Embedded Smart Controller for an Industrial Reefer Refrigeration[1]

Leon Reznik and Shane Spiteri
School of Communications and Informatics
Victoria University
PO Box 14428, Melbourne City MC VIC 8001 Australia

***Abstract**. The paper describes the development of a smart embedded fuzzy system for reefer refrigeration control. The goal of this project is twofold: to design an efficient and economically feasible intelligent industrial controller on one hand and to test and further develop a smart device design approach, which is based on design model formulation with the fuzzy rules given by an expert. Through prototyping and implementation on a general-purpose microprocessor the system can then be tested and used to iteratively identify and control the plant until the desired control system criteria are achieved. The paper describes the design, implementation and testing of an embedded fuzzy controller as a simple, practical solution to a regulation problem in reefer refrigeration systems. By an application of the Motorola HC12 MCU with its inbuilt fuzzy logic instruction set a rapid prototype design was implemented and the application of fuzzy logic and iterative identification and control to general industrial refrigeration systems was found to be a possible economically feasible solution.*

I. Introduction

With the development of system level integration and system-on-chip capabilities, the computer industry re-focuses itself from personal to embedded computing, which application in the industry produces new generation of smart (or intelligent) devices and products. Smart product design requires various combinations of high performance, low cost, and low power. Although the product technical and economic characteristics are very important, the design process itself becomes more and more critical in achieving a final triumph. The successful design and fabrication of an embedded controller gives a new product its distinguishing features and competitive advantage. Time to market has nowadays a huge impact upon a product life-cycle profits and can determine its success or failure [4]. On the other hand, the current lack of qualified designers threatens to cause a design bottleneck as the need for smart product design increases. So, the development of intelligent devices should be coupled with the advancement of intelligent design methodologies for such products.

The authors of this paper exploit the possibility of computational intelligence (fuzzy logic, neural networks) application for the embedded fuzzy controller design, fast prototyping and realisation. The proposed methods assume a formulation of an initial design with the help of fuzzy rules based on an information supplied by the expert, followed by its implementation on the general purpose microprocessor, which increases the design convenience and speed as unlike conventional methods, the proposed ones do not wait for an exact model to be developed. Neither they depend on a specialised microprocessor application, which reduces the design cost. The methods are advanced in two directions. The first one (see [2,6]) assumes the fuzzy system optimisation and tuning with neural networks for its implementation on a very simple and cheap (probably, 8-bit) microprocessor. The second one, investigated in this paper, assumes the possibility of optimisation through simple iterative identification and control methods. This method is based on the idea that the model and the control policy should be developed together and that no meaning shall be given to either one in isolation [7].

The paper is devoted to the design of an embedded fuzzy controller for an industrial reefer refrigeration system. In order to help reduce the cost associated with the transportation of refrigerated goods an economically feasible method to control the ever increasingly complex, reefer refrigeration systems needed to be found. Reefer is the term used in the refrigeration industry to describe a refrigerated container as shown in figure 1.

Figure 1. Reefer refrigeration system

[1] This is an extended version of the paper presented to the 10th IEEE International Conference on Fuzzy Systems “Meeting the Grand Challenge: Machines that Serve People”, Melbourne, Australia, December 2-5, 2001

The main requirement was to design a smart controller in such a way that modifications and further developments could be easily adopted to the existing design. In satisfying this requirement it was necessary to conduct extensive research in the refrigerated transport industry. Research into industry standards, typical solutions and customer demands helped to serve as a guide for basing the design. Existing reefer container controllers from incumbent companies such as Carrier, Mitsubishi and Daikin were found to be very expensive and offered only a limited range of features that did not always satisfy individual customer needs. Some of the features that the customers demanded were:

- remote modems - so that communication between the container and the head office can be established.
- upgradeable software – controllers should be able to download new software that enables a controller to work with different refrigerants and mechanical devices, eg. 4hp and 8hp compressors.
- downloadable data – insurance companies require shipping lines to accurately log temperature of the container over the duration of the trip.

