PetraFuz: a Low Cost Embedded Controller Based Fuzzy Logic Development System

Thiang, Anies Hannawati, Resmana Lim and Hany Ferdinando
Electrical Engineering Department, Petra Christian University
Siwalankerto 121-131, Surabaya 60236 - INDONESIA
Phone: 62-31-8439040; Fax: 62-31-8436418
E-mail: {thiang, anies, resmana, hanyf}@petra.ac.id

Abstract

This paper describes a low cost embedded controller based fuzzy logic development system (PetraFuz, Petra Fuzzy Development System), developed by Control System Laboratory, Electrical Engineering, Petra Christian University, Surabaya – Indonesia. This system consists of embedded controller MCS51 target hardware and a Personal Computer (PC)-based supported fuzzy development software. The target hardware is linked to the PC via serial communication interface. Fuzzy kernel based on Mamdani type has been developed and embedded on the controller. On the other hand, Windows based software has been developed providing a user-friendly interface.

PetraFuz is used to develop fuzzy logic application systems especially for control systems, which intend to be implemented on a low cost embedded controller. The abilities of this system are for supporting several stages in developing a fuzzy logic application such as: designing fuzzy control (membership function, fuzzy if-then rules, etc.), evaluating control surface, making assembly code for MCS51 and downloading compiled binary code to the target hardware of microcontrollers MCS51. Control system response can be acquitted by Petrafuz software from RS232 serial communication, therefore system response can be drawn at PC monitor which are very important to analyze for fuzzy tuning process (updating if-then rules & membership function/ MF). Controller tuning process can be done in an online manner.

The PetraFuz tools help people interesting in designing and implementing a fuzzy logic control system on an embedded controller. Hopefully, PetraFuz could give a positive contribution for academic & hobbies society interesting in developing various applications of fuzzy logic control.

1. Introduction

Fuzzy Logic has been applied in wide area, starting from consumer electronics to robotics and industrial control systems [1]. General-purpose microcontrollers or Fuzzy Logic Processors usually does the implementation of Fuzzy Logic Controller. They need special software to develop fuzzy logic application from design, evaluation, implementation, and tuning process.

This paper present PetraFuz, Petra Fuzzy Development Systems built by Control System Laboratory, Department of Electrical Engineering, Petra Christian University, Surabaya, Indonesia. It consists of minimum system MCS51 microcontroller and support software for PC. It will provide tools in developing system using fuzzy logic, especially in control system.

Motorola (FUDGE, Fuzzy Design Generator for 68HC11, 6805, and 68000 [2]) has inspired Control System Laboratory to develop PetraFuz. PetraFuz system has more feature like integrated fuzzy development system together with its hardware. The interaction between PC (via software) and target system is done through RS232 serial port. PetraFuz chose microcontroller from MCS51 family, since it is more popular and available in Indonesia. Hopefully, PetraFuz should be able to give positive contribution to Indonesia Control Society to develop many applications using fuzzy logic.

2. PetraFuz System

PetraFuz system consists of two parts, i.e. software (PC) called PetraFuz51 and hardware target system using MCS51 microcontroller with Fuzzy Inference System Kernel (FIS Kernel). Figure 1 shown Block Diagram of PetraFuz system.

PetraFuz51 does the following process: design, evaluation, assembly code generation, and downloading MCS51 machine code to target system, and the target does fuzzy logic process which interact to I/O interface. Fuzzy logic process consists of fuzzyfication, rule evaluation, and defuzzyfication. The processes in designing fuzzy logic controller are membership function design for input and output, and design of fuzzy if-then rule. The evaluation stage provides control surface for several input combination. Therefore, user can look at real control response done by hardware target system.

PetraFuz51 generate MCS51 assembly code, and it is compiled and downloaded to target system via RS232 communication system. Control action can be monitored from PC so that control response can be analyzed for tuning process. Beside the control response, each fuzzy logic input and output can also be monitored from PC.

