Development of Data Acquisition and Control System for Electrothermal Processes, Based on the Structure of Fuzzy Knowledge – Based Controller

Ivan G. Babatov [1]

Abstract – **This paper presents the development of Data acquisition and control system, based on the structure of Fuzzy Knowledge- Based Controller (FKBC). The system is intended for testing the applicability and efficiency of a new software model of a fuzzy system, built according to the principles of the Object- Oriented Programming.**

Keywords – **Fuzzy control system, FKBC, Embedded control system, Object-Oriented Programming.**

I. INTRODUCTION

The control systems, utilizing the principles of the Fuzzy Set Theory and Fuzzy Logic have been a matter of interest for many scientists and engineers during the past decade. It has been proved, that in many cases these systems perform better than the conventional ones and can cope with nonlinearities and run-time changes in the controlled object's properties that normally render the control system ineffective.

In spite of their benefits, these systems still are not widely used, due to a variety of reasons, spreading from classical control scientist's mistrust to purely mathematical stability problems that may occur in some algorithmic implementations of such systems.

One of the basic problems that one meets while designing a fuzzy system to be used in practice, i.e. not only in theory is that of converting the continuous equations into discrete ones and optimizing them for use with integer arithmetic, so that they can be implemented in a microcontroller unit and run concurrently in real-time conditions. This process follows the symbolic-to-meaning translation process (identified by the dotted arrow in Fig. 1).

Normally this problem is solved by simplifying the model of the system and its conversion to a set of tables and a simple algorithm for mapping a vector of output values to each vector of input values according to the entries in the tables.

Although being computationally effective and suitable for automation, this method has the disadvantage of highly reducing the information, built in the rulebase, since not the rules and the linguistic representations of the process variables are stored, but just a set of numbers, calculated according to them. Thus the basic form of the expert knowledge is stripped from the system prior to its implementation.

The basic goal of our research is to find a new method of translation that will allow for the linguistic information to be stored in the embedded system, rather than just used for creation of the tables and input- output mapping rules.

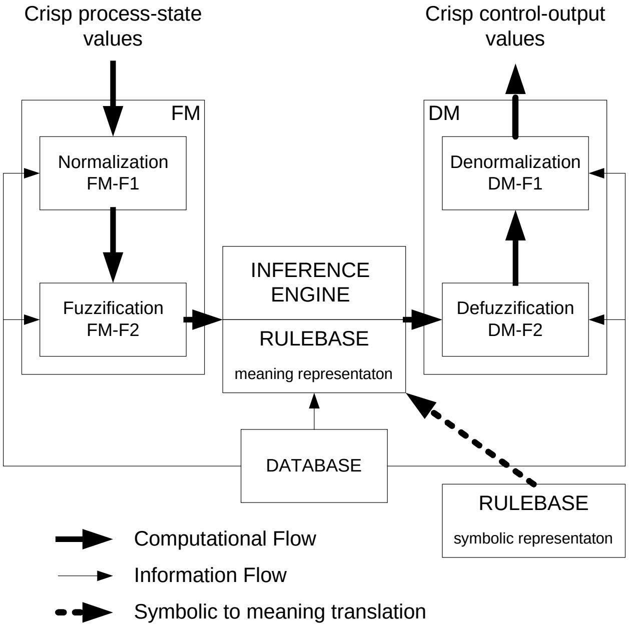

Fig. 1. The structure of a Fuzzy Knowledge-Based Controller [1]

The research is done in two stages:

1. Development of an Object-Oriented model of the Fuzzy Set Theory and Fuzzy Logic's basic concepts and its use for implementation of Fuzzy Control Systems;
2. Development of prototype embedded data acquisition and control system for model testing and verification.

The basic structure of the class hierarchy of the model is presented in [2]. This paper presents the structure and basic characteristics of the prototype embedded data acquisition and control system.

II. SYSTEM DEVELOPMENT

The system of discourse is intended for use for data acquisition, monitoring and control of electrothermal processes. It can monitor and control a variety of plants, spreading from domestic heaters and air- conditioning systems to induction- heating equipment. This wide spectrum of target devices is needed in order to test the applicability and reliability of the designed program model. Along with the relatively high computational complexity, implied by the model this leads to the necessity of separating the low- level data acquisition and control tasks, such as analog- to digital conversion of the process variables and waveform generation for the control output signals from the main control algorithm.

---

[1] eng. Ivan G. Babatov is a PhD student in the Department of Electronics, Technical University of Gabrovo. 5300 Gabrovo, 4 H. Dimiter str., Bulgaria. E-mail: babatow@yahoo.com

The need for on-line monitoring of the process and control activities in its turn require intense communication between the system and a host device, such as personal computer or industrial network concentrator.

The requirements, stated above lead to the idea of building several autonomous subsystems, each managing a distinct set of low- level tasks, and controlled by a kernel module (or subsystem) which carries out the main control algorithm and system-wide synchronization functions. The structure of the developed system is shown in Fig. 2.