The industrial orientation of this project meant that these features should be able to be designed into the prototype version of the reefer controller with minimal cost. To incorporate these features the design was focused on being universal. The universal controller should be able to operate effectively on any type of reefer container. A description of the refrigeration system is presented in section 2 as a typical regulatory problem. The design and development of the hardware circuitry necessary for implementation are discussed in section 3. Section 4 focuses on the application of fuzzy logic as a solution to the control system problem. Following this comes a description of the implementation of the developed fuzzy inference system.

II. Regulatory Problem

This section illustrates how an understanding of the refrigeration system operation was used to develop the control algorithm. As indicated from the section title the objective of the control system was to regulate the amount of superheat and maintain steady-state accuracy, simultaneously increasing efficiency whilst attaining operation under safe working conditions. A fuzzy inference system was applied to control an electronic valve position so that an optimal flow rate of refrigerant was fed into the evaporator. An optimal flow rate would ensure the refrigerant would become superheated by three degrees Celsius.

The problem could be likened to water flowing through a pipe being heated in a furnace [1]. Figure 2 illustrates that as the valve closes, the volume of water flowing through the pipe begins to decrease and the heat energy being absorbed from the furnace has a more profound effect on the water temperature than would be the case with a higher volume of water. The picture on the left shows a water tank with a standard valve tap attached to the bottom. As the valve opens, the flow rate increases. The water is heated by the furnace as it flows down the pipe. If the flow rate was too fast, the water would not heat up enough and would not turn into a gas. Alternatively if the flow rate was too slow then only half the pipe would be used and hence only operate at half the efficiency. An optimal flow rate would have the last droplet of water turn into a gas at the last section of piping. The efficiency would be maximised in this way. The more efficient a refrigeration system the faster it would adjust the temperature inside the container. If the flow rate was too fast the refrigerant would remain a liquid and flow into the compressor and destroy it. At a specific temperature (boiling point) the water would eventually become a superheated gas. More generally, the boiling point of any liquid depends on the pressure it is under at the time. If we imagine the furnace as the evaporator and the water as the refrigerant then the control aim is ‘to regulate the superheat of the refrigerant coming out of the evaporator’. This is achieved by controlling the actuating valve. Clearly the system was open-loop unstable.

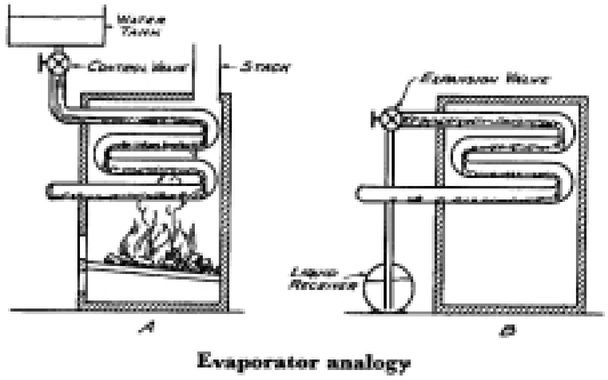

Figure 2. Water analogy used to understand control valve [1]

Another actuator that could be controlled by the controller was the modulation valve. This was a bypass valve. The hot-gas from the compressor bypasses the condenser and enters directly into the evaporator as the modulation valve was opened. The valve could be used to limit the capacity of the refrigeration system. By limiting the capacity of the refrigeration system control of the temperature inside the container could be maintained. The controller design incorporated the operation of the bypass valve by treating it as a disturbance input to the control system.

III. Conceptual model for the hardware design

The control system that was developed modelled the electronic expansion valve as the actuator and the refrigerant temperature as the plant to be controlled. The electronic expansion valve operation has been described in section 3.1. The aim of the electronic expansion valve was to control superheat. Superheat is determined by the following formula:

$$\text{Superheat } (°C) = \text{TDK } (°C) - \text{Boiling Point } (°C)$$

where TDK(°C) is the temperature read from the sensor located at the evaporator outlet. The boiling Point(°C) represents the vaporization point of the refrigerant for a given pressure. The boiling point was determined by reading the LP sensor value and then finding the boiling point for that refrigerant. The difference between the reference superheat and the measured superheat formed an error signal that was used as a control signal for the controller. If the superheat was found to be too high, corresponding to inefficient use of the evaporator coil, the valve was opened to increase the flow rate. If the superheat was too low then there was a risk that the refrigerant would exit the evaporator as a liquid and damaging the compressor. The valve opening would have been reduced to limit the flow rate and allow more heat to be absorbed by the refrigerant and hence boil into a vapour.