Feature:

- Maximum 5 Input and 3 Output
- 8 Membership Functions per Input
- 8 Membership Functions per Output
- 1024 if-then Rule
- 15 Characters per Name (Input, Output, Member).
- 4 Points per Input Member. (Trapezoid MF)
- 1 Point per Output Member. (Singleton MF)

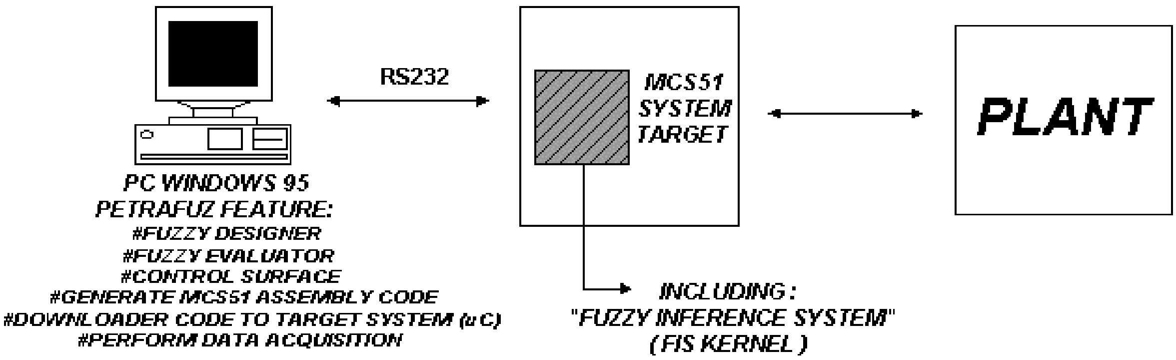

Figure 1. Block Diagram of PetraFuz System

3. PetraFuz51 Software

PetraFuz51 software was built using Delphi v.2 with graphical user interface so that makes it easier for user. The software provides facilities like input-output MF design, fuzzy if-then rule design, fuzzy logic evaluator, control surface, assembly code generation, downloader to target and data acquisition.

The first procedure of using PetraFuz is fuzzy inference system design. It includes fuzzy I/O MF and fuzzy if-then rule design. The development of database (it consists of MF and if-then rules) from the first procedure using MCS51 assembly language then combining with user software are the second procedure. The user software is software made by user to access to/from IO interface (analog or digital). The third procedure is to compile user software then download it to target system. The number of crisp input and output are limited to 5 and 3 respectively, since target system has limited memory also.

MF design for input and output can be done easily in graphic mode or by entering point value of the MF. MF shape is limited to trapezoid and triangle built from four points. Output MF uses singleton type built from one point. Figure 2 and 3 show example of input and output MF design. Input MF has 5 labels, 2 trapezoids and 3 triangles type.

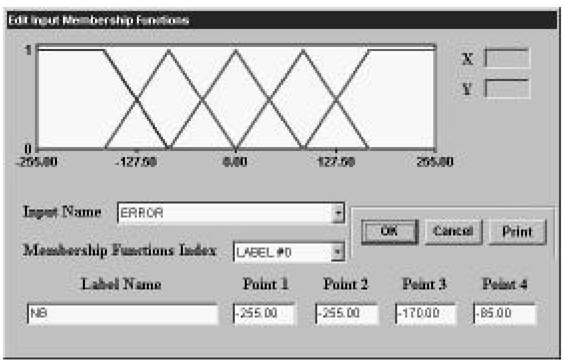

Figure 2. MF Input Design

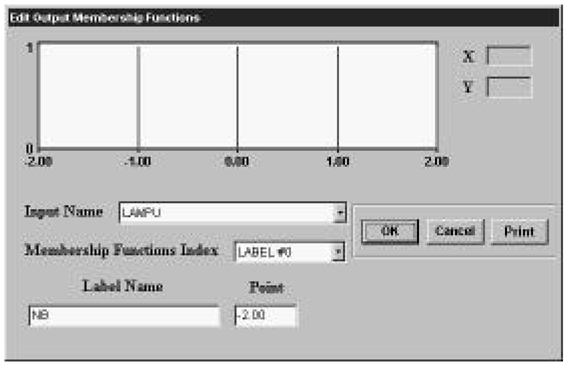

Figure 3. Output Membership Function Design

The maximum number of rule in PetraFuz51 is limited to 1024 rules, with unlimited antecedent and consequent for each rule. Between antecedents or consequents in one rule use ' and' operator. Figure 4 shows rule designed by PetraFuz51.

