Microcontroller Implementation of Fuzzy Guidance System

Ahmed ElSayed[1], Mohamed Rizk[2], Tarek Sobh[1]
[1] Department of Computer Science and Engineering, University of Bridgeport
[2]Electrical Engineering Department, Alexandria University

Abstract

In the last few years microcontrollers found a big place in industrial applications. It had been used in many applications such as washing machines, textile machines and many other fields. Microchip PIC microcontrollers are one of the most used types of microcontroller. In this paper we consider a method which can be used to implement a fuzzy control system and the description of this implementation technique. The proposed method will be applied to implement the fuzzy guidance systems discussed in [1],[2] on a microchip PIC16F877A microcontroller system.

Fuzzy Systems Implementation Techniques

There are number of methods to implement the fuzzy logic control system. Generally, we can divide them into two major categories as shown in table 1.

Table 1, Major implementation methodologies of fuzzy controller

Standard hardware	Fuzzy hardware
1) Computer: PC or WS 2) Processors: microprocessor, microcontroller, DSP 3) Look-up table: digital memory devices 4) A/D & D/A converters	1) Fuzzy logic controller: CMOS semi/full custom design 2) Fuzzy processor: microprocessor plus fuzzy add on 3) Fuzzy microcontroller: microcontroller plus FLC 4) Analog fuzzy controller: analog circuit design

Integrated Fuzzy Guidance System

The proportional navigation method is a guidance method where the controller will get the LOS angle from missile to target (λ) and the rate of change of this angle ($\dot{/}$) as two inputs, and will produce the elevator angle delta δ (fin deflection angle) as an output, Figure 1.

