FUZZY LOGIC IN EMBEDDED MICROCOMPUTERS AND CONTROL SYSTEMS. SOFTWARE CONSIDERATIONS

ROTAR DAN

University of Bacau, Energetic Mecatronic and Computers Science Department

Abstract: Fuzzy logic control can be used to implement a wide variety of intelligent functions including everything from consumer electronic goods and household appliances to auto electronics, process control, and automation. The paper presents the mains tools used for PIC microcontroller programming. The software techniques for fuzzy software programming are presented and the main aspects of the fuzzy controller are discussed.

Keywords: development systems**,** software**,** microcontroller, interface, fuzzy logic, linguistic variables, peripheral, memory, crisp numbers

1. INTRODUCTION

When it comes to making complex judgement calls, computers can't replace people. But with artificial intelligence, computers could be trained to think like humans do. Artificial intelligence allows computers to learn from experience, recognize patterns in large amounts of complex data and make complex decisions based on human knowledge and reasoning skills. Artificial intelligence has become an important field of study with a wide spread of applications in fields ranging from medicine to agriculture.

This paper shows the software facilities for the PIC fuzzy controller programming and the main techniques used. The main solutions to embedded system designs are presented and the microcontroller programming are discussed.

2. DEVELOPMENT TOOLS

<u>MPLAB™ IDE</u>

MPLAB is a Windows-based development platform for the Microchip Technology PIC16/17 one-time-programmable (OTP) microcontroller families. MPLAB Integrated Development Environment (IDE) offers a project manager and program text editor, a user-configurable toolbar containing four pre-defined sets and a status bar, which communicates editing and debugging information.
MPLAB is provided with MPLAB-ICE 2000 In-Circuit Emulator running under the Windows environment. This real-time emulator supports low-voltage emulation, to 2.0 volts, and full-speed emulation. MPLAB-ICE 2000 is designed to provide product development engineers with an optimized design tool for developing target applications. This universal in-circuit emulator provides a complete MCU design toolset for PIC micro MCU's in the PIC12CXXX, PIC16C5X, PIC16CXXX, PIC17CXXX and PIC18CXXX families. MPLAB-ICE 2000 is CE compliant.

<u>MPLAB-C17</u>

The MPLAB-C17 compiler is a full-featured C compiler for the Microchip Technology PIC17CXXX family of 8-bit One-Time-Programmable (OTP) microcontrollers.

Total Endurance™
Microchip's revolutionary Total Endurance Model provides electronic systems designers with unprecedented visibility into Serial EEPROM-based applications.

The FilterLab™

Active Filter Design Tool simplifies active filter design for embedded systems designers. The unique FilterLab software automates the design of the anti-aliasing filter for an analog-to-digital converter-based data acquisition system. FilterLab also provides full schematic diagrams of the filter circuit with component values, a SPICE model, and displays the frequency and phase response.

fuzzyTECH-MP
The fuzzyTECH-MP Development System comes in two versions. The Explorer, contains everything you need to gain a comprehensive working knowledge of fuzzy-logic system design. It is easy-to-use; all graphic editors and tools guide you step-by-step through the development phases of fuzzy systems. The Explorer supports two input variables and one output variable.

3. FUZZY LOGIC IN EMBEDDED MICROCOMPUTERS

Fuzzy logic control can be used to implement a wide variety of intelligent functions including everything from consumer electronic goods and household appliances to auto electronics, process control, and automation.

Dr. Lotfi Zadeh, the originator of fuzzy logic, noted that ordinary language contains many descriptive terms whose relevance is context-specific.Typically, fuzzy logic control applications fall into two categories. First, it can be used to enhance existing products with intelligent functions. Second, it can utilize sensors that continuously respond to changing input conditions. In addition, fuzzy logic simplifies dealing with non-linearities in systems, and allows for quicker product development cycles.

Computers–especially those in embedded applications–can be programmed to perform calculations in the fuzzy domain rather than the crisp domain. Fuzzy logic manipulations take advantage of the fact that linguistic variables are only resolved to crisp values at the resolution of the problem, a kind of self scaling feature that is objective-driven rather than data-driven.

Linguistic variables are central to fuzzy logic manipulations. Linguistic variables hold values that are uniformly distributed between 0 and 1, depending on the relevance of a context dependent linguistic term.

Linguistic variables in a computer require a formal way of describing a linguistic variable in the crisp terms the computer can deal with.

Fuzzy logic first translates the crisp inputs from the sensor into a linguistic description. Then it evaluates the control strategy contained in fuzzy logic rules and translates the result back into a crisp value.

Applications may be computed in either the fuzzy linguistic domain or the conventional crisp domain. Non-linear problems, such as process control in an environment that varies considerably from usage to usage, yield very workable results with impressively little development time when solved using fuzzy logic. Although fuzzy logic is not essential to solving this type of non-linear control problem, it helps in describing some of the possible solutions.

