

**Vote: 77955**

**FIELD PROGRAMMABLE GATE ARRAY (FPGA) BASED  
FUZZY LOGIC CONTROLLER FOR BOOST CONVERTER**

**DR. SHAHRIN BIN MD. AYOB**

**UNIVEERSITI TEKNOLOGI MALAYSIA**

**2011**



PUSAT PENGURUSAN PENYELIDIKAN (RMC)

UTM/RMC/F/0024 (1998)  
Pindaan: 0

**BORANG PENGESAHAN  
LAPORAN AKHIR PENYELIDIKAN**

TAJUK PROJEK :

FIELD PROGRAMMABLE GATE ARRAY (FPGA)

BASED FUZZY LOGIC CONTROLLER

FOR BOOST CONVERTER

Saya DR. SHAHRIN BIN MD. AYOB

**(HURUF BESAR)**

Mengaku membenarkan **Laporan Akhir Penyelidikan** ini disimpan di Perpustakaan Universiti Teknologi Malaysia dengan syarat-syarat kegunaan seperti berikut :

1. Laporan Akhir Penyelidikan ini adalah hakmilik Universiti Teknologi Malaysia.
2. Perpustakaan Universiti Teknologi Malaysia dibenarkan membuat salinan untuk tujuan rujukan sahaja.
3. Perpustakaan dibenarkan membuat penjualan salinan Laporan Akhir Penyelidikan ini bagi kategori TIDAK TERHAD.
4. \* Sila tandakan ( / )

SULIT

(Mengandungi maklumat yang berdarjah keselamatan atau Kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972).

TERHAD

(Mengandungi maklumat TERHAD yang telah ditentukan oleh Organisasi/badan di mana penyelidikan dijalankan).

TIDAK  
TERHAD

  
TANDATANGAN KETUA PENYELIDIK

**DR. SHAHRIN B. MD. AYOB**

Pensyarah  
Fakulti Kejuruteraan Elektrik  
Universiti Teknologi Malaysia  
81310 UTM Skudai Johor,  
Johor Darul Takzim.

Nama & Cop Ketua Penyelidik

Tarikh : 27 JANUARI 2011

**CATATAN :** \* Jika Laporan Akhir Penyelidikan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali sebab dan tempoh laporan ini perlu dikelaskan sebagai SULIT dan TERHAD.

## **ABSTRACT**

This project presents the application of the Single Input Fuzzy Logic Controllers (SIFLC) to regulate the output voltage of a Boost (step-up) DC to DC power converter. The SIFLC is derived from the Signed Distance Method which reduces the multi-input Fuzzy Logic Controller (FLC) with Toeplitz rule table structure to a single input FLC. Effectively, it allows for the rule table to be approximated to a one dimensional piecewise linear control surface. To validate the effectiveness of the SIFLC compared to the Conventional FLC, simulation and experimental works are carried out. The results show that the Boost converter performance is exactly identical when subjected to both controllers. However SIFLC requires nearly an order of magnitude less time to execute its algorithm.

## ABSTRAK

Projek ini mempersembahkan penggunaan aplikasi Pengawal Logik Kabur Masukan Tunggal (SIFLC) untuk mengawal keluaran voltan kepada penukar DC-DC. Pengawal SIFLC diterbitkan daripada Kaedah Jarak Bertanda untuk mengurangkan jumlah masukan kepada Pengawal Logik Kabur yang lazim. Dengan menggunakan kaedah ini, ia menjadikan dimensi jadual peraturan pengawal boleh dikurangkan kepada satu dimensi permukaan garis lurus secebis. Untuk mengesahkan keberkesanan SIFLC ini berbanding dengan FLC lazim, kerja simulasi dan eksperimen dijalankan. Hasil keputusan menunjukkan bahawa kedua-dua pengawal terbabit mempunyai keputusan yang sama dengan menggunakan. Walaubagaimanapun, masa yang diambil oleh SIFLC untuk melaksanakan arahan lebih pantas.

## TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	<b>ABSTRACT</b>	iii
	<b>ABSTRAK</b>	iv
	<b>TABLE OF CONTENTS</b>	v
	<b>LIST OF TABLES</b>	viii
	<b>LIST OF FIGURES</b>	ix
	<b>LIST OF SYMBOLS</b>	xii
	<b>LIST OF ABBREVIATIONS</b>	xiii
<b>1</b>	<b>INTRODUCTION</b>	1
	<b>1.1 Overview</b>	1
	<b>1.2 Objective, Scope and Importance     of Research</b>	3
	<b>1.2.1 Objective of Research</b>	3
	<b>1.2.2 Scope of Research</b>	4
	<b>1.2.3 Importance of Research</b>	4
	<b>1.3 Organization of Thesis</b>	5
<b>2</b>	<b>REVIEW OF BOOST CONVERTER AND ITS CONTROL</b>	6
	<b>2.1 Introduction</b>	6
	<b>2.2 Boost converter circuit and its     average model</b>	7
	<b>2.3 PID Controller</b>	9
	<b>2.4 Deadbeat Controller</b>	10

	<b>2.5 Conventional Fuzzy Logic Controller (CFLC)</b>	12
	<b>2.5.1 Fuzzification</b>	13
	<b>2.5.2 Rule base</b>	13
	<b>2.5.3 Defuzzification</b>	15
<b>3</b>	<b>SINGLE INPUT FUZZY LOGIC CONTROLLER, PIECEWISE LINEAR CONTROL SURFACE AND CONTROLLER DESIGN</b>	17
	<b>3.1 Introduction</b>	17
	<b>3.2 Single distance method</b>	18
	<b>3.3 SIFLC Control Surface</b>	21
	<b>3.4 Fuzzy logic controller design</b>	23
	<b>3.4.1 Using the Conventional FLC (CFLC)</b>	24
	<b>3.4.2 Using Single Input Fuzzy Controller (SIFLC)</b>	26
<b>4</b>	<b>SIMULATION RESULTS</b>	28
	<b>4.1 Introduction</b>	28
	<b>4.2 Boost converter parameters</b>	29
	<b>4.3 Comparison of SFLLC and CFLC</b>	29
	<b>4.4 Comparison between SIFLLC and PI controller</b>	34
	<b>4.5 Computational Comparison</b>	37
<b>5</b>	<b>HARDWARE IMPLEMENTATION AND EXPERIMENTAL RESULTS</b>	39
	<b>5.1 Introduction</b>	39
	<b>5.2 Controller circuit</b>	40
	<b>5.2.1 System schematic</b>	40
	<b>5.2.2 DE2 Board</b>	42

	<b>5.3 Proposed ADC</b>	43
	<b>5.3.1 Successive Approximation ADC</b>	43
	<b>5.3.2 Structure of proposed ADC</b>	44
<b>6</b>	<b>CONCLUSIONS AND FUTURE WORK</b>	47
	<b>6.1 Conclusion</b>	47
	<b>6.2 Future work suggestions</b>	48
	<b>REFERENCES</b>	49

**LIST OF TABLES**

<b>TABLE NO.</b>	<b>TITLE</b>	<b>PAGE</b>
3.1	FLC Rule table with Toeplitz structure	19
3.2	The reduced SIFLC rule table using the Signed distance method	20
3.3	Rule table for CFLC	25
3.4	Rule table for SIFLC	27



## LIST OF FIGURES

<b>FIGURE NO.</b>	<b>TITLE</b>	<b>PAGE</b>
2.1	Control methods for DC-DC converters	7
2.2	Boost converter circuit	7
2.3	Simulation model of Boost converter	8
2.4	PI controller	9
2.5	Simulink schematic of PI controlled Boost Converter	10
2.6	Mapping of $s$ -plane to $z$ -plane	11
2.7	Typical deadbeat controllers	11
2.8	Fuzzy Logic Controller schematic	12
2.9	FLC overall structure	13
2.10	Weight average method defuzzification	16
3.1	Derivation of distance variable, $d$	19
3.2	Controller structure, (a) SIFLC, (b) CFLC	21
3.3	PWL Control surface with symmetrical input and output membership functions	22
3.4	PWL control surface with peak locations of the asymmetrical input MF	22
3.5	PWL control surface with more piecewise linear regions using more input and output asymmetrical MFs	23
3.6	SIFLC with PWL Control Surface	23
3.7	Simulink simulation model for Boost converter with CFLC	24
3.8	Symmetric CFLC (a) Error MFs (b) Change of errorMFs	

	(c) Output MFs	
	(d) Control Surface associated with FLC	25
3.9	Asymmetrical CLFC, (a) Error MFs, (b) Change of Error MFs, (c) Output MFs	
	(d) Control surface associated with FLC	26
3.10	Simulink simulation model for Boost converter with SIFLC	27
3.11	PWL Control surface for SIFLC (a) with symmetrical input MFs (b) with asymmetrical input MFs	27
4.1	Output voltage responses for step change in input voltage from 10V to 15V (a) CFLC (b) SIFLC	31
4.2	Inductor current responses for step change in input voltage from 10V to 15V, (a) CFLC (b) SIFLC	31
4.3	Output voltage response for step change in reference voltage from 15V to 20V when input voltage $V_i=10V$ . (a) CFLC, (b) SIFLC	32
4.4	Inductor current response for step change in reference voltage from 15V to 20V when the input voltage $V_i=10V$ (a) CFLC (b) SIFLC	32
4.5	Voltage response for step change in load resistance from 5V to 10V when the input voltage $V_i=10v$ . (a) CFLC (b) SIFLC	33
4.6	Inductor current response for step change in load resistance from 5 $\Omega$ to 10 $\Omega$ when the input voltage $i=10V$ .(a) CFLC, (b) SIFLC	33
4.7	Output voltage of Boost converter for step change in input voltage from 10v to 15v in $t=0.2$ and $R=5$ Ohm	34
4.8	Inductor current of Boost converter for step change in input voltage from 10v to 15v in	

	t=0.2 and R=5 Ohm	34
4.9	Output voltage of Boost converter for step change in reference voltage from 10v to 15v in t=0.2 and R=5ohm,Vi=10	35
4.10	Inductor current of Boost converter for step change in reference voltage from 10v to 15v in t=0.2 and R=5 ohm,Vi=10v	35
4.11	Output voltage of Boost converter for step change in load value from 3 ohm to 5 ohm in t=0.2 and Vref=20v, Vi=10	36
4.12	Inductor current of Boost converter for step change in load value from 3 ohm to 5 ohm in t=0.2 and Vref=20v, Vi=10	36
4.13	Input of PWL control surface of SIFLC for step change in load value from 3 ohm to 5 ohm in t=0.2 and Vref=20v, Vi=10	37
4.14	CPU time consumed by CFLC and SIFLC	38
5.1	Digital controller schematic	41
5.2	Altera DE2 Educational board	43
5.3	Basic Successive Approximation ADC	44
5.4	Proposed ADC schematic	45
5.5	Simulation of proposed ADC in Quartus	46
5.6	Boost converter response using SIFLC in Vo=15v and load disturbance from 10Ω to 5Ω (200% of nominal load)	
	(a) Using symmetrical MFs	
	(b) Using asymmetrical MFs (PWL2)	47
5.7	Boost converter output voltage using SIFLC in 10Ω load and reference change from 12.5V to 15V	
	(a) Asymmetrical MFs (PWL2)	
	(b) Using Asymmetrical MFs (PWL1)	
	(c) Symmetrical MFs	48

**LIST OF SYMBOLS**

C - Capacitance

d - Distance

D - Duty cycle

e(k), E - Error signal

$\dot{e}(k)$ , DE - Derivative of Error

f<sub>s</sub> - Switching frequency

i<sub>L</sub> - Inductor current

K<sub>e</sub> - Input scaling factor for error signal

K $\dot{e}$ , DE - Input scaling factor for derivative of error signal

K $\dot{u}$ , DU- Output scaling factor

K<sub>i</sub> - Integral gain of PI controller

K<sub>p</sub> - Proportional gain of PI controller

L - Filter inductance

R - Load resistance

R<sub>L</sub> - Inductor resistance

V<sub>o</sub> - Output voltage

$\lambda$  - Main diagonal line slope

$\mu$ - Degree of membership

## LIST OF ABBREVIATIONS

ADC - Analogue-to-Digital Converter  
CFLC - Conventional Fuzzy Controller  
CoG - Centre of Gravity  
CPU - Central Processing Unit  
CS - Control Surface  
DSP - Digital Signal Processing  
FLC - Fuzzy Logic Controller  
FPGA - Field-Programmable Gate Array  
MF - Membership Function  
PI - Proportional-Integral  
PWL - Piecewise Linear  
PWM - Pulse Width Modulation

## CHAPTER 1

### INTRODUCTION

#### 1.1 Overview

DC to DC step-up power converter, or more popularly known as the Boost converter, is widely used in power electronics systems. Its application is widespread and wide ranging-Boost power supply can be found in the tiniest cell phone (mill watt) to the high power train propulsion system (hundreds of kilowatts). One of the main requirements of the converter is the robustness of its controller. A good controller should perform the following tasks: (1) able to regulate output voltage when the input voltage and reference is changed (2) able to stabilize the system for any input disturbances and load changes. The performance of the controller is normally characterized by its response to a step input reference, i.e. transient percentage of overshoot, settling time and steady state error.

Due to its nonlinear and time-invariant nature, the design of high performance controller for the Boost converter presents a challenging task. Traditionally, classical methods such as frequency response and root locus/pole placement techniques are employed. Examples of classical controllers are the Proportional Integral Derivative (PID)[1], Deadbeat controllers [2] and sliding mode controllers [3]. These

controllers are known as “model based”, relying heavily on mathematical model of the converter for accurate control action. An equivalent circuit-averaging model is derived to determine the converter’s variables within a switching period. Based on the averaged model, a suitable small-signal model is obtained by performing small signal perturbation and linearization around a specific (nominal) operating point. For a Boost converter, it is known that the poles and a right-half plane zero are dependent on the load resistance,  $R$  [4]. Since classical controllers are designed to operate at one nominal operating point (i.e. fixed duty cycle,  $D$ ), they are unable to respond satisfactorily to a large operating point variation (i.e. large change in  $D$ ). Similarly, it could not cope with large load disturbance (large change in  $R$ ). Moreover, classical controllers are sensitive to the changes in system parameters, resulting in unpredictable control performance when subjected to changes in temperature, aging, operating point etc.

To alleviate the dependency on the mathematical model, “non-model based” controllers have been proposed. Among the most popular is the Fuzzy Logic Controller (FLC). In essence, FLC is a linguistic-based controller that tries to solve problems by means of systematic rule inferences. It does not require precise mathematical model, very robust and has excellent immunity to external disturbances [5]. Although promising, FLC requires substantial computational power due to complex decision making processes, namely fuzzification, rule base storage, inference mechanism and defuzzification operations. To obtain optimized performance, FLC require a much longer time because for most cases, the design is done heuristically [6],[7].

The applications of the conventional FLC (CFLC) for Boost converter are reported by several researchers [8], [9], [10]. Implementation of CFLC using slow processor is usually not adequate; for example, in [10], an 8-bit microcontroller running at 16MHz clock takes about 250 $\mu$ s. This means if the duty cycle is to be updated for every new pulse, the switching frequency is very low, i.e. less than 4KHz- which is very undesirable for power converter. Consequently, many researchers opted for digital signal processor (DSP) [8]. However, DSP is costly and may not be justifiable in certain applications. Another way to reduce the computational burden

of CFLC is by reducing the number of rules inference. But by doing so, the accuracy of its control action is reduced.