The prototype controller was designed to perform a number of functions apart from the fuzzy control algorithm so that some experiments could be performed. Some of the functions were namely to read in data from sensors, perform calculations and decision making, display information and control the operation of system devices. Figure 3 shows a conceptual view of the hardware blocks required for the prototype controller to function such as an amplifier for the stepper motor. Outlined in the box is the section that was implemented in the microprocessor. Feeding into the microprocessor is the LP transducer and the TDK transducer. Strictly speaking the feedback signal was not superheat. Superheat feedback was actually calculated within the microprocessor from the information interpreted from the TDK transducer and the LP transducer. The computer program was then used to measure if superheat equal the desired superheat of 3°C. The error between these two signals was fed into the controller. The second input into the controller was the derivative input. The derivative of the error signal was simply the change in error from the previous value. The microprocessor was a Motorola HC12 MCU with fuzzy logic instructions that lent themselves to a simple PID controller of this nature. The fuzzy logic instruction set allowed optimized code to be generated and furthermore the code was easily readable and made efficient use of the limited onboard memory [5].

IV. Smart Fuzzy Regulator

Fuzzy logic provided an appropriate solution to the otherwise complex task of mathematically deriving an exact model for the non-linear refrigeration system upon which conventional control techniques could then be applied. Skelton [7] discusses the inseparability between the modeling and control problems from both a mathematical and physical viewpoints. The application of fuzzy logic to this control problem treats the two as one entity and hence through an iterative approach an appropriate control solution can be established.

A subjective approach to controller design was adopted. Consultation with an expert refrigeration mechanic formed the basis of the choice of input and output membership functions and rules. A simple example to illustrate this point would be:

Rule1: If *superheat* is *low* then *decrease* valve position.

Rule2: If *superheat* is high then *increase* valve position.

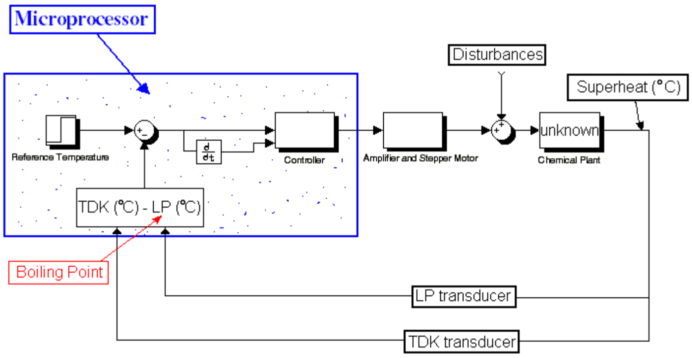

Figure 3. Conceptual model of the fuzzy PID-like hardware system

The fuzzy inference system was designed to act as a PID-like controller. A basic PD-like controller, with incremental output, was designed. Two inputs were used: (1)superheat and (2)change of superheat. Measurement noise, due largely to sensor error and A/D conversion resolution, was reduced by a simple but effective means of sample averaging. Without such a technique the derivative portion of the PID-like controller was too dominant [8]. Integral action is determined to be a continual change in actuation as a result of steady-state inaccuracy. The output controlled the change in valve position between each sample instant. This continual changing of valve position formed the integrating action of the fuzzy PID-like controller. Rules developed by the refrigeration expert for the control of the refrigeration system helped determine what inputs and outputs were most appropriate for the controller.

Based on the experience the refrigeration mechanic (the expert) has gained through the manual control of refrigeration systems, a linguistic description for the control of the electronic expansion valve was attained. This description formed the rules for which to develop the fuzzy inference system. The rules are outlined below:

1. If (SuperHeat is Low) and (dSuperHeat is Neg) then (TX~Output is Dec) (1)
2. If (SuperHeat is Low) and (dSuperHeat is No~Change) then (TX~Output is small~dec) (1)
3. If (SuperHeat is Low) and (dSuperHeat is Pos) then (TX~Output is Nothing) (1)
4. If (SuperHeat is OK) and (dSuperHeat is Neg) then (TX~Output is small~dec) (1)
5. If (SuperHeat is OK) and (dSuperHeat is No~Change) then (TX~Output is Nothing) (1)
6. If (SuperHeat is OK) and (dSuperHeat is Pos) then (TX~Output is small~inc) (1)
7. If (SuperHeat is High) and (dSuperHeat is Neg) then (TX~Output is Nothing) (1)
8. If (SuperHeat is High) and (dSuperHeat is No~Change) then (TX~Output is small~inc) (1)
9. If (SuperHeat is High) and (dSuperHeat is Pos) then (TX~Output is Inc) (1)

Figure 4. FIS rules developed by expert knowledge

The linguistic rules were converted to Fuzzy rules. The Fuzzy rules served to describe a qualitative relationship between the three variables [5]. The two input variables were ‘Superheat’ and ‘Delta Superheat’ and the output variable was ‘Change in Position’.

Expert knowledge was used to determine the fuzzy set for the input ‘superheat’. Three basic membership functions were chosen as an initial attempt to describe each input, being *low*, *okay* and *high*. The shape of each membership function was limited to the trapezoidal (triangular) type being the only type definable by the HC12 fuzzy instruction set. The universe of discourse of the fuzzy set is the range of values of superheat expected to be input into the system. It was found to be between 1°C and 5°C. *Negative*, *No change* and *Positive* were the labels of each membership function for the input ‘delta superheat’. The universe of discourse was given as -1°C to +1°C. The immediate slope of the negative and positive membership functions minimise the effect of noise, inherent in the ADC process. Alterations to the membership functions in the program code could be easily implemented and details of which are given in the next section.

V. Operation of the Fuzzy Inference Kernel

Motorola’s fuzzy instruction set allowed the FIS to be programmed with minimal code and execute very quickly. The HC12 FIS consisted of two parts. The first part was the knowledge base where the membership function definitions and rules were defined. The second part was the fuzzy inference kernel. This was programmed as a C function that accepted a crisp input and calculated an appropriate output action for the SISO system. Some constraints, inevitable for a general purpose microprocessor, were the restriction to the use of a Sugeno-type FIS and limited membership function definitions.[3]

Programming the knowledge base was a simple matter of converting the membership functions to data arrays and storing them in ROM. Each membership function is stored as a four byte array. The first and second elements tell the FIS the point of departure from the x-axis of the membership function plot and the point of return. The last two elements of the data array indicate the positive and negative slopes of the membership function. Special characters were used to indicate more complex membership functions. For example an infinite slope would be represented by the character $00 since this would otherwise be unused.

```
.area text
; Format is X0, X1, Slope0, Slope1.
; Super Heat Input Membership Function
;Definition
s_def::
 .byte  $00,$7F,$00,$04 ;Super Heat Low
;NOTE:$00 means infinite slope
 .byte  $3F,$BF,$04,$04    ; Super Heat OK
```

Figure 5. Program code for membership function definitions

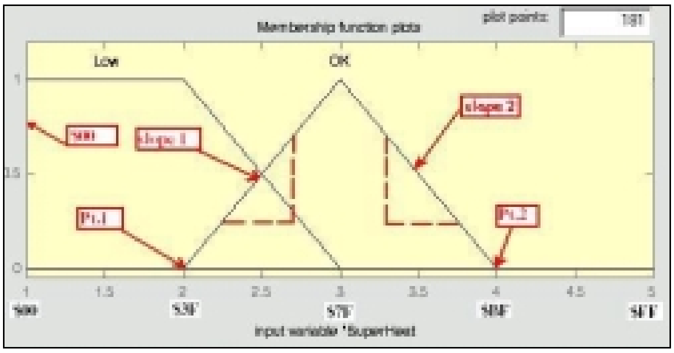

Figure 6. Membership function definition for HC12

Figure 6 shows two simple membership functions and figure 5 shows how the membership functions are implemented on the Motorola HC12 MCU. The slope 1 and slope 2 of the membership function were calculated in hexadecimal as shown in equation 1. The Fuzzy Instruction set uses 8-bit values and hence the maximum height of the membership function is 8bits, or 255 decimal.

$$Slope1 = \frac{255}{(127-63)} = \frac{256}{64} = \frac{\$FF}{\$40} = \$04 = 4 \quad (1)$$