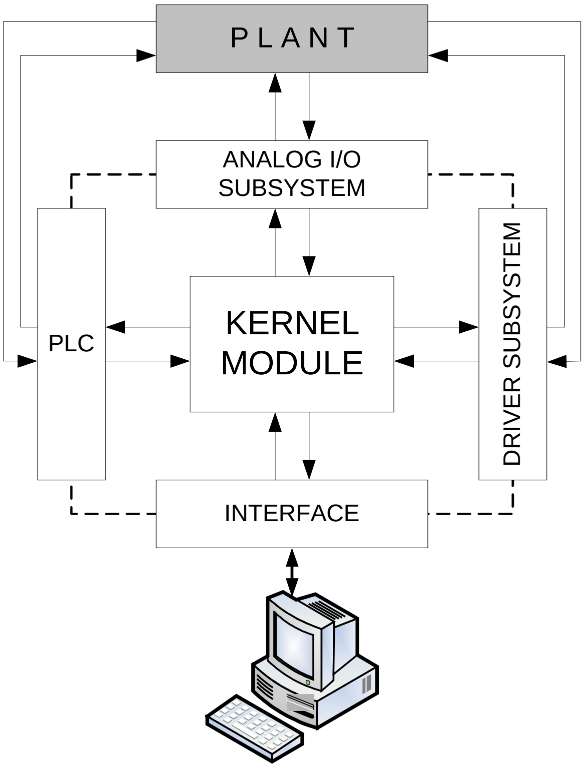

Fig. 2. System architecture

As shown, the system consists of four autonomous subsystems, controlled by a kernel module. The subsystems are as follows:

- Analog Input- Output subsystem;
- Driver subsystem;
- Interface subsystem;
- Programmable Logic Controller (PLC) subsystem.

The kernel module is the one that implements the structure of the Fuzzy Knowledge- Based Controller (FKBC), presented on Fig. 1 (redrawn from Drjankov *et al.* [1]). The controller consists of two- stage **fuzzification module** (FM), which basic functions are *input normalization*, carried out in the FM-F1 stage, which maps the physical values of the process variables into a normalized universe of discourse (normalized domain), and *fuzzification* (in FM-F2 stage), which converts a point-wise (crisp) current value of a process variable into a fuzzy set, in order to make it compatible with the fuzzy set representation of the process state variable in the rule- antecedent. These actions in the kernel are actually performed by instances of dedicated classes (TInputValue), responsible for handling the values of input variables.

The inverse operations are carried out in the **defuzzification module** (DM), which also consists of two stages: DM-F2, which performs the *defuzzification*, which converts the set of modified control output values into a single point-wise value. The second stage, DM-F1 performs an output *denormalization*, which maps the point-wise value of the control output onto its physical domain. The actual processing for the defuzzification module is carried out by instances of the TOutputVariable class from the model.

The **knowledge base** is the basic data repository of the FKBC. It consists of *database* and *rule base*.

The basic function of the database is to provide the necessary information for the proper functioning of the fuzzification module, the rule base and the defuzzification module. This information includes the membership functions, representing the meaning of the linguistic values of the process state and control output variables and the physical domains and their normalized counterparts together with the normalization/denormalization (scaling) factors.

In the current implementation these functions are built-in the very representation of the variables by means of instances of the classes, representing the concepts of fuzzy set and linguistic variable.

The rule base represents in a structured way the control strategy of an experienced process operator and/or control engineer in the form of a set of production rules in the form

$$if \ \langle process\_state \rangle \ then \ \langle control\_output \rangle \qquad (1)$$

The *if*- part of the production rule is called the *rule antecedent* and the part following the *then – rule consequent*. In the current implementation of the kernel the rule base is presented by a list of instances of the class, representing the production rule concept.

The **inference engine** is the part of the controller that mimics the human reasoning. It is here, where the input values are combined with the knowledge from the knowledge base and in logical way produce the output values. There are several types of fuzzy inference with Mamdani and Gödel being the most common. The model that will be tested with the system has means of representing both of them by using various inference classes.

Along with the main control algorithm, the kernel module performs various system- level functions as:

- System power-up testing;
- Nonvolatile memory (DataFLASH) management;
- Real- time clock tracking;
- Basic communication functions;
- Initialization and control of the autonomous subsystems;

The subsystems are connected to the kernel module via high- speed serial system bus, utilizing up to 1Mbps throughput between any two devices. At every instant of time only one subsystem is enabled to communicate with the kernel and their simultaneous communication is performed by means of time- sharing algorithm, run by the kernel module.

Each subsystem, however, has a means of generating interrupt requests to the kernel and the priorities of these requests are programmable, so that in different applications different time- sharing and response strategies can be implemented.

The kernel subsystem is based on the ATMega128 High-performance RISC microcontroller unit, produced by ATMEL Corporation.[3] This MCU has 8-bit architecture, utilizes 128KB of reprogrammable program memory, 4KB on-chip data RAM, and is capable of driving up to 64KB external RAM. This allows for creation of memory-intensive dynamic data structures, such as lists and object collections, and also provides enough space for buffering the incoming data from other subsystems before sending it up the communication interface to the host device.