In this project, a simplified Fuzzy controller, known as the Single Input FLC (SIFLC) is used to regulate the output voltage of a Boost converter. It paves a way for fast FLC execution without compromising the accuracy of the control performance. The SIFLC is based on the “signed distance method” which reduces the CFLC into a single-input FLC [11]. Effectively it simplifies the rule table to a one dimensional array. This reduction allows for the SIFLC control surface to be approximated by a simple piecewise linear (PWL) graph, resulting in significant simplification in parameter tuning and design [12]. The only constraint of this method is that it applies only to FLC with Toeplitz structure [11]. Fortunately, it was found that most of rule table used in power electronics converter are of such nature.

## **1.2 Objective, Scope and Importance of Research**

### **1.2.1 Objective of Research**

The objective of this research is to design and implement the Single Input Fuzzy Logic for Boost converter control using FPGA. The work concentrates on implementation of controller by Verilog hardware description language. Furthermore this research attempts to implement new Analog to digital converter (ADC) by using embedded counter inside of FPGA; this ADC has better performance in term of speed and wiring area compare to conventional ADC.



### **1.2.2 Scope of Research**

In order to reach objective of research, the following steps will be done:

- Average model of Boost converter is derived via state equations.
- PI controller is designed for regulating output voltage of converter.
- Proportional and integral gains used to define conventional Fuzzy Logic controller.
- SIFLC is derived from CFLC and several disturbances applied to both systems controlled by CFLC and SFLC to show symmetry of responses for both controllers.
- Digital blocks are simulated using Quartus software to insure proper operations of controller.
- Actual system implementation is done by construction of low power Boost converter and programming FPGA to do controller function.

### **1.3 Importance of Research**

Conventional Fuzzy Logic Controller (CFLC) needs heavy computations because it requires fuzzification, reference engine and defuzzification sections. This issue leads to slow response of controller and low switching frequencies which increase voltage ripple and inductor size. In the proposed SIFLC all components of CFLC are replaced by one look up table, hence control algorithm operates in faster speed compare to CFLC. Implementation of controller in FPGA increases the controller speed in comparison with DSP due to availability of parallel processing in FPGA and accessibility to fundamental components of constructed system.

## **1.4 Organization of Thesis**

This is divided in to six chapters, which are outlined as follows:

- Chapter one introduces main objective of the project and scope area of the work.
- Chapter two discusses about Boost converter circuit and its average model. Also different control techniques for Boost converter are introduced briefly.
- Chapter three is insight to single distance method. Also piecewise linear control is introduced from the idea of SIFLC.
- Chapter four presents simulation results for various disturbances applied to the converter.
- Chapter five highlights hardware implementation of system and experimental results.
- Chapter six concludes the work and explains the results.

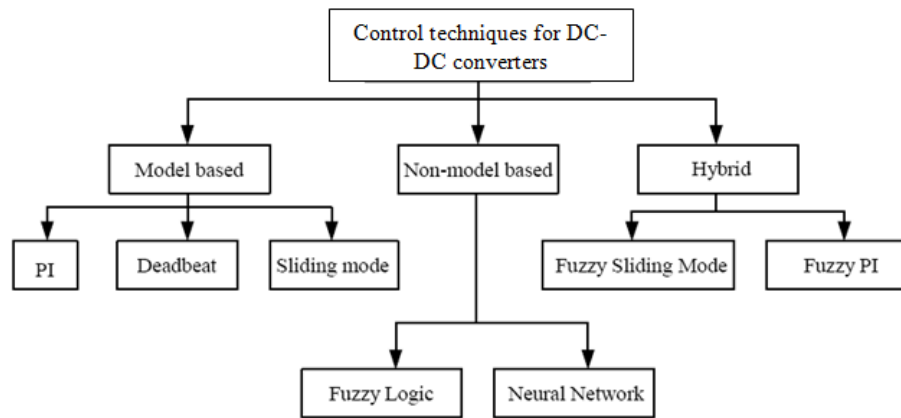
## CHAPTER 2

### REVIEW OF BOOST CONVERTER AND ITS CONTROL TECHNIQUES

#### 2.1 Introduction

In most of the applications it is needed to maintain the output voltage of converter regardless of changes in the load or input voltage. The DC-DC power converters are sited in middle stage of most of electrical power systems; their input is connected to solar cells and output is connected to inverters. Both input and output sides are prone to sudden changes in values, slow transient response increase losses in the system and leads to dramatic reduction of efficiency.

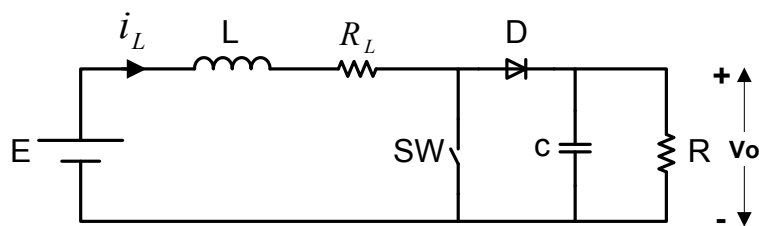
Several methods have been proposed by researchers to control output voltage of DC-DC converters these methods are illustrated in Figure 2.1. Classical methods (model based) rely on mathematical model of system hence they are sensitive to changes in transfer function. In opposite to classical methods, non-model based approaches use intelligence techniques which are not dependent on system mathematical model. For systems with non-constant transfer function or without mathematical methods non-model base controllers are more preferred.



**Figure 2.1 control methods for DC-DC converters**

## 2.2 Boost converter circuit and its average model

The circuit diagram of Boost converter is illustrated in Figure 2.2. When the switch SW is in the “on” (closed) state, the current in the boost inductor ( $i_L$ ) increases linearly and the diode is reverse biased. When SW is turned “off” (opened), the energy stored in the inductor is released through the diode to the output RC circuit. The converter is assumed to work in Continuous Current Mode in which inductor current not allowed to reach zero.



**Figure 2.2 Boost converter circuit**

To obtain the average model of the Boost converter,  $i_L$  and output voltage  $V_o$  are selected as state variables. The circuit has two states. i.e. SW is closed and opened. When SW is closed, the following expressions hold:

$$\frac{dV_o}{dt} = \frac{-V_o}{c \times R} \quad (2.1)$$

$$\frac{di_L}{dt} = \frac{E - i_L \times R_L}{L} \quad (2.2)$$

When SW is opened,

$$\frac{dV_o}{dt} = \frac{1}{c} \times \left( i_L - \frac{V_o}{R} \right) \quad (2.3)$$

$$\frac{di_L}{dt} = \frac{E - V_o - R_L \times i_L}{L} \quad (2.4)$$

By assuming duty cycle of control pulse to be  $D$  and its period to be  $T$ , the duration for the switch to be closed and opened are  $DT$  and  $(1-D)T$ , respectively. The average value of  $\frac{dV_o}{dt}$  can be obtained by multiplying Eqns (2.1) and (2.3) with their respective closed and opened switch duration and dividing the overall expression by  $T$ , i.e.

$$\frac{dV_o}{dt} = \frac{DT \times \left( \frac{-V_o}{RC} \right) + (1-D)T \times \frac{1}{c} \times \left( i_L - \frac{V_o}{R} \right)}{T} \quad (2.5)$$

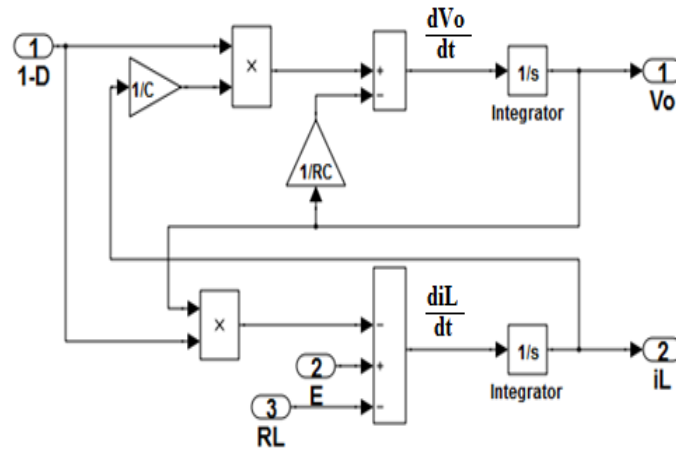
Simplifying (5) yields,

$$\frac{dV_o}{dt} = (1 - D) \frac{i_L}{c} - \frac{V_o}{R \times c} \quad (2.6)$$

Same operations on  $\frac{di_L}{dt}$  results in,

$$\frac{di_L}{dt} = \frac{E - R_L - V_o \times (1-D)}{L} \quad (2.7)$$

Using Eqns. (2.6) and (2.7), the average simulation model of the Boost converter is derived. The corresponding simulation model in Simulink is shown in Figure 2.3.



**Figure 2.3 Simulation model of Boost converter**

Next, the average model is perturbed by small signal perturbation around an operating point. If ( $\tilde{d}$ ) is the perturbation on duty cycle and ( $\tilde{V}_o$ ) is the resulting variation in the output voltage, the small signal model and transfer function of the converter can be obtained as follows [14].

$$\frac{\tilde{V}_o}{\tilde{d}} = \frac{E}{(1-D)^2} \left( 1 - \frac{L}{R} s \right) \frac{1}{LC(s^2 + \frac{s}{RC} + \frac{1}{LC})}, \text{ for } R_L=0 \quad (2.8)$$

The transfer function in (2.8) indicates that there is a zero in the right-half plane. This will affect the transient performance by limiting the control bandwidth. It should be noted that, narrow control bandwidth cause slow transient response. The transfer function also is dependent on steady state operating value D. Furthermore, it is can be seen from the same Eqn. that the zero and the poles values are load (R) dependent; different load value results in different poles and zero locations. Normally, a controller for Boost converter is designed based on a single (nominal) load value. However, when there exist large load variation (for example in step load change), most model-based controllers are not able to cope satisfactorily. To minimize the effect of load to the system, a load current sensing method has been proposed [15]. However, this solution requires an additional current sensor and more complicated control algorithm.

### 2.3 PID Controller

The proportional-integral (PI) controller is used to control DC-DC power converters in several researches [18],[19]. Main advantage of PI controller is simplicity of the implementation versus non-model base controllers.

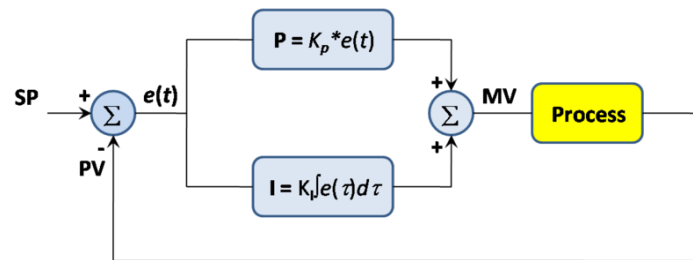
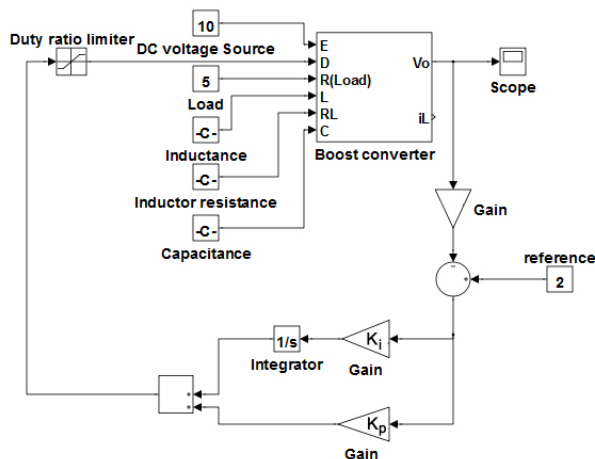


Figure 2.4 PI controller

Several methods are proposed to find appropriate values for  $K_p$  and  $K_i$ ; common point for all these approaches is dependency on zero and pole places directly or indirectly. As it was discussed in section 2.2 transfer function of Boost converter has poles dependent on load resistance and it varies with changes in duty cycle ( $D$ ); hence finding suitable values for  $K_i$  and  $K_p$  that can comply with all load values and working points is impossible.

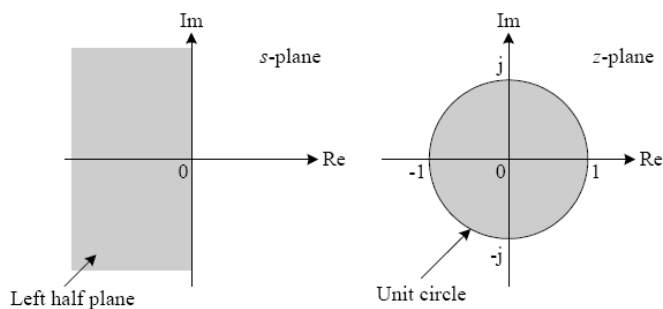
In order to show merit of SIFLC against PI controllers, Boost converter controlled by PI is simulated, its diagram is depicted in Figure 2.5.



**Figure 2.5 Simulink schematic of PI controlled Boost Converter**

### 2.4 Deadbeat Controller

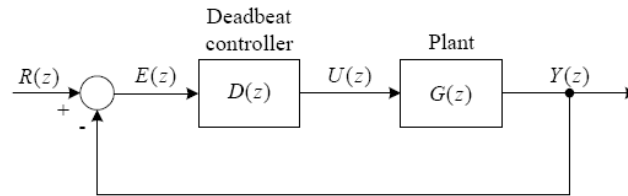
A deadbeat controller is a classical feedback controller where the control gains are set using a table based on the plant system order and normalized natural frequency. In this method all system poles should be placed in the left side of s-plane ( $s \rightarrow \infty$ ), it leads to fast dynamic response of the system. Setting system poles to infinity is possible by mapping analogue s-plane to discrete z-plane as it is illustrated in Figure 2.6.



**Figure 2.6 Mapping of s-plane to z-plane**



The deadbeat controller is used to control DC-DC power converters by Saggini and Stefanutti in [2] and by Bibian and Jin [19]. Typical diagram of a system controlled by this controller is shown in Figure 2.7.



**Figure 2.7 Typical deadbeat controllers**

The close loop transfer function can be written as:

$$\frac{Y(z)}{R(z)} = M(z) = \frac{D(z) \times G(z)}{1 + D(z) \times G(z)} \quad (2.9)$$

In which  $G(z)$  is the plant transfer function and  $D(z)$  is deadbeat controller, solving for  $D(z)$  from equation (2.9) yields:

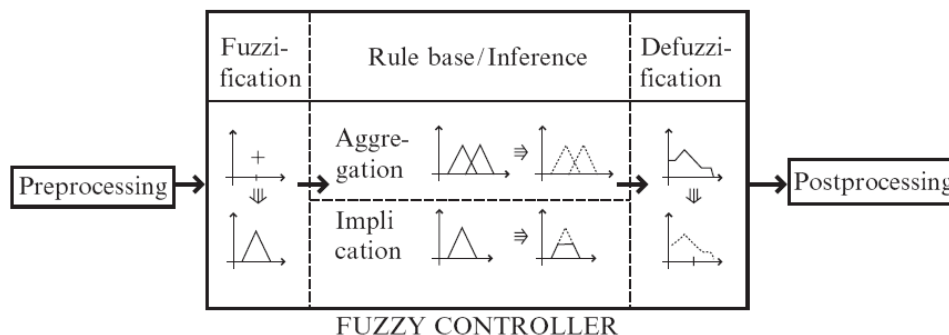
$$D(z) = \frac{1}{G(z)} \left( \frac{M(z)}{1 - M(z)} \right) \quad (2.10)$$

From (2.10) it will be concluded that design of deadbeat controller is to cancel out zero and poles of uncompensated system. This controller type has drawback of sensitivity to system parameter and strong dependency on poles location.