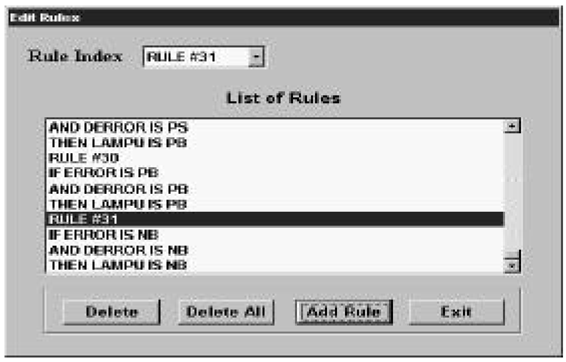

Figure 4. Fuzzy If-Then Rule Design

Fuzzy logic evaluator facility provide evaluation tool to analyze control system. It can evaluate fuzzyfication result, active rules, and defuzzyfication result for predetermined value. Therefore, it will give description about fuzzy logic controller action briefly. Figure 5 shows fuzzy logic evaluator.

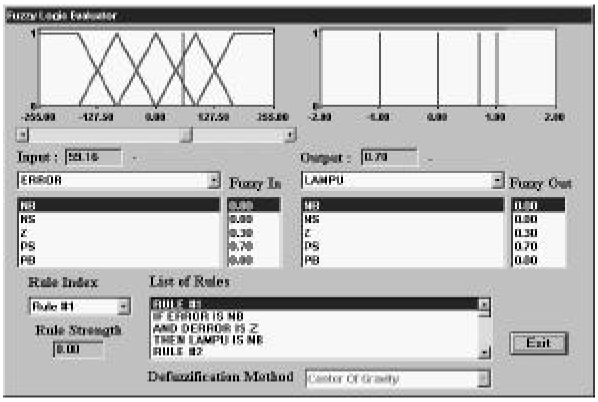

Figure 5. Fuzzy Logic Evaluator

Control surface facility helps user to evaluate control system action before being applied to target system. It is shown in 2D graphics between fuzzy output and one input variation. If number of input more than one, then the other inputs are treated as constant. Figure 6 shows sample of graphics from fuzzy logic controller system.

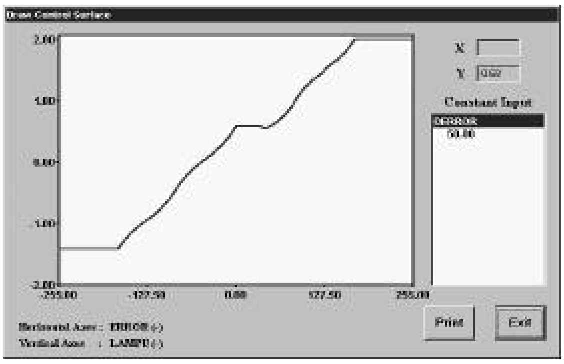

Figure 6. Sample of Graphics From Control Surface

Designed MF and rules have to be converted to MCS51 assembly language form, so it can be combined with user software and downloaded to target system. This can be done using generate code facility. The facility will prepare database from MF and rules then save them in .ASM file. All range value for input and output will be converted in 0-FFh ranges to avoid negative number. Figure 7 shows database structure in MCS51 assembly language generated by PetraFuz51.

File from generate code facility then combined with user software to compile. After compiling user software to machine code, PetraFuz51 will download it to system target to do fuzzy logic controller process. User software must be compiled in Intel Hex format since PetraFuz51 software downloader work in Intel Hex format.

Data acquisition facility provide for data acquisition from target system. It shows control system response and input-output value versus time. So, control system response can be evaluated by online. This facility is important, especially in tuning MF and if-then rule process. Based on this facility, user can change the design parameter and download it to target system to see its response. This process can be done in repeat way and easily until it reaches optimum point. Figure 8 and 9 show sample of data acquisition.

```
                .CODE
                ORG 2100H
INPUT_MFS    ;Input Membership Functions
INMF         ;ERROR
             DB 000H,0FFH,02AH,006H   ;NB
             DB 02AH,006H,055H,006H   ;NS
             DB 055H,006H,080H,006H   ;Z
             DB 080H,006H,0AAH,006H   ;PS
             DB 0AAH,006H,0FFH,0FFH   ;PB
                .CODE
                ORG 21A0H
OUTPUT_MFS   ;Output Membership Functions
OUTMF        ;LAMPU
             DB 000H                     ;NB
             DB 040H                     ;NS
             DB 080H                     ;Z
             DB 0BFH                     ;PS
             DB 0FFH                     ;PB
                .CODE
                ORG 21C0H
RULE_START   DB 000H                     ;IF ERROR IS NB
             DB 00AH                     ;AND DERROR IS Z
             DB 080H                     ;THEN LAMPU IS NB

             DB 000H                     ;IF ERROR IS NB
             DB 00BH                     ;AND DERROR IS PS
             DB 081H                     ;THEN LAMPU IS NS
```