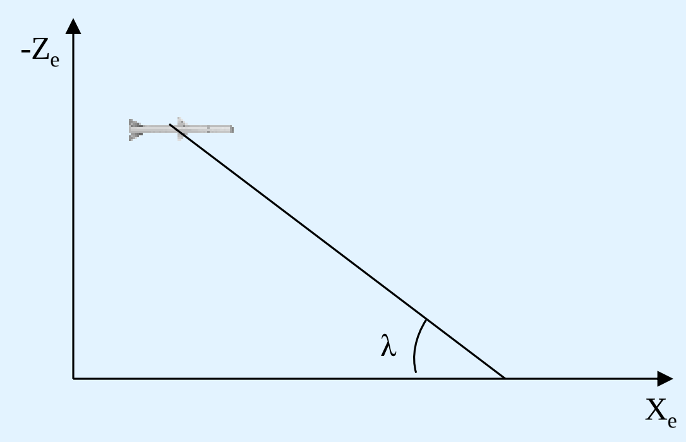

Figure 1, The proportional Navigation Guidance System

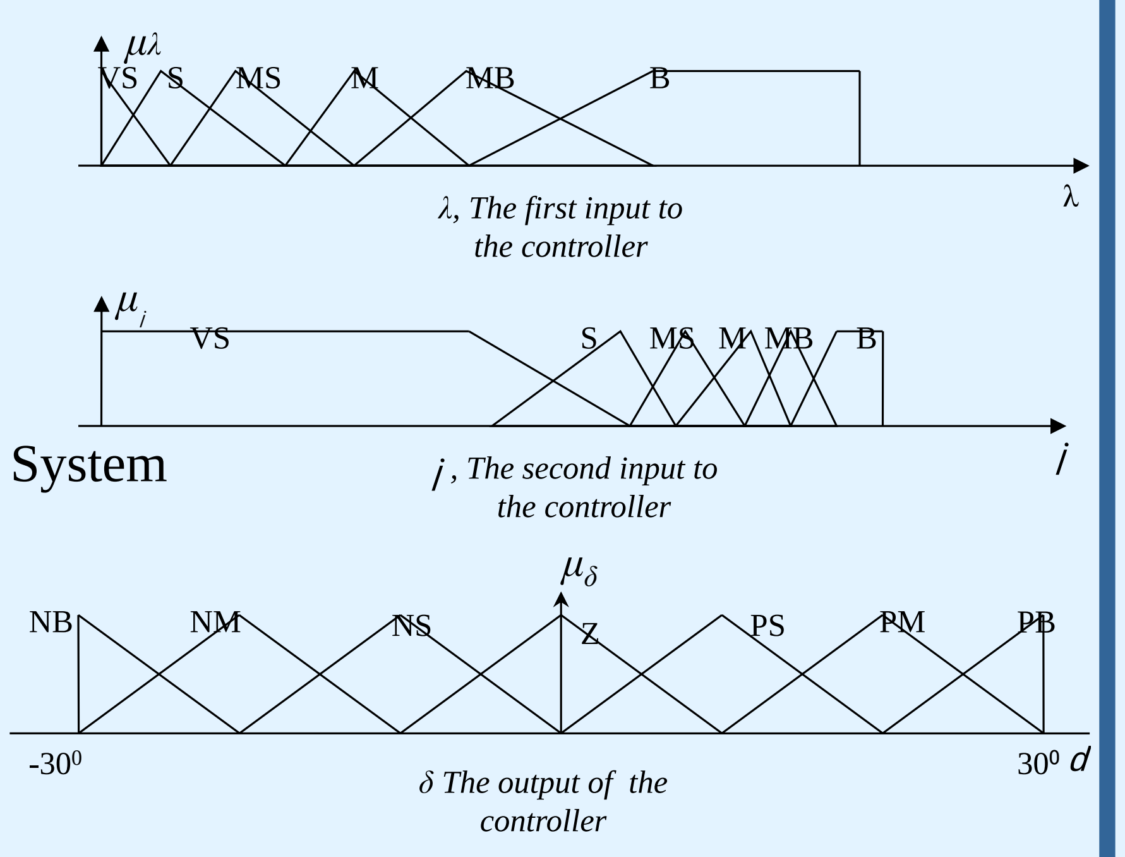

Figure 2, The membership function of the Integrated proportional navigation fuzzy guidance system illustrated in [1][2].

Proposed System Architecture

Figure 3 shows block diagram of the proposed system with its PC interface to configure the serial memory I2C. Figure 4 shows the structure of fuzzy system with fuzzification and defuzzification which we used in our implementation.