## 2.5 Conventional Fuzzy Logic Controller (CFLC)

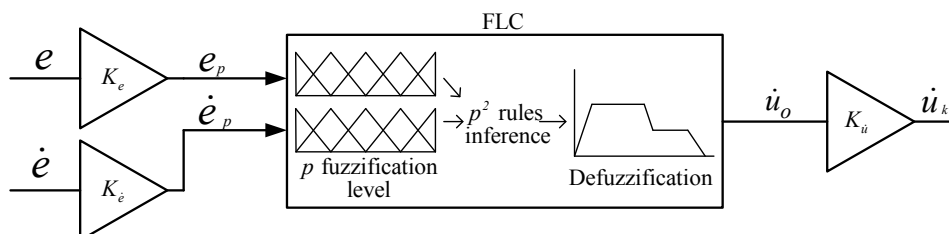
Figure 2.8 shows a schematical breakdown of a fuzzy controller [20]. As we can see, the fuzzy controller is preceded by a preprocessor and followed by a postprocessor block. The preprocessor usually is a device that makes crisp measurements, which

are most often numerical in nature, rather than linguistic. During the preprocessing, already some calculations are performed which have no real connection to the fuzzy control process, but nevertheless can yield a lot of influence.



**Figure 2.8 Fuzzy Logic Controller schematic**

Fuzzy logic controller has four main blocks, fuzzification, inference engine, rule base and defuzzification. General structure of FLC is shown in Figure 2.9.



**Figure 2.9 FLC overall structure**

**2.5.1 Fuzzification**

The first block inside the controller is *Fuzzification*, which converts each piece of input data to degrees of membership by a lookup in one or several membership functions. The fuzzification block thus matches the input data with the conditions of the rules to determine how well the condition of each rule matches that

particular input instance. There is a degree of membership for each linguistic term that applies to that input variable [21].

### **2.5.2 Rule base**

The basic function of the rule base is to represent the expert knowledge in a form of if-then rule structure. Four methods of deriving the rule base can be described as follows:

- (i) Expert experience and control engineering knowledge
- (ii) Based on operator's control actions
- (iii) Based on fuzzy model of a process
- (iv) Based on learning

#### **(i) Expert experience and control engineering knowledge:**

This method is the least structured of the four methods and yet it is one of the most widely used today. It is based on the derivation of rules from the experience based knowledge of the process operator and/or control engineer.

#### **(ii) Based on operator's control actions:**

This method tries to model an operator's skilled actions or control behavior in terms of fuzzy implications using the input-output data connected with his control actions. The idea behind this technique is that it is easier to model an operator's actions than to model a process, since the input variables of the model are likely found by asking the operator what kind of information is used in control actions.

**(iii) Based on fuzzy model of a process:**

In the linguistic approach, the linguistic description of the dynamic characteristics of a controlled process may be viewed as a fuzzy model of the process. Based on the fuzzy model, we can generate a set of fuzzy control rules for attaining optimal performance of a dynamic system. The set of fuzzy control rules forms the rule base of the fuzzy logic controller. Although this approach is somewhat more complicated, it yields better performance and reliability and provides a more tractable structure for dealing theoretically with the fuzzy logic controller.

**(iv) Based on fuzzy model of a process:**

In the linguistic approach, the linguistic description of the dynamic characteristics of a controlled process may be viewed as a fuzzy model of the process. Based on the fuzzy model, we can generate a set of fuzzy control rules for attaining optimal performance of a dynamic system. The set of fuzzy control rules forms the rule base of the fuzzy logic controller. Although this approach is somewhat more complicated, it yields better performance and reliability and provides a more tractable structure for dealing theoretically with the fuzzy logic controller.

### **2.5.3 Defuzzification**

Defuzzification is a mapping from a space of fuzzy control actions defined over an output universe of discourse into a space of non-fuzzy (crisp) control action. This process is necessary because in many practical applications crisp control action is required to actuate the plant.

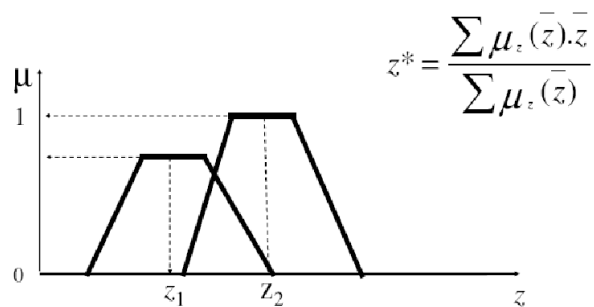
The most common fuzzification methods are as follows:

- (i) The most popular method is the *Centre of Gravity method* (CoG), which is described by the equation:

$$u = \frac{\sum_{i=1}^n \mu(x_i)x_i}{\sum_{i=1}^n \mu(x_i)} \quad (2.11)$$

In (2.11) variable  $n$  represents the number of output membership function while  $\mu(x_i)$  is the degree of membership function and  $x_i$  is the peak location of the membership function.

- (ii) Another method is *Weight average method*, which is only valid for symmetrical output membership functions. The weight average method is formed by weighting each membership function in the output by its respective maximum membership value,  $z$ .



**Figure 2.10 Weight average method defuzzification**

## CHAPTER 3

### SINGLE INPUT FUZZY LOGIC CONTROLLER, PIECEWISE LINEAR CONTROL SURFACE AND CONTROLLER DESIGN

#### 3.1 Introduction

Conventional fuzzy logic controllers have two inputs mainly, error and derivative of error; also rule table and control surface are 2-D and 3-D respectively. Having 2 inputs for CFLC leads to complicated design in which all components of FLC must be implemented in the processor. Implementation of such a system has two main drawbacks, one in design stage and another one in execution stage. In fact designing a complete CFLC needs complex programming algorithm, in the case of using FPGA, system program size will be massive. [22]. In the execution stage having heavy computations of control algorithm reduces control action time and leads to poor transient response.

The idea of single input fuzzy logic controller (SIFLC) was proposed by Choi [11], in which SIFLC was applied to a pendulum system, he showed that results using CFLC and SIFLC are similar. Conditions in which CFLC can have identical SIFLC are: 1- Triangular membership function, 2- Toeplitz rule table. Any FLC which have these requirements can be represented by SIFLC.

Fortunately most power electronic converters have Toeplitz rule table, it means they are prone to be controlled by SIFLC. Using SIFLC in controlling power converters instead of CFLC has several advantages like: lower cost of controller, higher switching frequency, lower inductor size and more simple control algorithm.

### 3.2 Single distance method

Typically, FLC has two controlled inputs, namely error ( $e$ ) and the change of error ( $\dot{e}$ ). Its rule table can be created on a two-dimensional space of the phase-plane as shown in Table 3.1. Typically the rule table has the same output membership in a diagonal direction. Additionally, each point on the particular diagonal lines has a magnitude that is proportional to the distance from its main diagonal line  $L_Z$ . This is known as the Toeplitz structure [11].

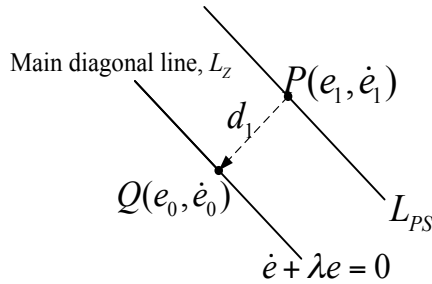
**Table 3.1 FLC Rule table with Toeplitz structure**

$\dot{e}$	$e$	PL	PM	PS	Z	NS	NM	NL
NL	Z	NS	NM	NL	NL	NL	NL	NL
NM	PS	Z	NS	NM	NL	NL	NL	NL
NS	PM	PS	Z	NS	NM	NL	NL	NL
Z	PL	PM	PS	Z	NS	NM	NL	NL
PS	PL	PL	PM	PS	Z	NS	NM	NL
PM	PL	PL	PL	PM	PS	Z	NS	NL
PL	PL	PL	PL	PL	PM	PS	Z	NL

In Table 3.1, instead of using two-variable input sets ( $e, \dot{e}$ ), it is possible to obtain the corresponding output,  $u_{\phi}$  using a single variable,  $d$ . It represents the absolute distance magnitude of the parallel diagonal lines from the main diagonal line  $L_Z$ . To derive  $d$ , let  $Q(e_0, \dot{e}_0)$  be an intersection point of the main diagonal line and the line

perpendicular to it from a known operating point  $P$ . It can be noted that the main diagonal line can be represented as a straight line function, i.e.

$$\dot{e} + \lambda e = 0 \tag{3.1}$$



**Figure 3.1 Derivation of distance variable,  $d$**

The distance  $d$  from point  $P(e_1, \dot{e}_1)$  to point  $Q(e_0, \dot{e}_0)$ , can be formulated as

$$d = \frac{\dot{e} + \lambda e}{\sqrt{1 + \lambda^2}} \tag{3.2}$$

The derivation of  $d$  resulted in a one-dimensional rule table, depicted in Table 3.2, in which  $L_{NL}$ ,  $L_{NM}$ ,  $L_{NS}$ ,  $L_Z$ ,  $L_{PS}$ ,  $L_{PM}$  and  $L_{PL}$  are the diagonal lines of Table 3.1. These diagonal lines correspond to the new input to Table 3.2, while NL, NM, NS, Z, PS, PM and PL represent the output of corresponding diagonal lines. The control action of using this table is now exclusively determined by a single input variable  $d$ . It is therefore appropriate to call it the Single Input FLC (SIFLC).

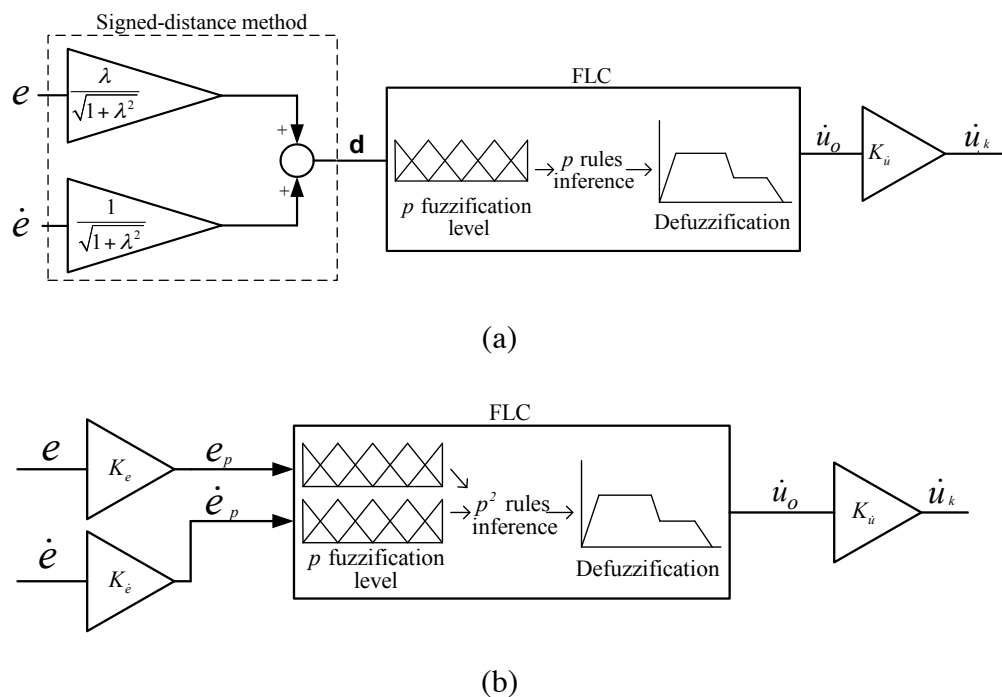
**Table 3.2 The reduced SIFLC rule table using the Signed distance method**

$d$	$L_{NL}$	$L_{NM}$	$L_{NS}$	$L_Z$	$L_{PS}$	$L_{PM}$	$L_{PL}$
$\dot{e}$	NL	NM	NS	Z	PS	PM	PL

The overall structure of SIFLC based on the signed distance method and its corresponding rule table can be shown as a block diagram in Figure 3.2.a. The input



to the Fuzzy block is  $d$ , while its output is the change of control output,  $\dot{u}_k$ . The final output is obtained by multiplying  $\dot{u}_k$  with an output scaling factor, denoted as  $K_u$ . For comparison, the structure of the CFLC is shown in Figure 3.2.b. It has two inputs to its Fuzzy block. Clearly, the main feature of SIFLC is the significant reduction in the number of rules. For a two input CFLC with fuzzification level  $p$ , the number of rules to be inferred is  $p^2$ . An equivalent SIFLC requires only  $p$  rules.

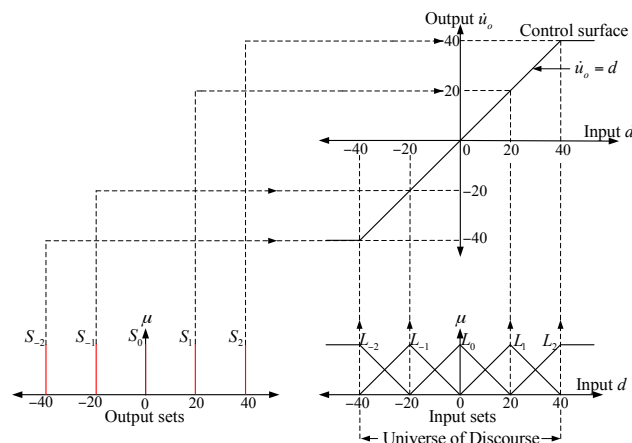


**Figure 3.2 Controller structure, (a) SIFLC, (b) CFLC**

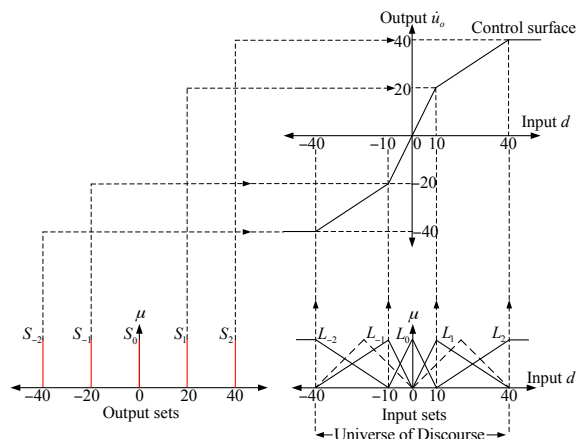
### 3.3 SIFLC Control Surface

With the rule table reduced to a one-dimensional array, it is possible to approximate the control surface of SIFLC as a one dimensional piecewise linear (PWL). This can be achieved with the following conditions: (a) the input membership function (MF) is triangular shape (b) the output membership function is singleton (c) the fuzzification and defuzzification process uses Center of Gravity (CoG) method [13]. The PWL control surface can be simply constructed using look-up table, resulting in much faster computational time.