Figure 7. Database Structure in MCS51 Assembly Language

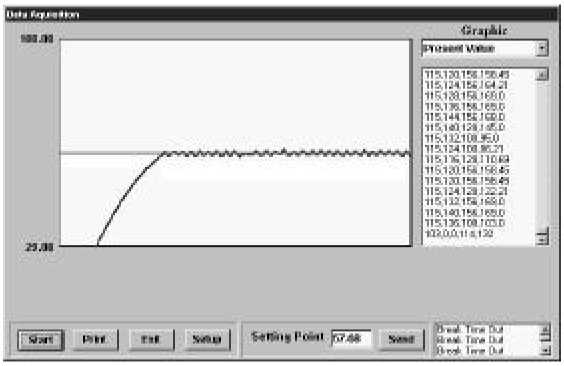

Figure 8. Data Acquisition From Target System (1)

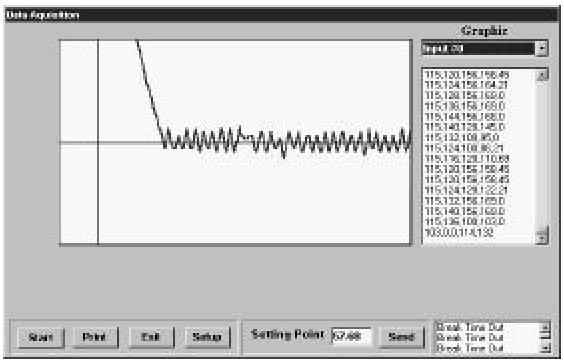

Figure 9. Data Acquisition From Target System (2)

4. Target System

Target system uses MCS51 microcontroller with downloading system. It consists of 89C51 processor with internal 4 KB PEROM, 8 KB EEPROM, PPI 8255, and RS232 serial interface. Figure 10 shows memory map of target system. The development of target system was one by Innovative Electronics, Surabaya, Indonesia.

Downloading system software and FIS kernel were placed in internal PEROM. To run FIS kernel, user software needs to jump to FIS kernel routine at address 40Ah. Downloaded user software was placed in EEPROM starting from address 2000h. Therefore user software must start from this address. Target system has parallel interface using 8255 to add number of I/O port. RS232 serial interface is used to communicate between target system and PC for downloading and data acquisition.

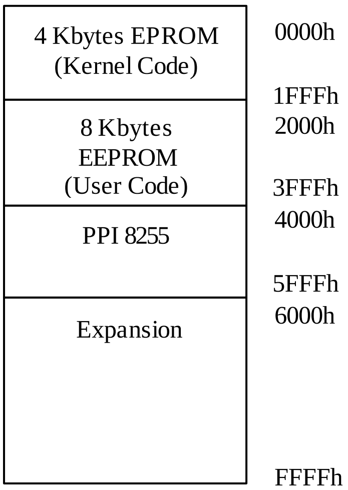

Figure 10. Memory Map of Target System

5. Data Structure and FIS Kernel

FIS kernel is a routine for fuzzy inference process, i.e. fuzzyfication, rule evaluation, and defuzzyfication. It uses center of gravity for defuzzyfication method. It is designed for database format generated by PetraFuz51 software specially.

Each label of input MF database uses four bytes wide. The first and third byte is point 1 (P1) and 2 (P2) respectively, while the second and fourth byte are slope 1 (S1) and 2 (S2) respectively (look at figure 11). Output MF database uses one byte for each label.

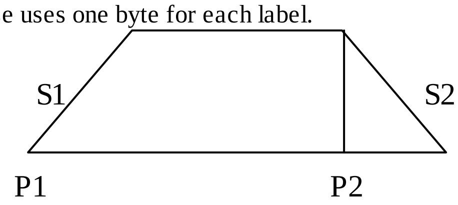

Figure 11. Parameter of Input MF Database

Memory used by rule database depends on the number of antecedent and consequent for each rule. One antecedent or one consequent uses one byte of memory. If the rule uses two antecedents and one consequent, then it uses three bytes of memory. Figure 12 shows rule database structure.