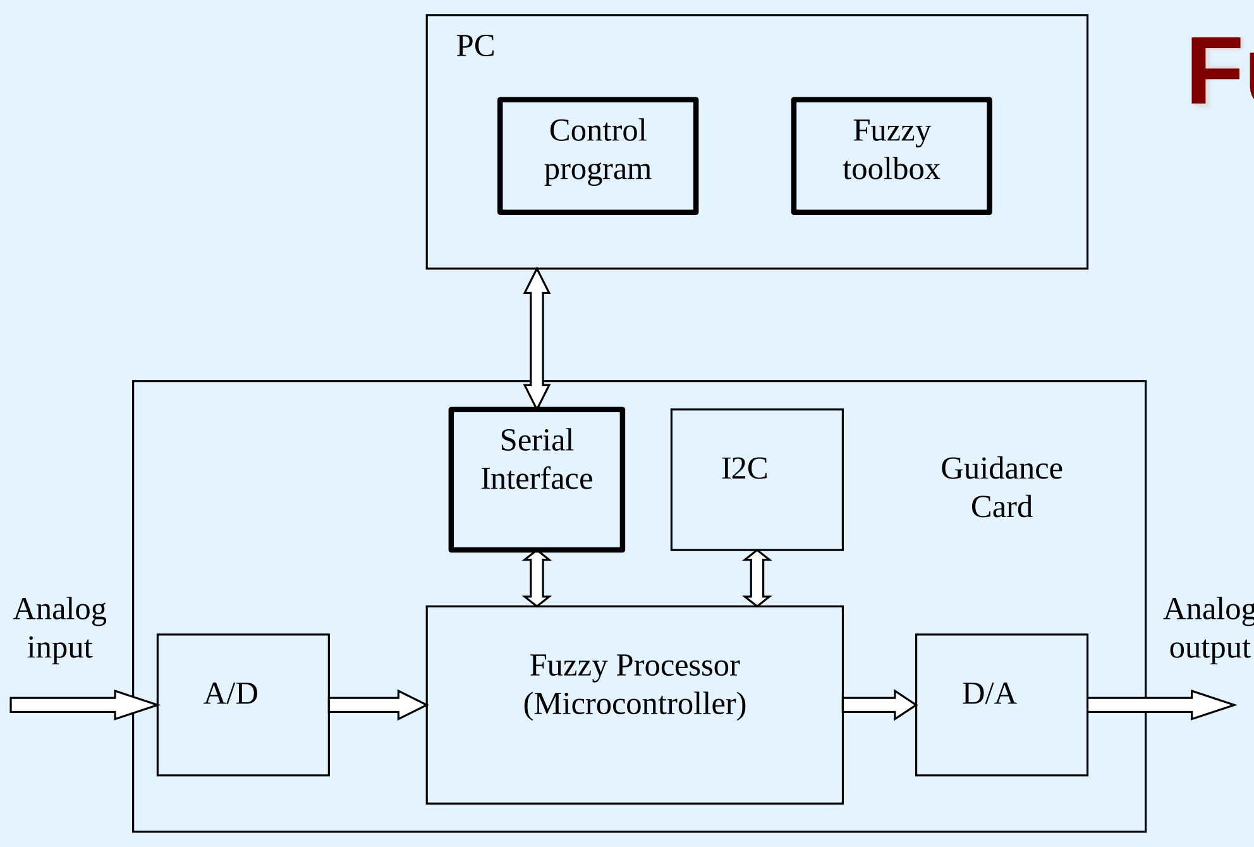

Figure 3: Proposed System Architecture

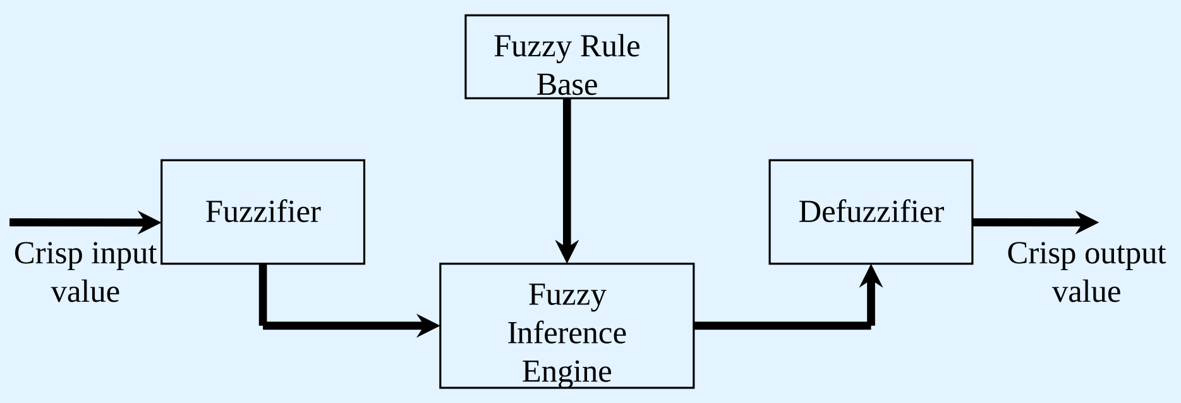

Figure 4, Configuration of fuzzy system with fuzzifier and defuzzifier

Implemented Fuzzy Algorithm

The fuzzy algorithm used in this paper is modified from conventional fuzzy algorithms to optimize the system performance by reducing computation operations. The following subsections describe each stage of the fuzzy algorithm.

a) Fuzzification: The input membership functions are defined in a look-up table based method. The universe of discourse of the input variables is limited to ten bits length (A/D resolution), ranges from 0 to 1023. The resolution of membership grade (denoted by μ) is 8 bit wide, ranges from 0 to 255. Thus, the value of input variables must be normalized to ten bits wide before running fuzzification process. Figure (5) shows an example of one input membership function defined in the look-up table. It takes 1024 bytes to record one membership function in knowledge base memory (KBM). Although the value of membership μ changes quite discretely, it is still acceptable, as fuzzy values do not require being too precise.

b) Fuzzy Inference Engine: A simple fuzzy inference technique is used. To record rules in KBM, for each rule, the membership of each input is stated and the corresponding required output membership is also stated. For example, say there is a system of 2 inputs and 1 outputs, one of the rules are as below:

If In1 is NM and In2 is PS, then Out1 is PM.

In the knowledge base, it is recorded as follow:

[In1NM] [In2PS] [Out1PM]

[In1NM] = 2-byte offset address of where the NM membership of input 1.
[In2PS] = 2-byte offset address of where the PS membership of input 2.
[Out1PM] = 2-byte offset address of where the PM membership of output 1.