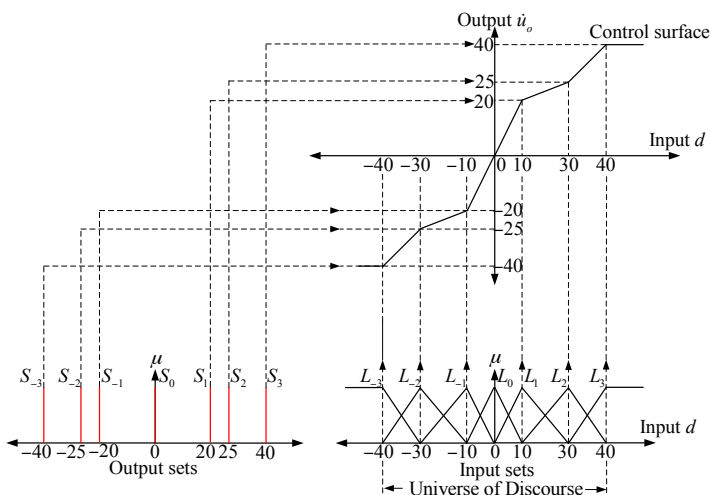
Figure 3.3 shows an example of a PWL control surface which has a constant slope throughout the Universe of Discourse (UoD). This is called the SIFLC with *symmetrical* MF. Figure 3.4 depicts a PWL surface constructed when the peak location of the input and output MFs are arranged in unequal spaces. Note that it results in a multiple regions PWL with linear lines of different slopes. It also introduces break-point ( $d_{bp}$ ), which is defined as the transition point between two piecewise linear slopes. This type is defined as the SIFLC with *asymmetrical* MF. Alternatively, the asymmetrical type SIFLC can be constructed by changing the location of the singleton output MF. To obtain more piecewise linear regions on the control surface, more MFs are required, as depicted in Figure 3.5. The additional asymmetric MFs result in more piecewise linear regions, with additional break-points being created.



**Figure 3.3 PWL Control surface with symmetrical input and output membership functions**



**Figure 3.4 PWL control surface with peak locations of the asymmetrical input MF**



**Figure 3.5 PWL control surface with more piecewise linear regions using more input and output asymmetrical MFs**

By reducing the control surface to a PWL, the control block of the SIFLC can be represented as in Figure 3.6. This structure is very simple and allows for very rapid computation. The fuzzification, rule inference and defuzzification processes are no longer required.

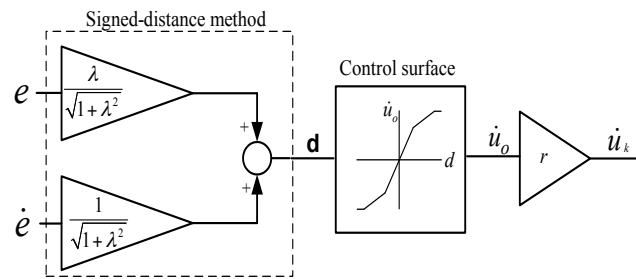


Figure 3.6 SIFLC with PWL Control Surface

### 3.4 Fuzzy logic controller design

#### 3.4.1 Using the Conventional FLC (CFLC)

The Simulink model of boost converter with CFLC is depicted in Figure 3.7. The averaged model of the Boost converter derived in previous Section is included. Two parameters are applied to the CFLC input, namely the error (ERROR) and derivative of error (D-ERROR). The controlled parameter is output voltage. The output of CFLC is the change in duty cycle (D); hence an integrator is needed to generate D. The duty ratio limiter is required limit  $D < 0.8$ , to avoid converter's instability [16].

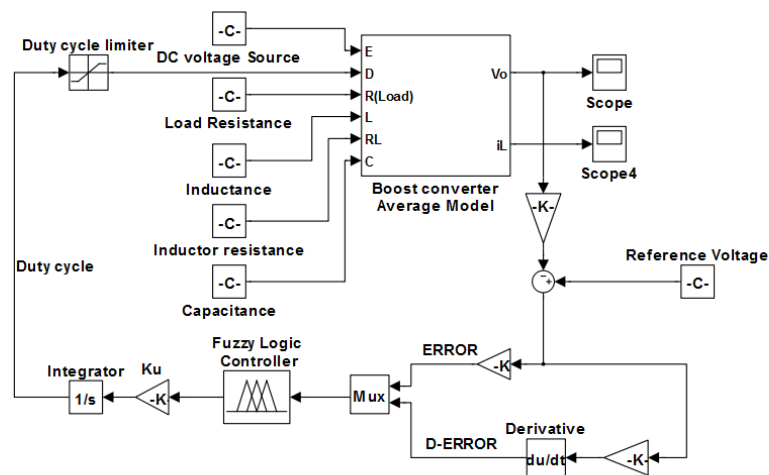
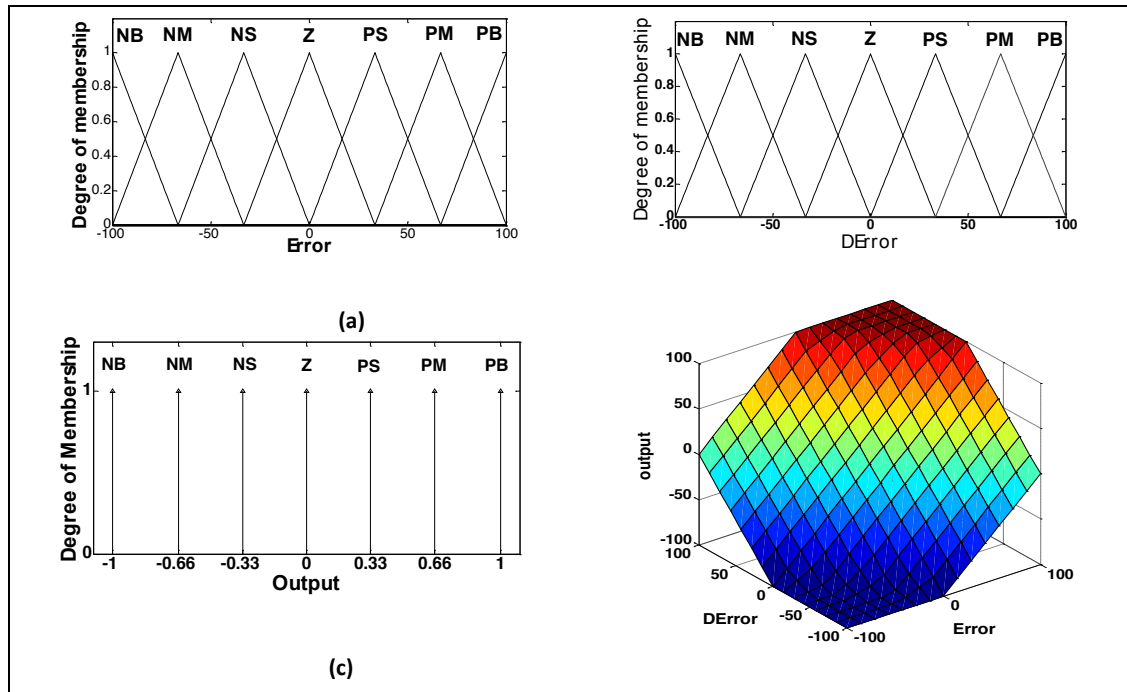


Figure 3.7 Simulink simulation model for Boost converter with CFLC

The input and output MFs of the CFLC are illustrated in Figure 3.8 (a) to (c). It is a Sugeno type with its input MFs equally spaced with 50% overlapping. The output MF is singleton. The choice of output MF is primarily for increased computational

speed. The inference rules, shown in Table 3.3, are designed and optimized heuristically using the MATLAB Fuzzy Toolbox. The corresponding three dimensional control surface is plotted in Figure 3.8 (d).

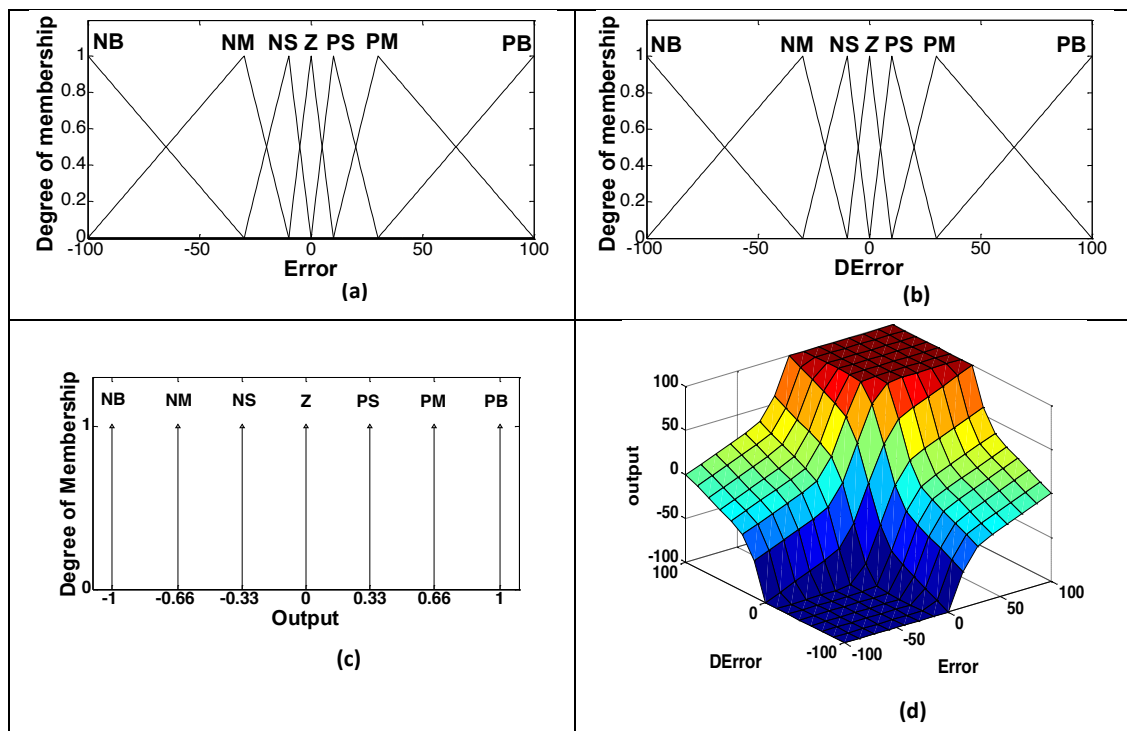


**Figure 3.8 Symmetric CFLC (a) Error MFs (b) Change of error MFs (c) Output MFs (d) Control Surface associated with FLC**

**Table 3.3 Rule table for CFLC**

$e$	NB	NM	NS	Z	PS	PM	PB
$\&$							
NB	-100	-100	-100	-100	-66	-33	0
NM	-100	-100	-100	-66	-33	0	33
NS	-100	-100	-66	-33	0	33	66
Z	-100	-66	-33	0	33	66	100
PS	-66	-33	0	33	66	100	100
PM	-33	0	33	66	100	100	100
PB	0	33	66	100	100	100	100

The CFLC designed in Figure 3.8 has symmetrical distribution MFs. However, if the overlaps are not equal, as illustrated in Figure 3.9, CFLC with asymmetrical MF will result. The control surface of the CFLC with asymmetrical input MFs is depicted in Figure 3.9(d). Output MF remains singleton, similar to Figure 3.9 (c).



**Figure 3.9 Asymmetrical CLFC, (a) Error MFs, (b) Change of Error MFs, (c) Output MFs, (d) Control surface associated with FLC**

### 3.4.2 Using Single Input Fuzzy Controller (SIFLC)

Figure 3.10 shows the Simulink model for the Boost converter controlled by the SIFLC. The Fuzzy controller block is replaced by a simple PWL look up table. The rule table for the SIFLC is shown in Table 3.4, which corresponds to the CFLC rule table of Table 3.3. The PWL control surface for this rule table is shown in Figure 3.11 (a). It is the SIFLC with symmetric MF.

By manipulating the input MF, i.e. by making them unequally spaced, the same rule table can be used to produce SIFLC with asymmetrical MF. The PWL control surface for this is shown in Figure 3.11 (b). It has three PWL sections with

highest slope for  $0 < d < 10$ . Then the slope is reduced after each subsequent breakpoint (BP1, BP2). Beyond  $U_{oD}$  ( $d > 100$ ), saturation limit is imposed.

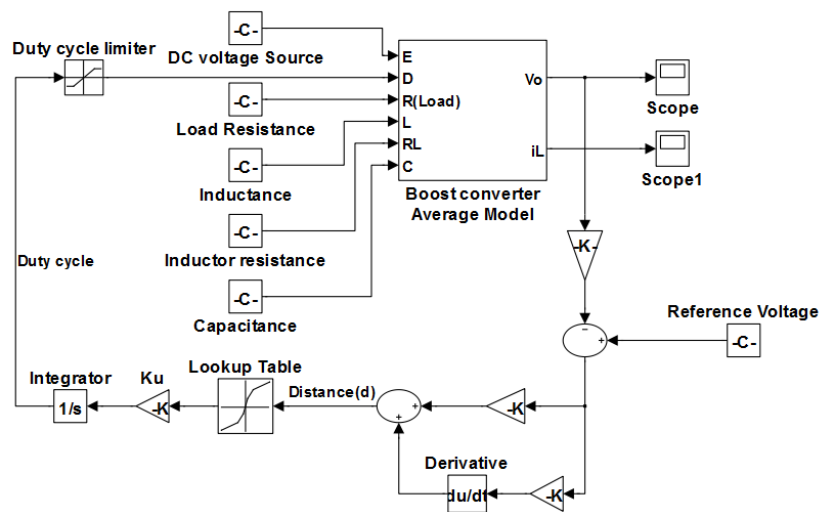
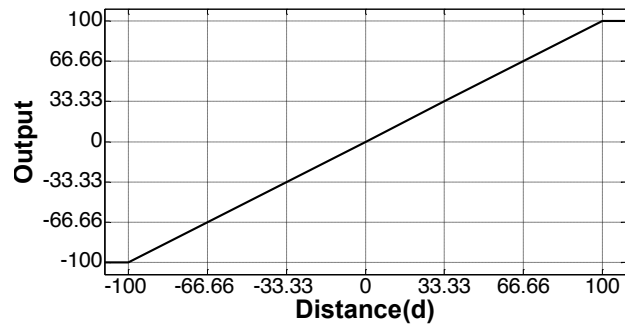


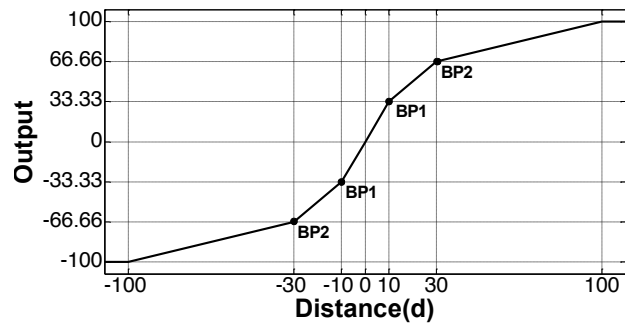
Figure 3.10 Simulink simulation model for Boost converter with SIFLC

Table 3.4 Rule table for SIFLC

Distance ( $d$ )	NB	NM	NS	Z	PS	PM	PB
Output	-100	-66.66	-33.33	0	33.33	66.66	100



(a)



(b)

**Figure 3.11 PWL Control surface for SIFLC (a) with symmetrical input MFs, (b) with asymmetrical input MFs**



## CHAPTER 4

### SIMULATION RESULTS

#### 4.1 INTRODUCTION

In this section, Boost converter is simulated using Simulink schematics described in 3.4.1 and 3.4.2. In order to investigate performance of CFLC and SIFLC converter is tackled with three disturbances:

- Step change in input voltage
- Step change in reference voltage
- Step change in load resistance

These disturbances will appear in real applications; for instance if the input of converter is connected a solar cell, voltage of cell is prone to sudden changes due to shading. Also if the converter is used to drive DC motor it is subjected variations in reference value to reach new speed value. In addition having non-linear loads in the output of converters, increase the possibility of happening sudden changes in load value.