Fuzzy logic operators provide a formal method of manipulating linguistic variables. It is a reasonable comment to describe fuzzy logic as just another programming paradigm.

For example, to add the linguistic variable HOT to a computer program running in an embedded controller, we need to translate the graphical representation (Figure 1) into meaningful code. The following C code fragment gives one example of how we might do this. The function Temperature_HOT returns a degree of membership,

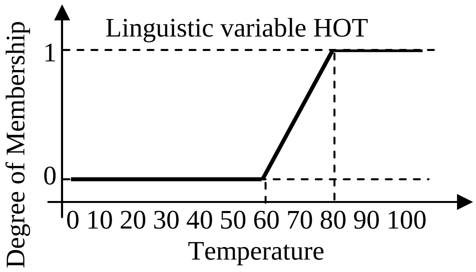

Figure 1. Graphic representation
for a linguistic variable.

scaled between 0 and 255, indicating the degree to which a given temperature could be HOT. This type of simple calculation is the first tool required for calculations of fuzzy logic operations.

```
unsigned int Temperature; /* Crisp value of Temperature */
unsigned char Temperature_HOT (unsigned int __CRISP)
    {
        if (__CRISP < 60) return(0);
        else
        {
            if (__CRISP <= 80) return(((__CRISP - 60) * 12) + 7);
            else
            {
                return(255);
            }
        }
    }
```

The same code can be translated to run on many different embedded micros, as displayed in the next example.

```
fuzzyfication   movfw crisp
                sublw D'60'       ; if (__CRISP < 60) return(0)
                btfsc STATUS,Z
                goto equal1
                btfsc STATUS,C  ;else return(255)
                retlw D'0'
equal1          movfw crisp       ; if (__CRISP <= 80) return(((__CRISP - 60) * 12) + 7)
                sublw D'80'
                btfsc STATUS,Z
                goto equal2
                btfsc STATUS,C
                goto equal2
                retlw D'255'
equal2          movlw D'60'
                subwf crisp,w
                movwf m1
                movlw D'0'
                movwf m2
                movlw D'12'
                movwf d1
multiply        movfw m2
                addwf m1,w
                decfsz d1
                goto multiply
                return
```

We consider, for example an air conditioning system. When fuzzy logic is used in an application program, we might implement an air conditioner controller with a single fuzzy statement.

IF temperature IS hot THEN air_conditioner is on;

Central to the manipulation of fuzzy variables are fuzzy logic operators that parallel their boolean logic counterparts: f_and, f_or and f_not. We can define these operators as three macros to most embedded system C compilers as follows.

#define f_one 0xff;

#define f_zero 0x00;

#define f_or(a,b) ((a) > (b) ? (a) : (b));

#define f_and(a,b) ((a) < (b) ? (a) : (b));

#define f_not(a) (f_one+f_zero-a);

The description of the linguistic variable MUGGY is, however, more complex. Typically, we think of the condition MUGGY as a combination of HOT and HUMID. We can describe a controlling parameter ACcontrol for an air conditioner with the following equation.

IF Temperature IS HOT AND Humidity IS HUMID THEN ACcontrol is MUGGY;

Different variables can have the same linguistic member names. Like members of a structure or enumerated type in most programming languages, they do not have to be unique. It is important to note that many of the linguistic conclusions are a result of the general form of the above equation. Using the fuzzy logic methods the resulting execution speed increase can be impressive, even on simple 8 bit microcomputers.

4. CONCLUSIONS

Fuzzy logic is an innovative technology that enables users to implement intelligent functions in microcontroller devices. With an embedded fuzzy logic control system, application engineers can use low-cost 8-bit microcontrollers to implement intelligent, complicated functions and adaptive control loops with limited resources.

REFERENCES

[1] H. Watanabe, *RISC Approach to Design of Fuzzy Processor Architecture*, Proc. IEEE Int'l Conf. Fuzzy Systems, IEEE, pp. 431-441, 1992.

[2] C. Von Altrock, B. Krause, and H.J. Zimmermann, *Advanced Fuzzy Logic Control Technologies in Automotive Applications*, Proc. IEEE Int'l Conf. Fuzzy Systems, IEEE, Piscataway, N.J., pp. 835-842, 1992.

[3] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Englewood Cliffs, NJ, 1995.

[4] Lisa Meeden. *Using Robots As Introduction to Computer Science.* In John H. Stewman, editor, Proceedings of the Ninth Florida Artificial Intelligence Research Symposium (FLAIRS), pages 473--477. Florida AI Research Society, 1996.

[5]*** Microchip, PICmicro™ Mid-Range Reference Manual (DS33023).