Value of crisp input and output are placed in internal RAM of 89C51 processor. Crisp input address is 08h to 0Ch, while the crisp output is 0Dh to 0Fh. Address 08h is for the first crisp input, address 09h is for the second one, and so on, so is the crisp output starts from address 0Dh. Detail of this address can be looked at table 1. Before running FIS kernel, user software has to save crisp input values to appropriate address and FIS will take value from this address to do fuzzy inference process. User software takes result of fuzzy inference process at crisp output location.

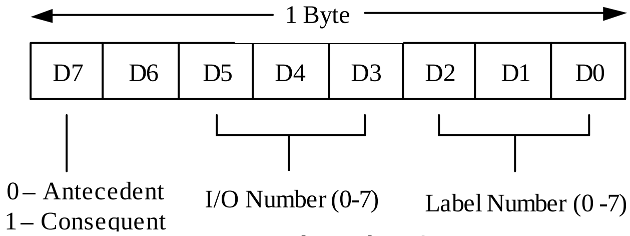

Figure 12. Rule Database Structure

Table 1. Crisp Input and Output Location

Address	Function
08h	Crisp Input 1
09h	Crisp Input 2
0Ah	Crisp Input 3
0Bh	Crisp Input 4
0Ch	Crisp Input 5
0Dh	Crisp Output 1
0Eh	Crisp Output 2
0Fh	Crisp Output 3

6. PetraFuz Application

Control system designers have difficulty in applying fuzzy logic method, especially in hardware implementation. One of the reason is there is no fuzzy development tools equipped with hardware target system. PetraFuz system provides complete tools from MF design stage and fuzzy if-then rule development until fuzzy logic control process in hardware target system. However, the most difficult processes are designing MF and fuzzy rules, so that it can result the optimum fuzzy logic controller. User needs to experiment in repeat way to get the best result (tuning process). If it does not support with good tools, it will need more time.

Experimental result showed that FIS kernel could evaluate 16000 rules in one second approximately. The experiment was done in the following condition: each rule has two antecedents and one consequent and 89C51 uses 11.0592 Mhz crystals. It can be concluded that FIS kernel needs less than 62.5 µs. If fuzzy logic controller has two inputs and one output (five labels MF for each input), then maximum number of rule will be 25 rules, so that it needs less than 1.5625 ms to evaluate the rules.

With faster access time, PetraFuz can be applied to relative fast control system like DC motor speed control, position control, etc. It can also be applied to industrial control like pressure control, chemical control, etc.

Control System Laboratory Department of Electrical Engineering Petra Christian University uses PetraFuz system as one of the exercise module, and it helps students to understand and design fuzzy logic controller. FIS kernel was modified for 8088 microprocessors and applied it to predict number of vehicles based on their noise.

PetraFuz51 software can be downloaded freely at http://ee.petra.ac.id/basiclab/ petrafuz.htm

7. Conclusion

The experiment in developing and applying PetraFuz, it can be concluded the following:

- PetraFuz system is easy to use and useful to beginners who want to do an experiment in fuzzy logic control system. Nowadays, PetraFuz uses MCS51 family and Intel 8088 microprocessors, but it can be expand to another processors, like MCS96 family, Z80 family, Mitsubishi microcontrollers, etc.
- PetraFuz provides interactive and integrated ability with hardware target system. It will give more chance to be applied in many fields in Indonesia, especially in agriculture.
- The next future of PetraFuz will be accommodate more I/O number, MF shape variation, and many defuzzyfication method. It can also be combined with another intelligent system such as Artificial Neural Network to result an intelligent control.

Acknowledgement

On this occasion, writers would like to thank to Control System Laboratory staff Department of Electrical Engineering Petra Christian University, Ban Eng, Jaury Adi Wijaya, Silviya, Kardy Antolis, Darwin, Eric PS, and Heri Soehartono for their support in developing this project, our colleagues at our department for their support until we finished this project.

Reference

[1] Klir, George J, *Fuzzy Sets and Fuzzy Logics: Theory and Applications*, NJ: Prentice Hall, 1995.

[2] Alex De Castro, Jason Spelman, John Dumas, *Fuzzy Designer Generator (FUDGE)*, Motorola Inc., 1994

[3] Center for Emerging Computer Technologies, *Fuzzy Logic Education Program*, Motorola Inc., 1994

[4] Advanced Micro Device, *Microcontrollers Handbook*, California: Advanced Micro Device, 1988.

[5] Sanchez, Julio, *IBM Microcomputers: A Programmer's Handbook*, New York: McGraw-Hill, 1990