The offset addresses are used to point to membership functions instead of using indexed number. Thus, there is no further calculation needed to locate the address where the memberships are recorded. This method reduces computation time in fuzzy inference process. Rules for different outputs are defined separately

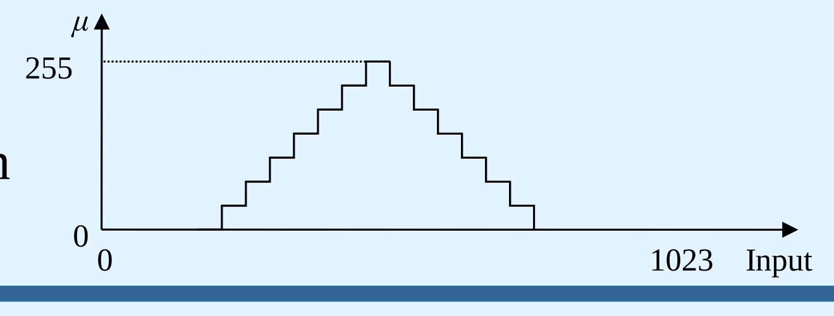

Figure 5, Discretization of the Membership function to store it in as Lookup Table

The value of each crisp input will be added to the given offset address to get μ of that input. Then, the minimum (denoted by α) among the μ values from the inputs is calculated. α will be used in aggregation process.

c) Fuzzy rule base: The fuzzy rule base is stored with the inputs and output membership functions in the knowledge base memory (KBM). Figure (6) shows the memory map of KBM. In this map, the address increases from top to bottom. From left to right, the map items are shown from the biggest group division to the smallest item details hierarchically. The rightmost column shows the memory space needed to store the data at the corresponding rows.

The first byte of the memory stores the number of rules for the output. The following parts are the actual fuzzy rule base definition. They are fuzzy rule table, input membership function table and output membership function table.

This memory organization creates flexibility for the user. The number of rules for can be up to 255. The number of memberships for inputs and outputs is not stated in the memory and can be defined freely by user, as the rule table will directly point to the offset address of the membership function.

d) Defuzzifier: After applying the rules, the output of the rules is aggregated using the max operation to get the aggregated membership function. The last operation in controller is the defuzzification process. The defuzzification method which used in this controller is the centroid method.

There are two integration processes that must be computed to can apply the centroid method. This integration processes can be simplified using the numerical method by taking a sample every degree in the output variable delta δ (fin deflection angle).

