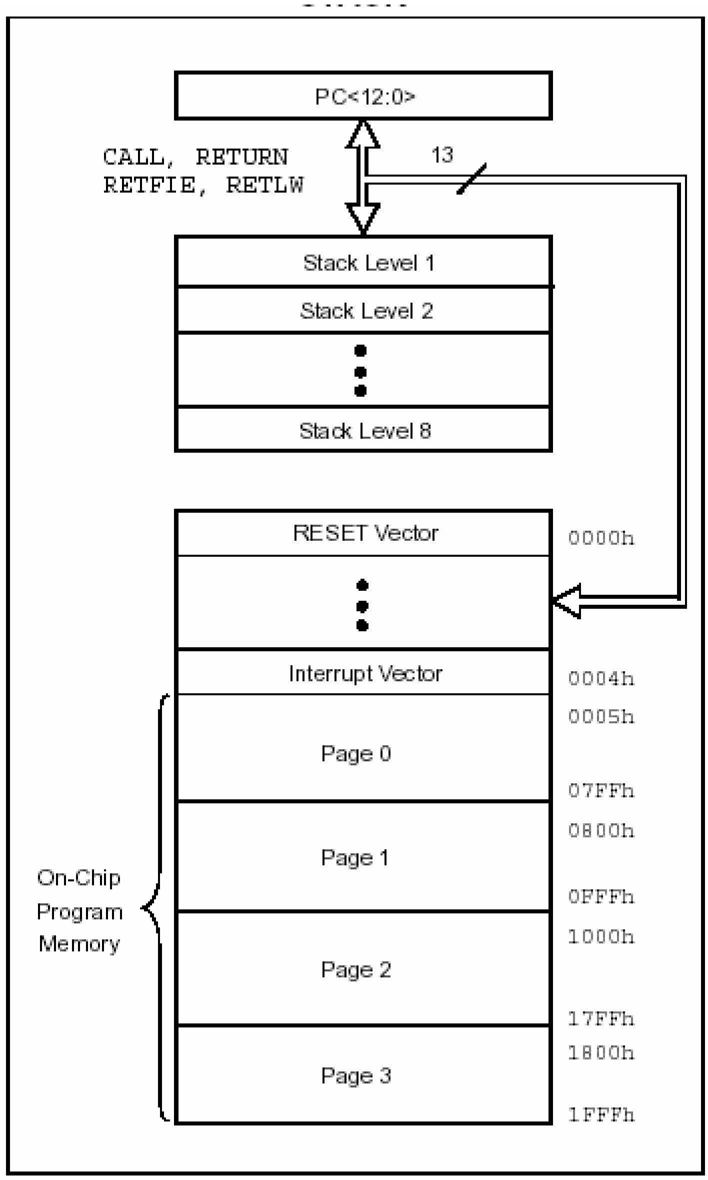


Figure 2. PIC16F877 Program Memory Map and Stack

Bank 0	File Address	Bank 1	File Address	Bank 2	File Address	Bank 3	File Address
Indirect addr.(*)	00h	Indirect addr.(*)	80h	Indirect addr.(*)	100h	Indirect addr.(*)	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
PORTD(1)	08h	TRISD(1)	88h		108h		188h
PORTE(1)	09h	TRISE(1)	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reserved(2)	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reserved(2)	18Fh
T1CON	10h		90h	General Purpose Register 16 Bytes	110h	General Purpose Register 16 Bytes	190h
TMR2	11h	SSPCON2	91h		111h		191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPADD	93h		113h		193h
SSPCON	14h	SSPSTAT	94h		114h		194h
CCPR1L	15h		95h		115h		195h
CCPR1H	16h		96h		116h		196h
CCP1CON	17h		97h		117h		197h
RCSTA	18h	TXSTA	98h		118h		198h
TXREG	19h	SPBRG	99h		119h		199h
RCREG	1Ah		9Ah		11Ah		19Ah
CCPR2L	1Bh		9Bh		11Bh		19Bh
CCPR2H	1Ch		9Ch		11Ch		19Ch
CCP2CON	1Dh		9Dh		11Dh		19Dh
ADRESH	1Eh	ADRESL	9Eh		11Eh		19Eh
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
General Purpose Register 96 Bytes	20h	General Purpose Register 80 Bytes	A0h	General Purpose Register 80 Bytes	120h	General Purpose Register 80 Bytes	1A0h
			EFh		16Fh		1EFh
		accesses 70h-7Fh	F0h	accesses 70h-7Fh	170h	accesses 70h - 7Fh	1F0h
	7Fh		FFh		17Fh		1FFh

▭ Unimplemented data memory locations, read as '0'.
* Not a physical register.

Note 1: These registers are not implemented on the PIC16F876.
2: These registers are reserved, maintain these registers clear.

Figure 3. PIC16F877 Register File Map