In the simulation results firstly, comparison of CFLC and SFLC for same disturbances is illustrated; in this part both symmetrical and asymmetrical MF's are applied. Secondly SIFLC is compared with PI controller; PI parameters are similar to

SIFLC parameters by assuming  $K_p=K_{DE}$  and  $K_i=K_E$  in which  $K_p$ ,  $K_i$ ,  $K_{DE}$  and  $K_E$  are proportional gain, integrator gain, (PI) and derivative of error gain, error gain(SIFLC) respectively. Eventually, CPU time consumed by SIFLC and CFLC are compared for same conditions and same simulation profile.

## 4.2 Boost converter parameters

The Boost converter is designed with the following parameters:

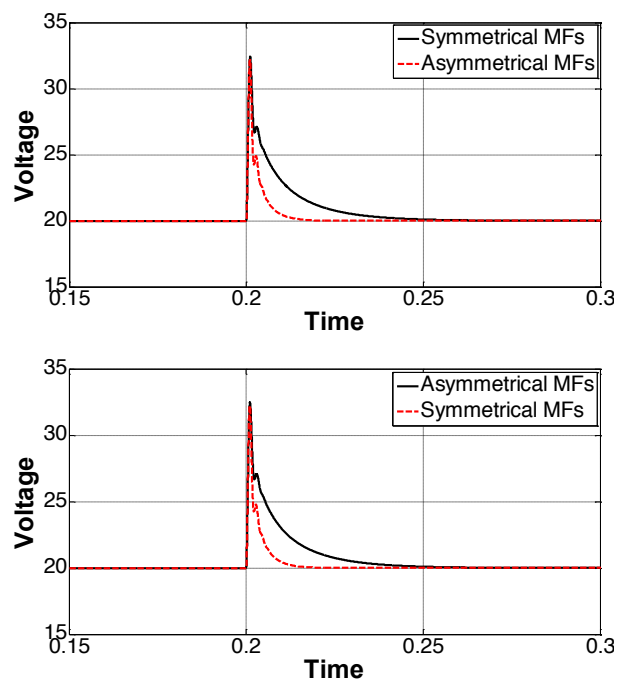
Input Voltage	$V_i=10V$
Output Voltage (nominal)	$V_o=20V$
Nominal load	$R=5\Omega$
Switching frequency	$f_s=100\text{ KHz}$
Filter Capacitor	$C=100\mu F$
Filter Inductor	$L=250\mu H$

## 4.3 Comparison of SFILC and CFLC

For MATLAB simulation, the Boost average model as described in Section 2.2 is used. In this case, the switching frequency is irrelevant. It is only used in the design process to determine the values of L and C.

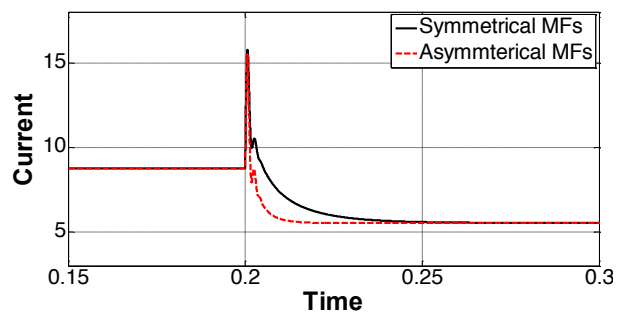
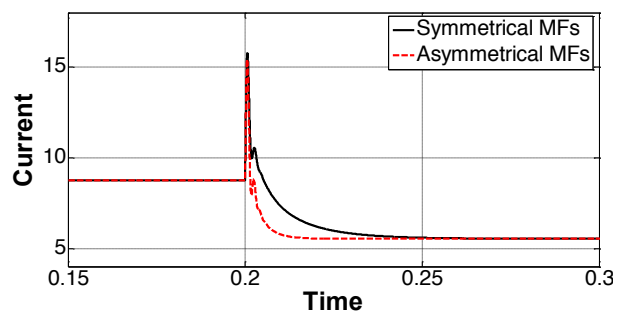
These disturbances most commonly found in power electronics converters are applied to the converter, namely (a) step change in reference voltage, (b) step change in input voltage, (c) step change in load resistance. The performances of CFLC and SIFLC controllers are compared. Note that the objective of the simulation is to show that these two controllers have identical control characteristics, albeit the marked difference in their control structures. The simulation is not designed to optimize the Boost converter controller performance.

Figure 4.1(a) shows the output voltage response of the converter when the input voltage is stepped from 10 to 15V using CFLC controller. The load resistor (therefore the output power) is maintained constant. From the response, it can be seen that after 50 msec, the output voltage is restored to its original value. It can also be observed in Figure 4.1(a) that the response for CFLC with asymmetrical MF is better than the symmetrical MF in terms of overshoots and settling times. Figure 4.1(b) shows the response using SIFLC. Clearly the response is almost identical to CFLC for both the asymmetrical MF and symmetrical MF cases. It can be concluded that the CFLC controller can be replaced by SIFLC without degrading the performance of the former. The inductor current responses for both types of CFLC and SIFLC are depicted in Figure 4.2(a) and (b), respectively.

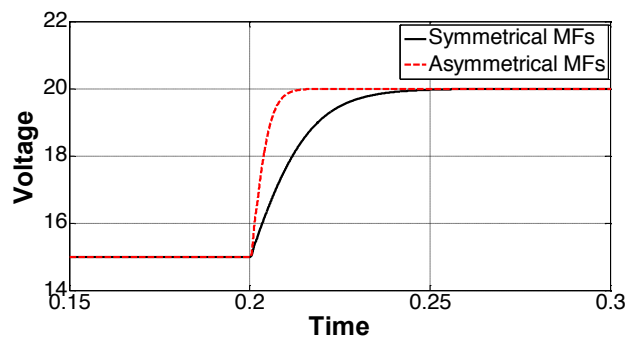
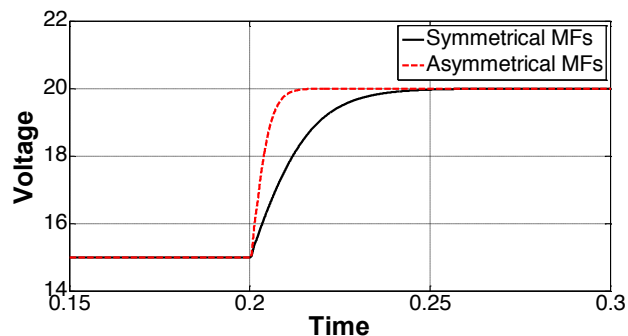


**Figure 4.1 Output voltage responses for step change in input voltage from 10V to 15V (a) CFLC (b) SIFLC**

(b)  
Figure 4.3 through 4.6 show the response of CFLC and SIFLC for step change in reference voltage and step load changes. As can be observed, the responses of both controllers are almost identical.

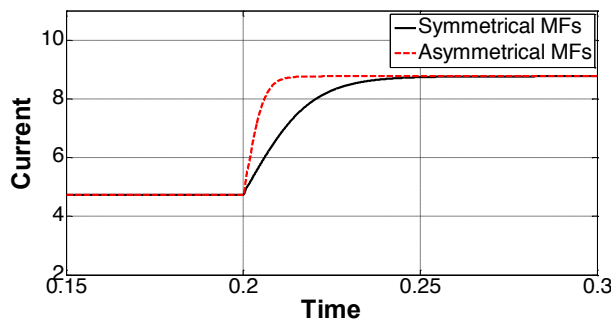


**Figure 4.2 Inductor current responses for step change in input voltage from 10V to 15V, (a) CFLC, (b) SIFLC**



**Figure 4.3 Output voltage response for step change in reference voltage from 15V to 20V when input voltage  $V_i=10V$ . (a) CFLC, (b) SIFLC**

(b)



(a)

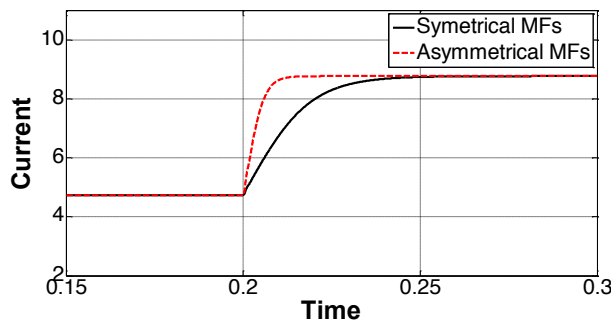
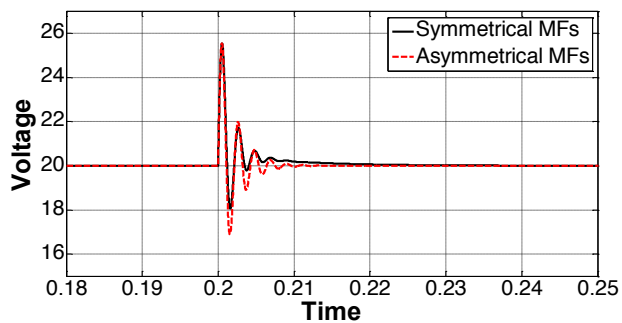
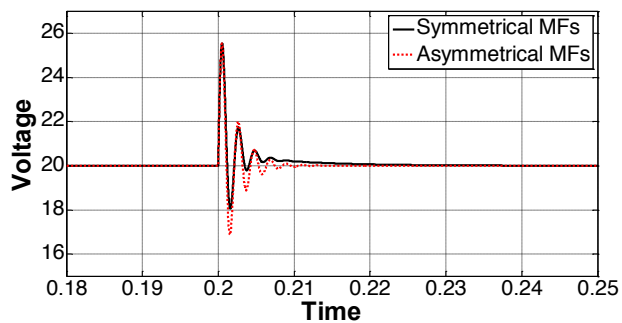
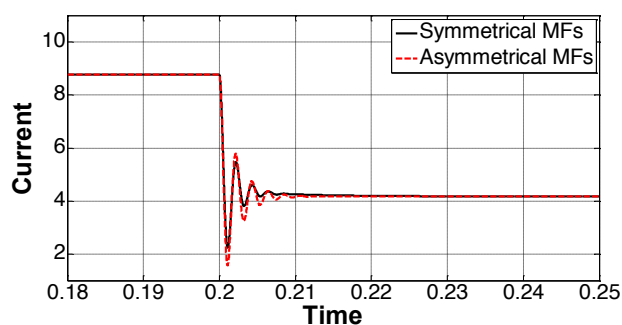
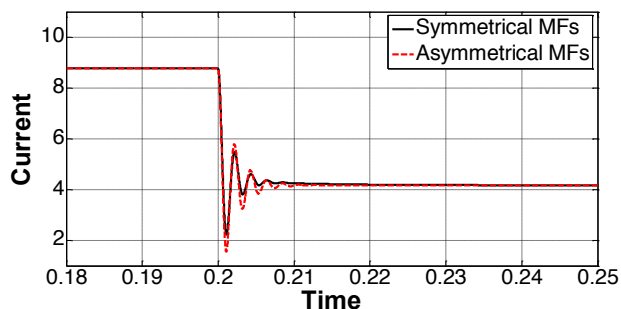


Figure 4.4 Inductor current response for <sup>(b)</sup>step change in reference voltage from 15V to 20V when the input voltage  $V_i=10V$ . (a) CFLC (b) SIFLC



**Figure 4.5 Voltage response for step change in load resistance from  $5\Omega$  to  $10\Omega$  when the input voltage  $V_i=10v$ . (a) CFLC (b) SIFLC**

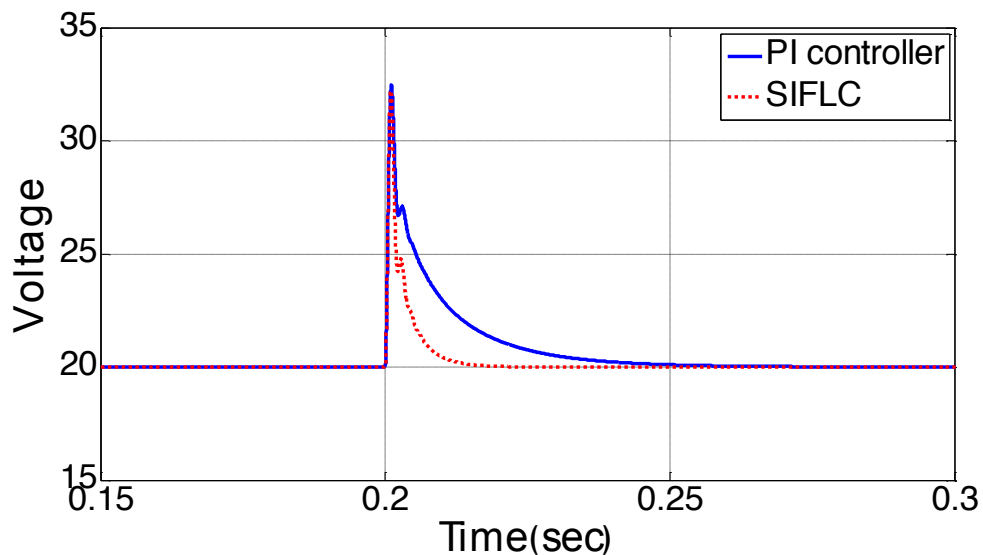


**Figure 4.6 Inductor current response for step change in load resistance from  $5\Omega$  to  $10\Omega$  when the input voltage  $V_i=10V$ . (a) CFLC, (b) SIFLC**

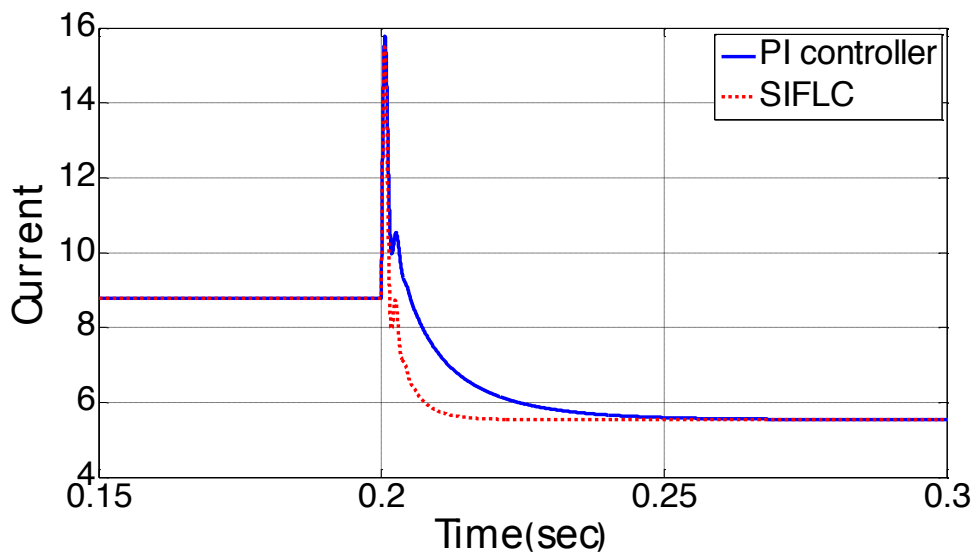
#### 4.4 Comparison between SIFLC and PI controller

In this section performance of PI controller and SIFLC (with asymmetrical MFs) are compared. PI parameters are chosen similar to SIFLC parameters by assuming  $K_p=K_{DE}$  and  $K_i=K_E$  in which  $K_p$ ,  $K_i$ ,  $K_{DE}$  and  $K_E$  are proportional gain, integrator gain, (PI) and derivative of error gain, error gain<sup>(a)</sup>(SIFLC) respectively.