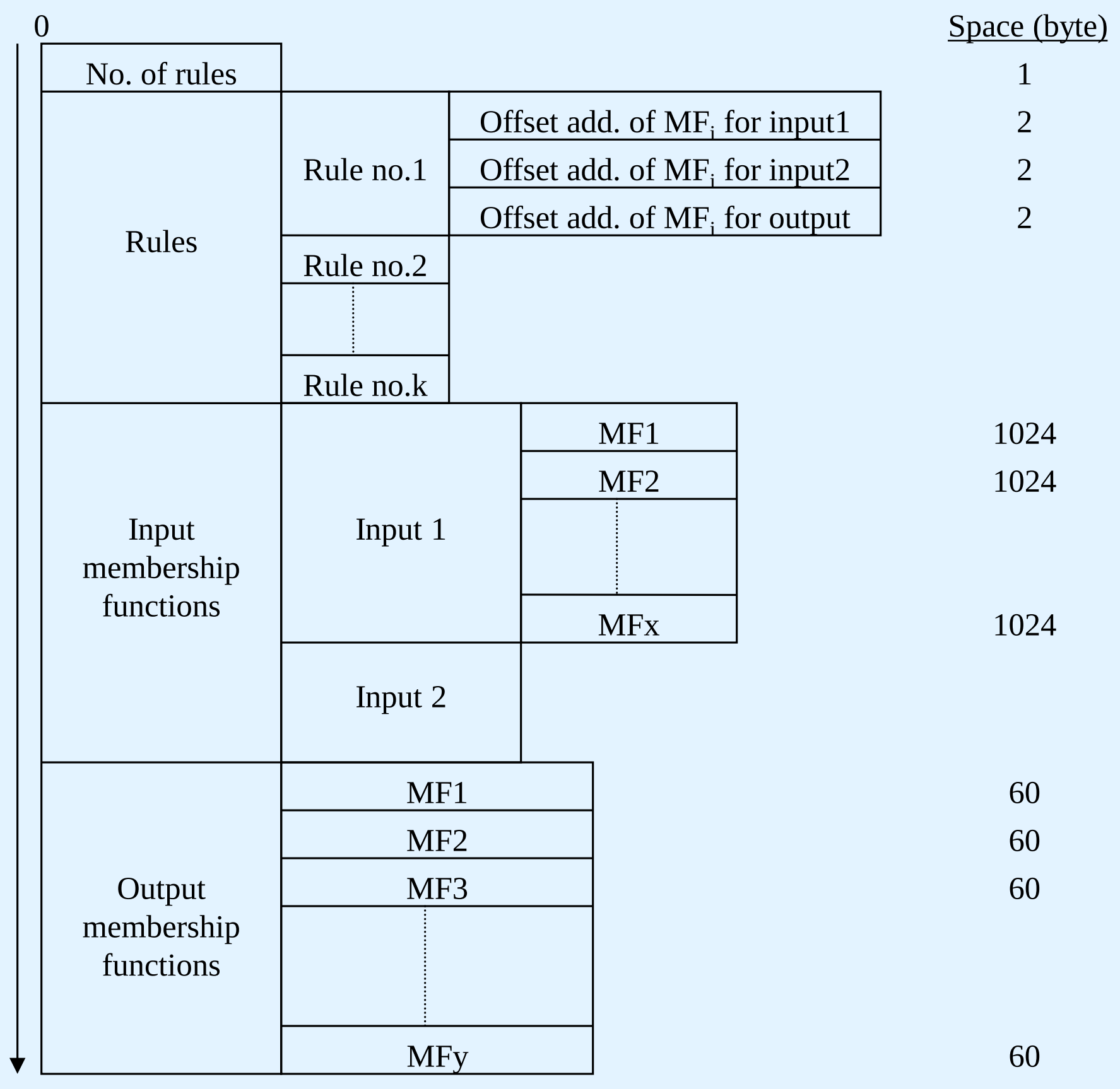
0
Space (byte)
No. of rules
1
Rules
Rule no.1
Offset add. of MF$_i$ for input1
2
Offset add. of MF$_i$ for input2
2
Offset add. of MF$_i$ for output
2
Rule no.2
Rule no.k
Input membership functions
Input 1
MF1
1024
MF2
1024
MFx
1024
Input 2
Output membership functions
MF1
60
MF2
60
MF3
60
MFy
60

Figure 6, The memory structure for knowledge base memory KBM

Table 2, The Fuzzy Guidance Rules

/ /	VS	S	MS	M	MB	B
VS	NM	PB	PB	PB	PB	PB
S	NS	NP	NM	PM	NP	NP
MS	Z	NS	Z	PS	NP	NP
M	PS	PS	PS	NM	NP	NP
MB	NM	PB	NB	NP	Z	NM
B	NS	NS	Z	PM	NP	NP