When defining the rule base the first elements in the array, the antecedents, are logically AND’ed together. To separate the antecedents from the consequents the character $FE was used. Each rule was evaluated with a single instruction until the end of the rule table was reached, being signified by a character $FF.

```
; Define rules
rules::
       .byte    low,neg,$FE,dec,$FE
       .byte   low,no_change
               ,$FE,small_dec, $FE
       ….        …….
       .byte   $FF
```

Figure 7. Implementation of the rule table on a HC12

With the knowledge base defined the fuzzy instructions are put to use. The fuzzy instructions that are used by the HC12 are *MEM, REV and WAV.* These three instructions were included in the fuzzy function. The input that was passed to the fuzzy routine had to be scaled first into the FIS universe of discourse. Then for a given input the instruction MEM is called for each membership function. This will automatically calculate the degree of truth of each membership function associated with the 8bit input value passed to the fuzzy C routine. A single instruction REV would evaluate each rule defined in the knowledge base resulting in a data array of output values for each of the rules being evaluated. The WAV instruction converted the data array of fuzzy outputs into a single crisp output using equation 2. Five singleton fuzzy outputs were chosen and easily implemented into program code as one simple array of five 8-bit numbers each representing the location of the singleton function.

$$SystemOutput = \frac{\sum_{i=1}^{n} SiFi}{\sum_{i=1}^{n} Fi} \quad (2)$$

where n = number of output labels in the system
S = singleton position of each label
F = calculated fuzzy output values in RAM

To test the fuzzy routine a known crisp input was forced into the routine and the output was compared with that produced by a fuzzy simulation package such as MATLAB™. Being able to compare results from two sources served to help identify any programming errors through unexplained discrepancies between outputs.

Beginning with a relatively simple model the system was stabilized. Performance objectives could then be achieved through repeated testing and experimentation. Changes to any part of the fuzzy system could easily be implemented, as discussed in this section, and hence a controller appropriate for the control objective at hand could be developed after several iterations of the control algorithm design.

VI. Conclusion

The application of fuzzy logic in reefer refrigeration control systems was useful in determining a solution to the regulation control problem. The fuzzy controller was able to maintain a superheat at 3°C.

Fuzzy logic systems were found to be very flexible and easy to comprehend and hence appropriate for a subjective solution such as that presented. Fuzzy logic allowed the modelling and control problem to be treated simultaneously and combined with the speed and ease in which the fuzzy system could be developed allowed an iterative solution to present itself. Fuzzy Logic was proved to be a very cost-effective and practical alternative to conventional methods.

The application of fuzzy logic and iterative solutions could be used in many other control systems where a cost effective and practical solution is required. Any system where the operation can be described in linguistic terms would benefit from the application of fuzzy logic, especially if the system changes dynamically or is highly non-linear.

References

[1] King, G.R. “Basic Refrigeration”, 3rd edition, Nickerson & Collins Co., ILL, pp.1-47, pp.53-71, 1971

[2] Little, A. and L. Reznik “Implementation of Fuzzy Controllers with Radial Basis Neural Networks”, The Ninth IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2000 San Antonio, USA, 7-10 May 2000 Proceedings IEEE Vol. 2, pp.581-586, 2000

[3] Motorola HC12 manual section 9

[4] Rau, B.R. and M.S. Schlansker “Embedded Computer Architecture and Automation”, Computer, vol.34, No. 4, pp. 75 – 83, April 2001

[5] Reznik, L. “Fuzzy Controllers”, Newnes-Butterwirth-Heinemann, Oxford-Boston, 287pp., 1997

[6] Reznik, L. and A.Little **“**Choice of the Radial Basis Function Approximation in Neural Networks Used for Fuzzy System Implementation”, Joint 9th IFSA World Congress and 20th NAFIPS International Conference, July 25-28, 2001, Vancouver, Canada, Proceedings, IEEE, p.3032 - 3037

[7] Skelton, R.E. “On the structure of modelling errors and the inseparability of the modelling and control problems”,IFAC Model Error Concepts and Compensation, 1985.

[8] Valvano, J.W. “Embedded Microcomputer Systems”, Coles/Brook, CA, 2000