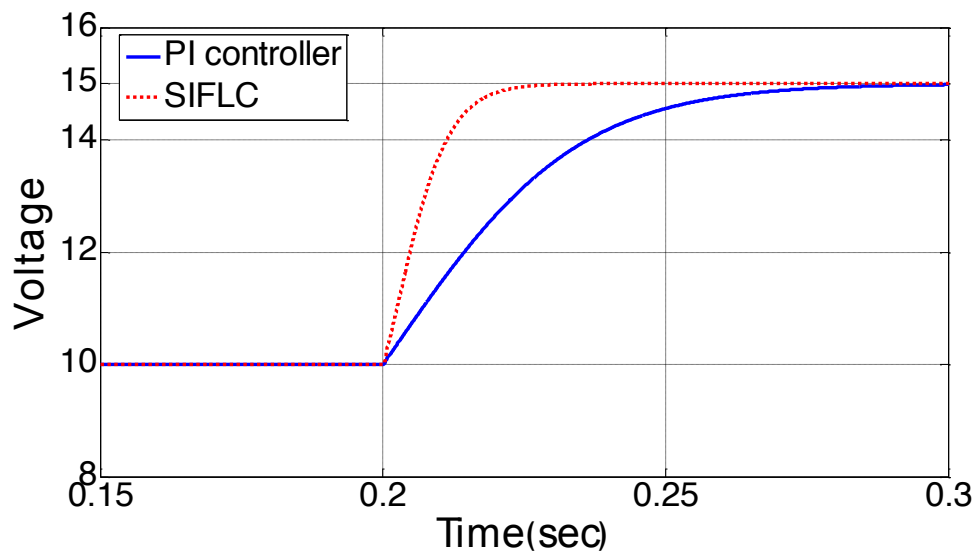
(b)



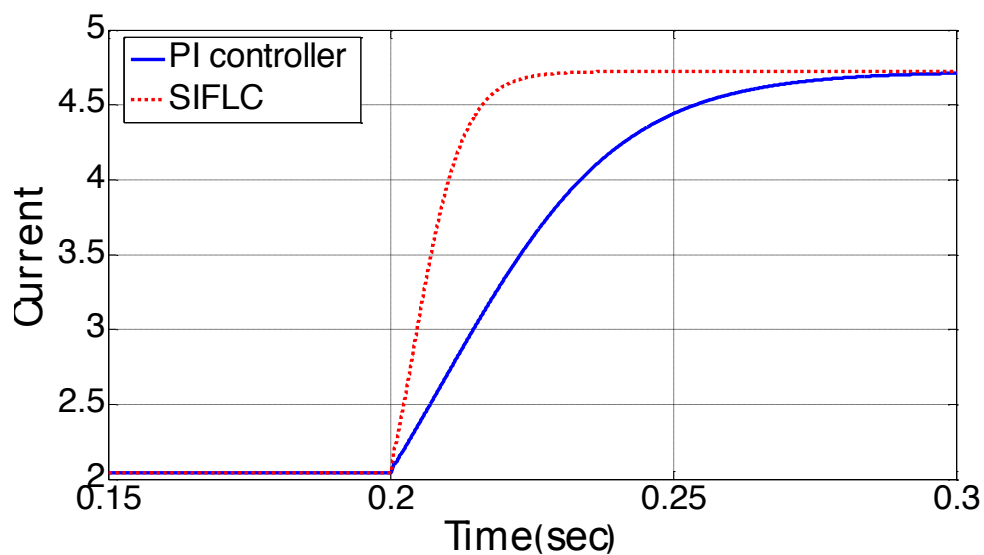
**Figure 4.7 Output voltage of Boost converter for step change in input voltage from 10v to 15v in  $t=0.2$  and  $R=5\Omega$**



**Figure 4.8 Inductor current of Boost converter for step change in input voltage from 10v to 15v in  $t=0.2$  and  $R=5\Omega$**

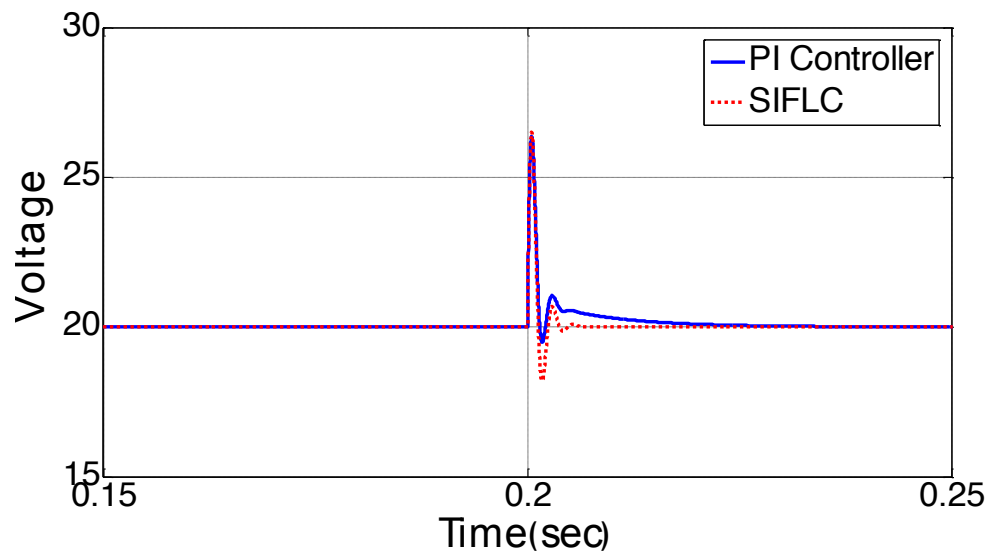


**Figure 4.9** Output voltage of Boost converter for step change in reference voltage from 10v to 15v in  $t=0.2$  and  $R=5\Omega$ ,  $V_i=10$

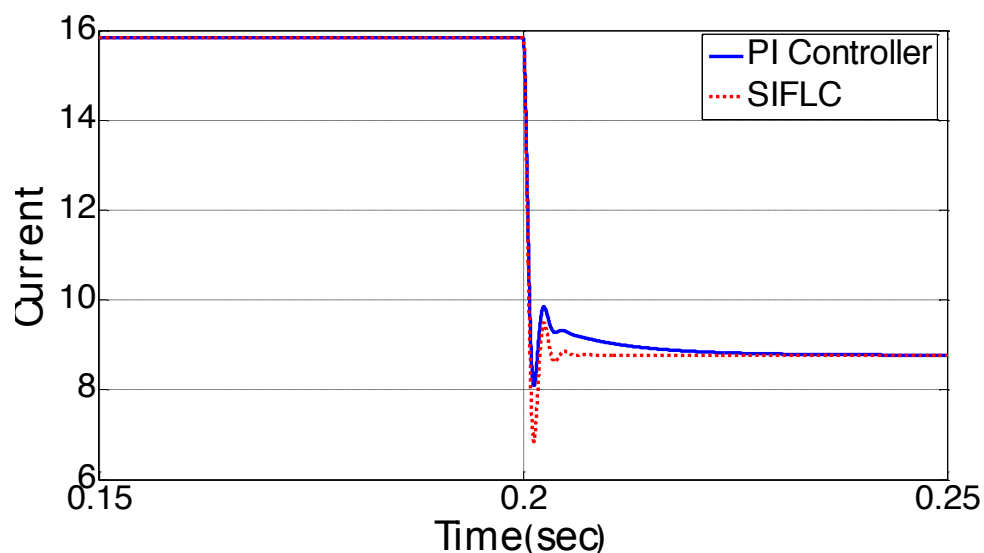


**Figure 4.10** Inductor current of Boost converter for step change in reference voltage from 10v to 15v in  $t=0.2$  and  $R=5\Omega$ ,  $V_i=10$





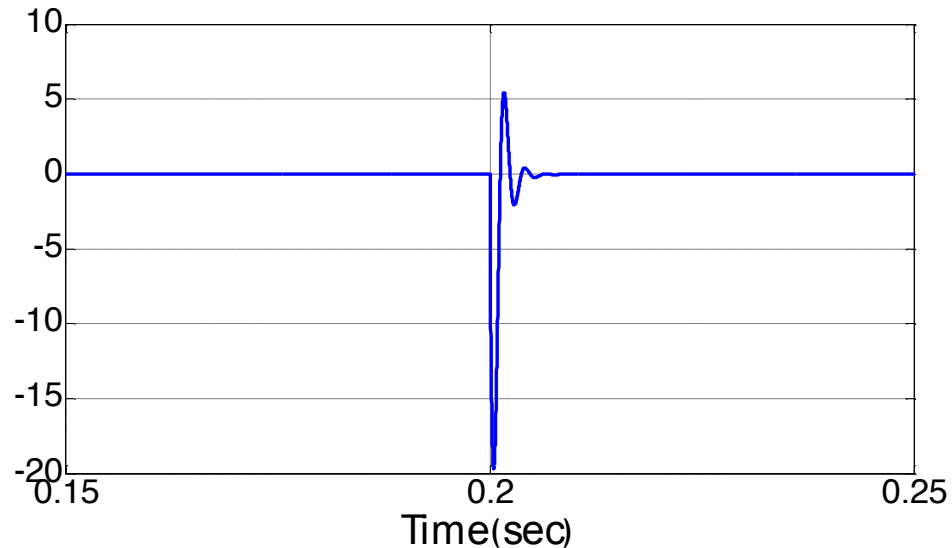
**Figure 4.11** Output voltage of Boost converter for step change in load value from  $3 \Omega$  to  $5 \Omega$  in  $t=0.2$  and  $V_{ref}=20v$ ,  $V_i=10$



**Figure 4.12** Inductor current of Boost converter for step change in load value from  $3 \Omega$  to  $5 \Omega$  in  $t=0.2$  and  $V_{ref}=20v$ ,  $V_i=10$

Investigation of Figures 4.7 through 4.12 show that SIFLC with asymmetrical input MFs is faster in all cases, although it has higher overshoot and undershoot for step change in load response. Reason for higher resonance values in SIFLC can be justified by exceeding input of PWL control surface from first BP in Figure 3.13 (b), as it is depicted in Figure 4.13 input of PWL controller surface reaches to  $-20$  which means it exceed from first PB which is  $-10$ , but for positive section it rises to  $5$  that

do not go above of first PB which 10; that is why value of overshoot is much lower compare to undershoot as it is shown in Figure 4.11.



**Figure 4.13 Input of PWL control surface of SIFLC for step change in load value from  $3 \Omega$  to  $5 \Omega$  in  $t=0.2$  and  $V_{ref}=20v$ ,  $V_i=10$**

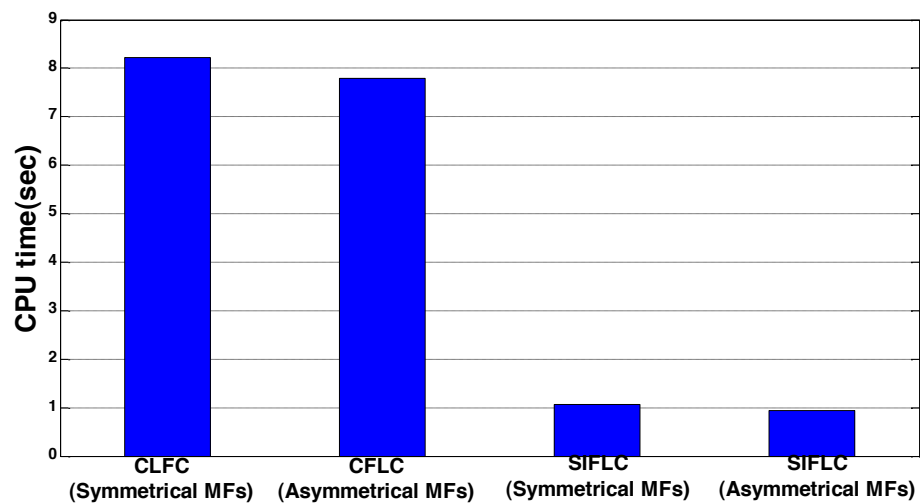
#### 4.5 Computational Comparison

Figure 4.14 compares the time taken to execute the controller's algorithm in MATLAB. The suitable benchmark is the total CPU run-time used by the Simulink program [17]. For each simulation run the controller's parameters and Simulink simulation profile is maintained equal for both CFLC and SIFLC. Ten sets of simulation are performed and the average results are shown in Fig. 21.

As can be observed, the SIFLC requires much shorter time to execute its algorithm. For the CFLC with symmetrical MF, the CPU run-time is 8.2 sec. In comparison, SIFLC requires only 1.1sec – a reduction of almost 8 times. Similar trend is

observed for the asymmetrical case. The run-times for CFLC and SIFLC are 7.8sec and 0.9sec, respectively.

The SIMULINK CPU run-time tests conclude that SIFLC requires nearly one order of magnitude less time compared to the CFLC, with almost no difference in its performance. This conclusion indicates that for hardware implementation, it is possible that SIFLC can achieve higher control bandwidth for a given converter switching frequency. Alternatively, the controller can be implemented using a slower processor.



**Figure 4.14 CPU time consumed by CFLC and SIFLC**

## CHAPTER 5

### HARDWARE IMPLEMENTATION AND EXPERIMENTAL RESULTS

#### 5.1 Introduction

Implemented hardware has two parts, one is the Boost converter circuit and another part is control circuit. The converter includes power circuit and driver circuit; also ADC comparator is embedded in this part. The control circuit is implemented in FPGA; it generates control pulse for power switch in converter to maintain output voltage in desired value.