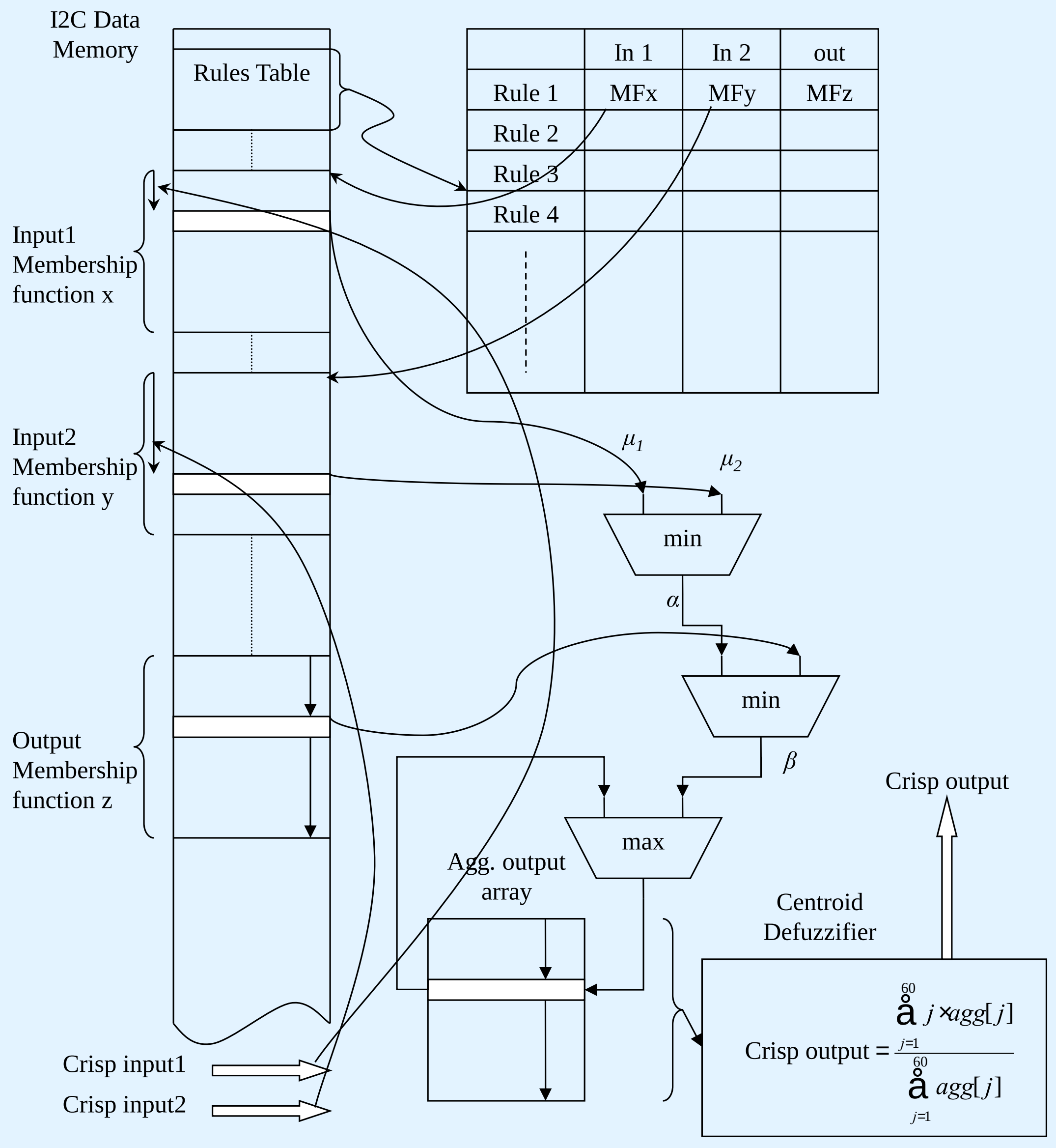

Figure 7, Graphical illustration showing how the fuzzy algorithm works. The

Results

The fuzzy knowledge base, or known as fuzzy inference system, is developed and tested with both Matlab's Fuzzy logic toolbox and JAVA programming language. The knowledge base stored on the I2C, which used in this implementation, is computed with JAVA and stored on the I2C via the serial connection between the PC and the guidance card.

The program which downloads the data and makes the serial connection is also developed using JAVA programming language

Conclusion

The fuzzy guidance card has been tested to be fully functional, including interfacing with computer, microcontroller calculations, A/D converter and D/A converter. The fuzzy guidance card implements the fuzzy algorithm efficiently. The proposed controllers in [1][2] could be implemented using a common hardware device, which reduces the cost of the controllers and increase its efficiency.

References

[1] M. Rizk, A. ElSayed, "AIR-TO-GROUND INTEGRATED FUZZY GUIDANCE SYSTEM", Ubiquitous Computing and Communication Journal (ISSN 1992-8424), UbiCC Journal, Volume 6: Issue 2.

[2] M. Rizk, A. ElSayed, "LEARNING AIR-TO-GROUND INTEGRATED FUZZY GUIDANCE SYSTEM USING TABU SEARCH", Ubiquitous Computing and Communication Journal (ISSN 1992-8424), UbiCC Journal, Volume 6: Issue 2.

[3]Costa, A., Gloria, A. De & Olivieri, M. (1996). Hardware Design of Asynchronous Fuzzy Controllers. IEEE Transactions on Fuzzy Systems. 4 (3); 328-338.

[4]Tan, Kok Leong (1998). Fuzzy Expert System in Navigation Control” Thesis B. Eng., Universiti Teknologi Malaysia.