The PI controller and SIFLC are designed inside of FPGA. One toggle switch changes the controller type. The objective of hardware implementation is to compare performance of SIFLC and PI controller; beside of that new ADC is implemented which is faster compare to conventional ADC's also it is superior in term of wiring area and cost. Altera DE2 board is chosen for implementation of the controller. It is educational board suitable for research projects and academic purposes. The FPGA uses fundamental logic gates to build digital systems; here we use QUARTUS software and Verilog language to program FPGA.

Main advantage of using FPGA versus DSP is flexibility of hardware for parallel processing. In FPGA it is possible to do several tasks at the same time; this leads to faster execution of algorithm thus controller has more ability to control system dynamics.

## **5.2 Controller circuit**

### **5.2.1 System schematic**

In digital system design, system algorithm must be divided to several blocks in which every block has specific function, these blocks can work in serial or parallel with other blocks depend on system algorithm. In this design whole system is separated to nine blocks including:

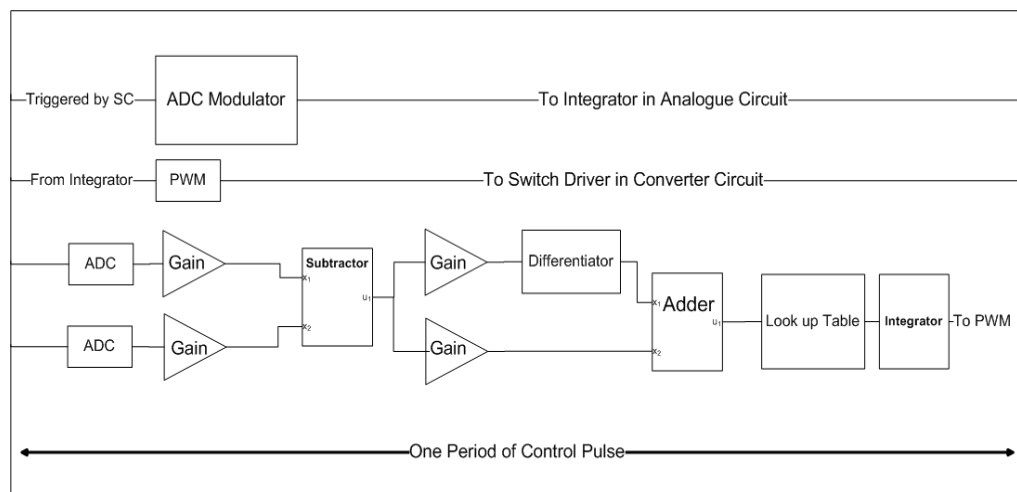
- **Gain**  
Acts as multiplier for E and DE coefficients.
- **ADC**  
Analogue to Digital Converter, it is simply a counter here.
- **Integrator**  
Integrates input data, it is used after the PWL controller.
- **Differentiator**  
It is used for getting digital derivative of data.
- **Adder**  
It adds data, mainly used in adding E and DE to feed to PWL controller.
- **Subtractor**  
It is used to generate error data by subtracting actual value from reference value.
- **Look up table**  
PWL controller is implemented by this block.
- **PWM**  
It generates PWM signal to control the switch in the converter.
- **ADC modulator**

This block creates square wave signal which is needed for analogue integrator to generate saw tooth wave form for comparator.

Digital controller schematic is illustrated in Figure 5.1, as it indicates three parallel tasks are done in one period of control pulse. Principle of system work is as following steps:

- (i) ADC modulator creates the modulating signal for external integrator to start digitization.
- (ii) Data of external comparator come to ADC, after maximum 0.8 of control pulse period data is ready for next block which is gain, from this point it takes utmost 34 clock pulse of digital system to generate data for PWM block.
- (iii) After readiness of data in previous stage PWM block generates control pulse with predetermined frequency (here 100 KHz) and duty cycle obtained from last stage.

It should be considered that although these above stages are sequential, but all the stages are working at the same time; by the other words they are one or two period of control pulse after or before of each other.

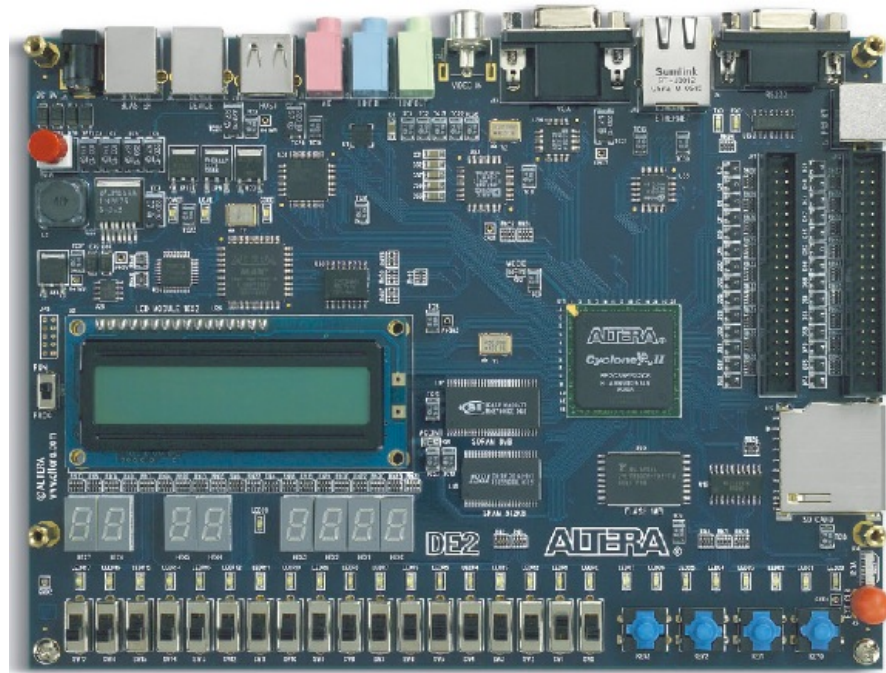


**Figure 5.1 Digital controller schematic**

The system clock of DE2 board is 50 MHz . If control pulse frequency is chosen to be 100 KHz, whole control algorithm can takes  $\frac{10\mu}{20n} = 500$  clock pulse (CP) to execute. The most time consuming block inside of the control algorithm is ADC, as it was cited before it can occupy 0.8 of whole control pulse period; thus for the other parts after ADC in Figure 5.1 we have  $500 - 500 \times 0.8 = 100$  CP time to finish the algorithm, this is plenty of time which is quite enough for the rest of blocks to prepare their data. Also it must be considered that 400 CP is worst case to time be taken by ADC.

### 5.2.2 DE2 Board

The DE2 board features a state-of-the-art Cyclone® II 2C35 FPGA in a 672-pin package. All chief components on the board are connected to pins of this chip, allowing the user to control all aspects of the board's operation. For simple experiments, the DE2 board includes a sufficient number of robust switches (of both toggle and push-button type), LEDs, and 7-segment displays. For more complex experiments, there are SRAM, SDRAM, and Flash memory chips, as well as a 16 x 2 character display. For experiments that require a processor and simple I/O interfaces, it is easy to instantiate Altera's Nios II processor and use interface standards such as RS-232 and PS/2. For experiments that deal to sound or video signals, there are standard connectors for microphone, line-in, line-out (24-bit audio CODEC), video-in (TV Decoder), and VGA (10-bit DAC); these features can be used to create CD-quality audio applications and professional-looking video. For larger design projects the DE2 provides USB 2.0 connectivity (both host and device), 10/100 Ethernet, an infrared (IrDA) port, and an SD memory card connector. Finally, it is possible to connect other user defined boards to the DE2 board by means of two expansion headers.[23]



**Figure 5.2 Altera DE2 Educational board**

### 5.3 The Proposed ADC

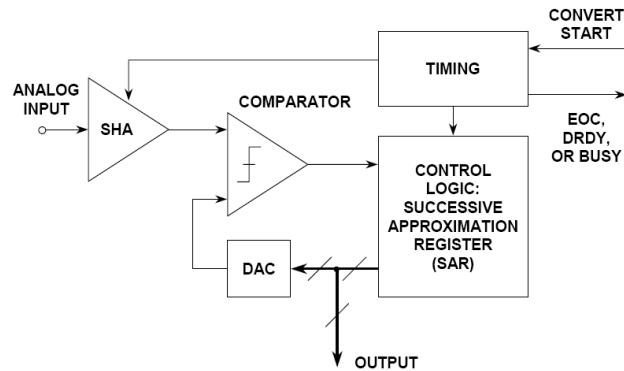
Based on operation of **Successive Approximation** ADCs, new ADC circuit is proposed here. In order to get better perception of this ADC, firstly we look at operation of **Successive Approximation** ADC.

#### 5.3 Successive Approximation ADC

The basic successive approximation ADC is shown in Figure 5.3. It performs conversions on command. On the activation of the CONVERT START command, the sample-and-hold (SHA) is located in the *hold* mode, and all the bits of the successive approximation register (SAR) are reset to "0" except the MSB which is set to "1". The SAR output drives the internal DAC. If the DAC output is larger than the analog input, this bit in the SAR is reset, otherwise it is left set. The next most



significant bit is then set to "1". If the DAC output is greater than the analog input, this bit in the SAR is reset, otherwise it is left set. The process is repeated with each bit in turn. When all the bits have been set, tested, and reset or not as appropriate, the contents of the SAR match to the value of the analog input, and the conversion is complete. These bit "tests" can form the basis of a serial output version SAR-based ADC.[24]

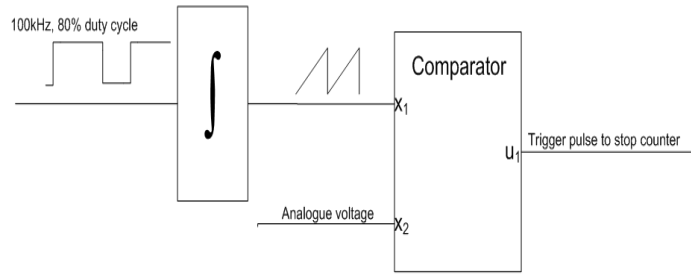


**Figure 5.3 Basic Successive Approximation ADC**

### 5.3.1 Structure of proposed ADC

From the previous section and Figure 5.3 idea of our ADC initiates, as it is shown in Figure 5.4, we move the control logic to FPGA and replace the DAC with sawtooth waveform, so main operation of converter is as follows:

- (i) Square wave generated by ADC modulator block integrates by analogue integrator to provide sawtooth waveform. Meanwhile counter inside of FPGA starts to count.
- (ii) Sawtooth waveform compares with the analogue voltage (sample), whenever analogue voltage becomes greater than sawtooth voltage output of comparator activates, when it activates counter in FPGA stops, so current value of counter is digital amount of analogue sample.



**Figure 5.4 Proposed ADC schematic**

Main advantage of proposed ADC is great reduction in wiring, for example if we use 10 bit ADC, we need at least 12 wires connected to FPGA in which 10 of them data signals and the rest are control signal; but using our ADC we need only two wires one for square wave and one for trigger (stop) signal. Another benefit of this scheme is faster speed compare to low cost ADCs.

In order to obtain conversion speed and resolution of ADC following equations are used.

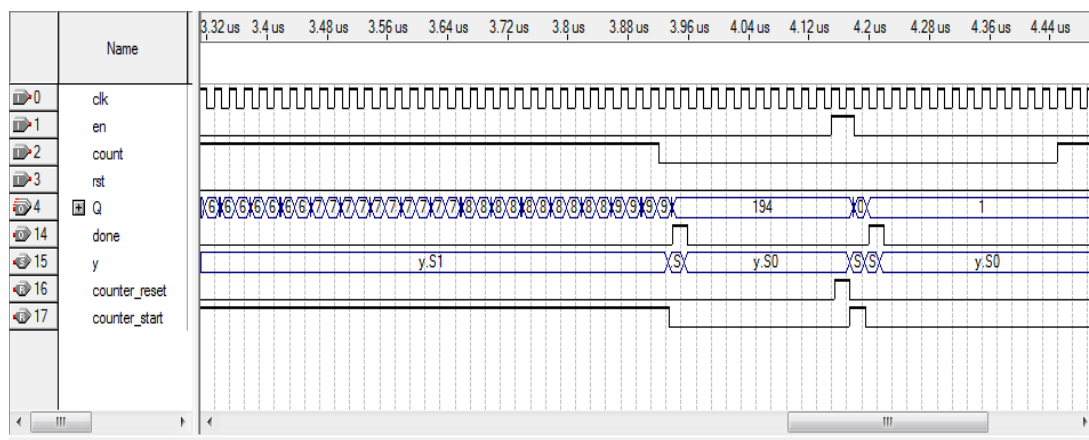
$$F_{\text{sampling}} = F_{\text{square wave}} \quad (5.1)$$

$$\text{Resolution} = \frac{DT_{\text{square wave}}}{T_{\text{counter}}} \quad (5.2)$$

From (5.1) if we choose  $F_{\text{square wave}}$  equal to control pulse (in fact it is compulsory in this case) and based our design in 4.2 control pulse frequency is 100kHz so  $F_{\text{sampling}} = 100 \text{ kHz}$ . Also according on what is shown in Figure 5.4 duty cycle of square wave is 80% thus whole duty cycle time is  $10\mu\text{s} \times 0.8 = 8\mu\text{s} = DT_{\text{square wave}}$ , in the other side period of counter is period of FPGA clock pulse which is 20ns, hence resolution of ADC from equation (5.2) obtains,

$$\text{Resolution} = \frac{8\mu\text{s}}{20\text{ns}} = 400 \quad (5.3)$$

The resolution obtained in (5.3) is more than 8 bits and less than 9 bits which is fair for our application. Simulation of ADC is using Quartus software is illustrated in Figure 5.5, as it indicates counter starts to count from beginning of each control pulse, it only stops when count input deactivates or by the other words stop command from external system activates, so in time of 390 us which count signal deactivates counter stops and done pin activates which means conversion is finished and data is ready. When done activates next blocks is ready to process data from Figure 5.1 this block is Gain.

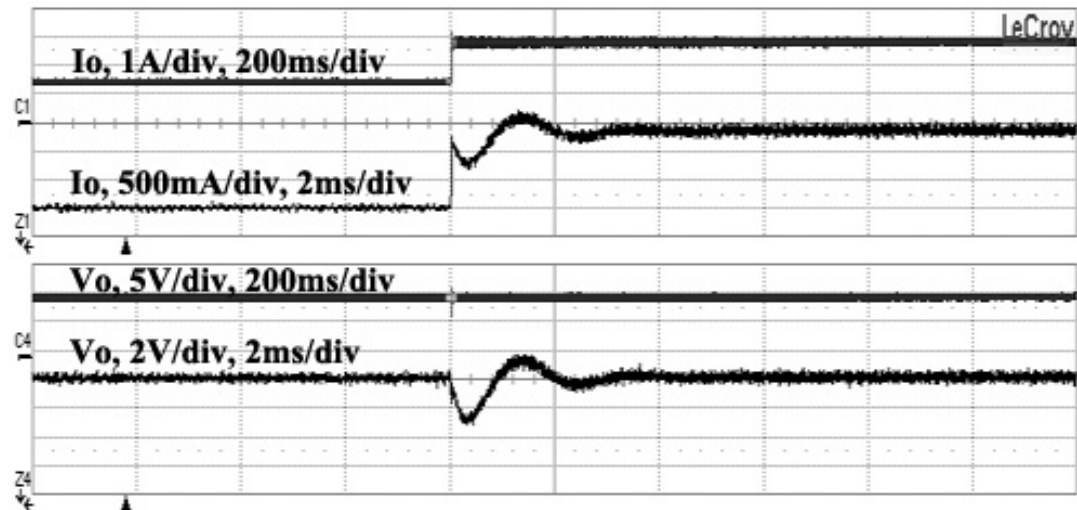


**Figure 5.5 Simulation of proposed ADC in Quartus**

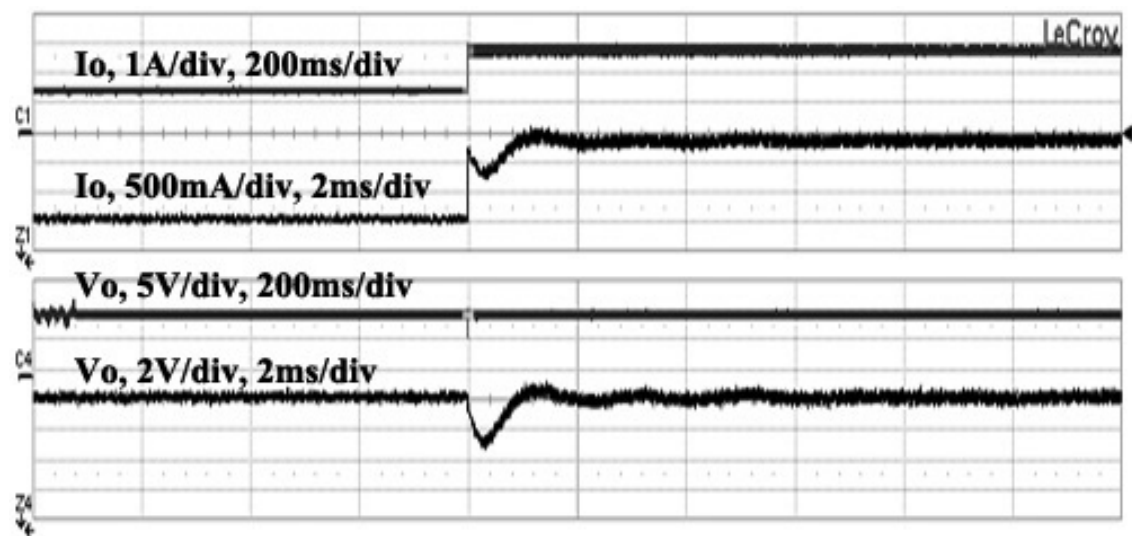
#### 5.4 Experimental Results

The controller hardware is implemented on FPGA, Altera Educational board (DE2) is utilized for this purpose. Since the design doesn't include common computational parts of CFLC, it occupies small areas of FPGA and numbers of used gates are a little portion of available gates. Total numbers of gates used for this design are 408 which include less than 1% of available logic gates in EP2C35F672C6N (32,216 gates). This issue makes it possible to use CPLD (complex programmable logic device) instead of FPGA, which has lower price and less pins and gates.

The boost converter specifications are as same as the simulation model. The applied disturbances to converter are similar to conditions in section. System responses to these disturbances are shown in following diagrams.



(a)



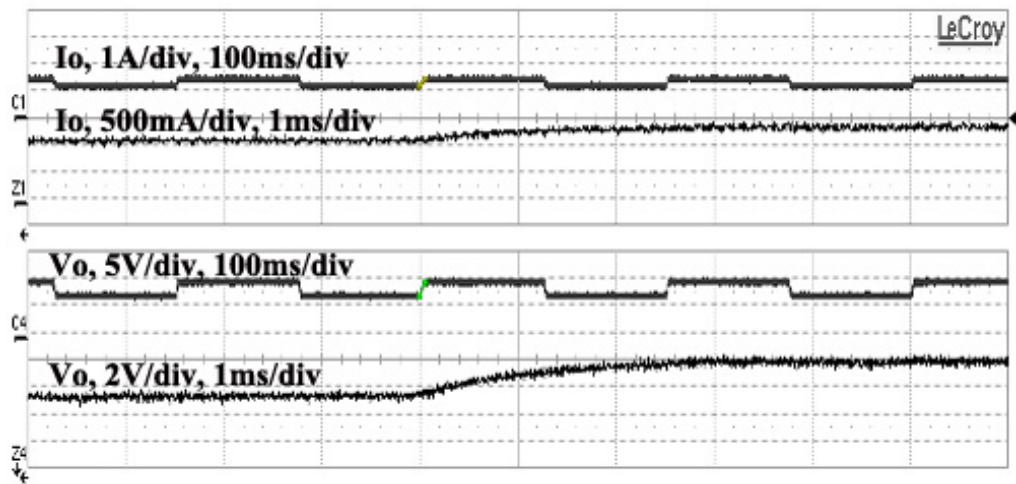
(b)

**Figure 5.6 Boost converter response using SIFLC in  $V_o=15\text{v}$  and load disturbance from  $10\Omega$  to  $5\Omega$  (200% of nominal load), (a) Using symmetrical MFs, (b) Using asymmetrical MFs (PWL2)**

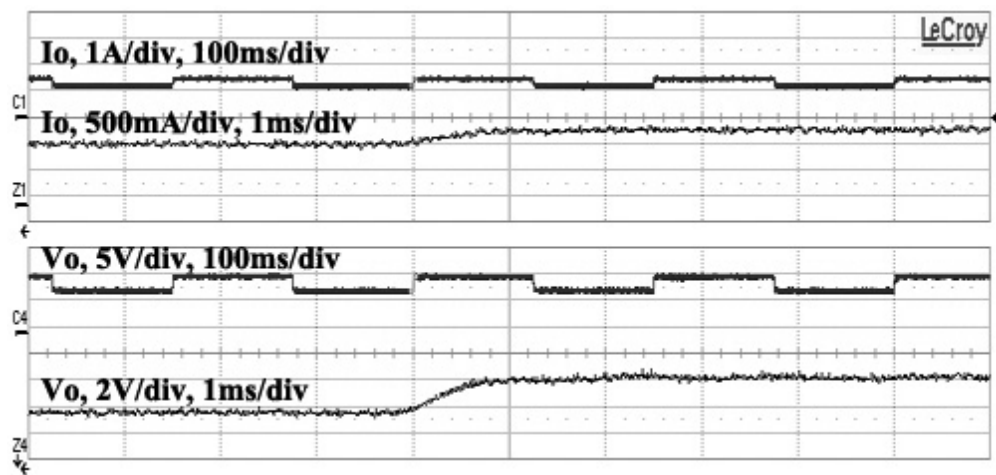
Experimental results for load disturbance show better performance of PWL2 controller. Output voltage is settled faster compare to symmetrical MFs. Furthermore PWL2 response doesn't have overshoot; consequently boost converter can be more

efficient using this controller type. Using this controller is valuable for systems with frequent load disturbances.

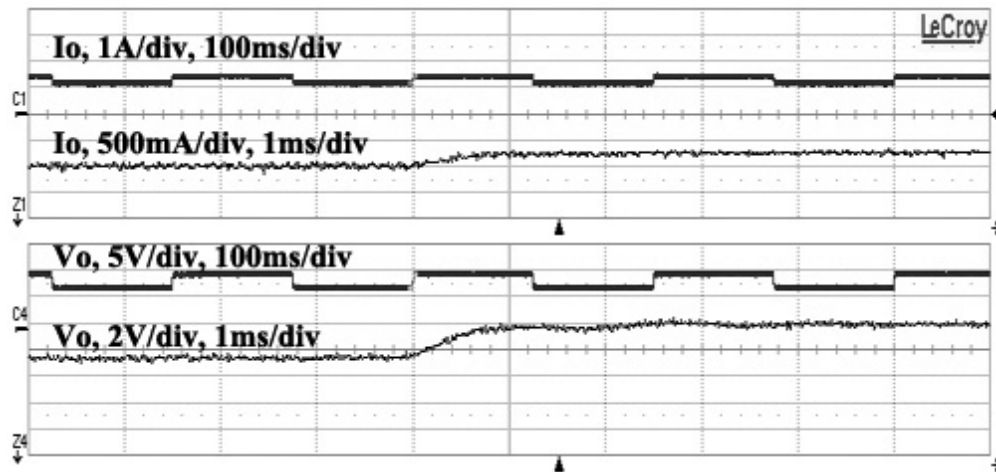
In Figures 5.7, periodic reference changes are applied to converter. Three MFs are used to investigate their response. The slowest response belongs to PWL2, PWL1 and symmetrical MFs have similar response but PWL1 seems to be a bit faster.



(a)



(b)



(c)

**Figure 5.7 Boost converter output voltage using SIFLC in  $10\Omega$  load and reference change from 12.5V to 15V, (a) Asymmetrical MFs (PWL2), (b) Using Asymmetrical MFs (PWL1), (c) Symmetrical MFs**

## CHAPTER 6

### CONCLUSIONS AND FUTURE WORK

#### 6.1 Conclusion

In this work SIFLC has been proposed to regulate the Boost converter output voltage. Simulation results of both the conventional FLC and SIFLC show identical responses for same disturbances. However, the measured CPU times for each controller are markedly different; SIFLC only requires about one tenth of CPU execution time. This suggest that SIFLC can be implemented using a much lower speed processor, or alternatively its control bandwidth can be increased significantly.

Simulation results also revealed that application of Toeplitz rule table for FLC leads to a PI controller with non-linear function controller; this issue can be a good clue to design new rule tables for power electronic systems which can not be replaced by non- intelligence controllers. In fact one importance of this research is showing this reality that if Toeplitz structure is used for FLC, paying extra cost in loosing speed or component price to implement FLC is useless, because operation of controller is as same as non-linear function controllers. Comparison of PI controller with SIFLC

demonstrated SIFLC with asymmetrical MFs has faster dynamic response although it has higher overshoot and undershoot for step change in load value.

In the hardware implementation part new ADC was proposed which has several advantages in case of wiring size and speed compare to conventional types.

## **6.2 Future work suggestions**

Although in this work we tried to contemplate some issues related to application of SIFLC to power converters, but it is possible to concentrate more on the CFLCs with different rule tables and MFs. In this case using DSP has more advantages versus FPGA because of simpler programming and its flexibility; by use of DSP several FLCs with different rule tables can be implemented and performances of them can be compared.

Another study which can be assumed to be improvement of this work is implementation of double loop controller with same controller characteristics for Boost converter; by adding current loop to close loop system it will be more robust to load changes.



## REFERENCES

1. Benjamin J. Patella, Aleksandar Prodic, Art Zirger, Dragan Maksimovic (2003), High-Frequency Digital PWM Controller IC for DC–DC Converters, *IEEE TRANSACTIONS ON POWER ELECTRONICS*, VOL. 18, NO. 1, 438-446.
2. S. Saggini, W. Stefanutti, E. Tedeschi, and Paolo Mattavelli (2007), Digital Deadbeat Control Tuning for dc-dc Converters Using Error Correlation, *IEEE TRANSACTIONS ON POWER ELECTRONICS*, VOL. 22, NO. 4, 1566- 1570.
3. Siew-Chong Tan, Y. M. Lai and Chi K. Tse (2006), Implementation of Pulse-width-Modulation Based Sliding Mode Controller for Boost Converters, *IEEE POWER ELECTRONICS LETTERS*, VOL. 3, NO. 4, 130-135.
4. Philip T. Krein (1998), *Elements of Power Electronics*, OXFORD UNIVERSITY PRESS.
5. V. S. C. Raviraj and P. C. Sen (1997), Comparative Study of Proportional–Integral, Sliding Mode, and Fuzzy Logic Controllers for Power Converters, *IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS*, VOL. 33, NO. 2, 518-524.
6. Chiu, S. (1998), Using Fuzzy Logic in Control Applications: Beyond Fuzzy PID Control, *IEEE Control System Magazine*, VOL. 18, issue 5, 100-105.
7. K. S. Tang, Kim Fung Man, Guanrong Chen, and Sam Kwong, An Optimal Fuzzy PID Controller, *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*, VOL. 48, NO. 4, (AUGUST 2001) 757-765.
8. Wing-Chi So, Chi K. Tse, Yim-Shu Lee (1996), Development of a Fuzzy Logic Controller for DC/DC Converters: Design, Computer Simulation, and

Experimental Evaluation, IEEE TRANSACTIONS ON POWER ELECTRONICS, VOL. 11, NO. 1, 24-32.

9. Abdul R. Ofoli, Ahmed Rubaai (2006), Real-Time Implementation of a Fuzzy Logic Controller for Switch-Mode Power-Stage DC–DC Converters, IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS, VOL. 42, NO. 6, 1367-1374.
10. R. R. Boudreaux, R. M. Nelms, John Y. Hung (1997), Implementation of a Fuzzy Controller for DC–DC Converters Using an Inexpensive 8-b Microcontroller, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 44, NO. 5, 661-669.
11. Byung-Jae Choi, Seong-Woo Kwak, and Byung Kook Kim (2000), Design and Stability Analysis of Single-Input Fuzzy Logic Controller”, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 30, NO. 2, 303-309.
12. Kanakasabai Viswanathan, Ramesh Oruganti, Dipti Srinivasan (2005), Nonlinear Function Controller: A Simple Alternative to Fuzzy Logic Controller for a Power Electronic Converter, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 52, NO. 5, 1439-1448.
13. S. M. Ayob, Z. Salam and N. A. Azli(2007), Piecewise Linear Control Surface for Single Input Nonlinear PI-Fuzzy Controller, The 7th International Conference on Power Electronics and Drive Systems (PEDS 2007), , Bangkok, 27 – 30 November , 1533 – 1536.
14. Ned Mohan (2003), First course on Power Electronics and Drives, MNPERE Minneapolis, Chapter 4.

15. Brad Bryant and Marian K. Kazimierczuk (2005), Modeling the Closed-Current Loop of PWM Boost DC–DC converters operating in CCM With Peak Current-Mode Control, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, VOL. 52, NO. 11, 2402-2412.
16. M. H. Rashid (2001), Power electronics handbook: devices, circuits, and applications, Academic Press.
17. SIMULINK User's Guide, © Math Works Inc. , 2007.
18. Liping Guo(2007), Implementation of Digital PID Controllers for DC-DC Converters using Digital Signal Processors, IEEE EIT 2007 Proceedings, 306-311.
19. Benjamin J. Patella, Aleksandar Prodic', Art Zirger, and Dragan Maksimovic' (2003), High-Frequency Digital PWM Controller IC for DC–DC Converters, IEEE TRANSACTIONS ON POWER ELECTRONICS, VOL. 18, NO. 1, 438-446.
20. Jan Jantzen (1998), *Design of fuzzy controllers*. Tech. report no. 98-E 384, TU denmark, Dept. of Automation, Lyngby, Denmark.
21. L.A. Zadeh (1975), The concept of a linguistic variable and its application to approximate reasoning. Information Sci. 8, 199–249 and 9, 43–80.
22. Daijin Kim (2000), An Implementation of Fuzzy Logic Controller on the Reconfigurable FPGA System, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 47, NO. 3, 703-715.
23. Altera DE2 Development and Education board © Altera Corporation.
24. Walt Kester , ADC Architectures II: Successive Approximation ADCs, MT-021 TUTORIAL, © Analog Devices.