The kernel subsystem is equipped with 4MB of non-volatile data memory (a DataFLASH® device) which allows for off- line monitoring of the system operation process. Depending on the amount of data and the frequency of recording, this device is capable of storing information about the system activities for more than a week of continuous operation.

The system is equipped with real- time clock that keeps track of the astronomical time, and, if programmed to, will form the time- base for the operating system task switching.

The kernel runs real- time multitasking operating system, also built upon the principles of the Object- Oriented Programming. This operating system allows for virtually simultaneous execution of several tasks of various priorities, and is the basic feature that secures the operation of the entire instrument.

All four peripheral subsystems are built upon the ATMega32 microcontroller units, produced by ATMEL Corporation. Their functions are as follows:

The **analog input- output subsystem** structure is shown on Fig.3. It performs the low- level data acquisition and analog control functions. It consists of eight programmable gain single- ended voltage measurement channels with ranges from (0÷10) mV, (0÷50) mV, (0÷100) mV, (0÷500) mV, (0÷1000) mV, (0÷5) V and (0÷10) V. Additional adaptors can be used for measurement of resistance and/or cold- end compensation of thermocouple sensors if needed.

The voltages are normalized and filtered by analog circuitry, and digitized by 12-bit analog-to-digital converter. The samples are then packed into blocks and sent to the kernel on request.

This subsystem also has 4- analog outputs, produced by 12-bit digital-to-analog converter. The outputs are passed through programmable gain amplifiers, thus capable of producing voltages within the ranges (-1 ÷ 1) V, (0 ÷ 1) V, (-5 ÷ 5) V and (0 ÷ 10) V. The maximum load of each analog output is 1A.

The **driver subsystem** (Fig. 4) contains the circuitry, needed to control two independent solid- state pulse-width modulated (PWM) outputs, full- bridge autonomous inverter and controlled monophase bridge rectifier. This subsystem includes all the circuitry for obtaining the feedback signals from the controlled devices, and thus is capable of operation as autonomous servo- control system.

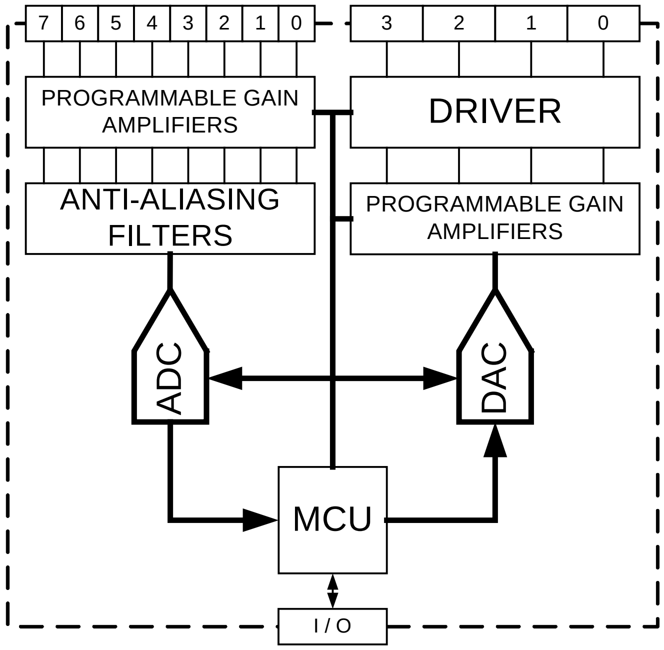

Fig. 3. Analog Input – Output Subsystem architecture

The distribution of the control signals and protection mechanisms are highly configurable due to the use of Complex Programmable Logic Device (CPLD) XC9572, produced by XILINX Corporation.

The driver subsystem is designed to act as a stand- alone programmable controller, capable of driving single electrothermal process. In this way it is possible to set the kernel module to supervisory control only, which will allow for testing variety of indirect control modes of the fuzzy knowledge- based controller.

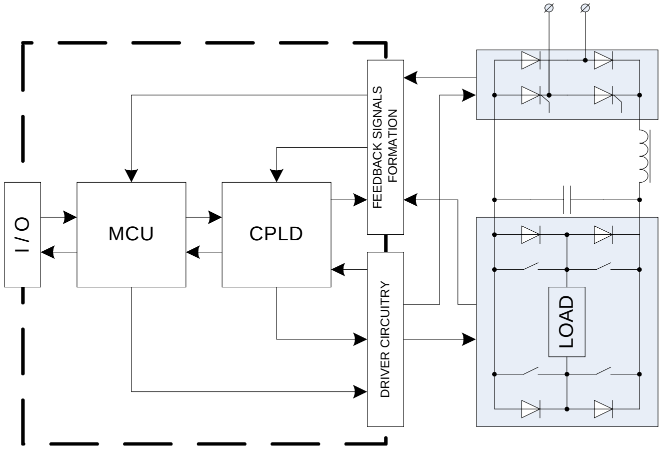

Fig. 4. Driver subsystem architecture

The **programmable logic controller (PLC) subsystem** is basically digital input – output system, capable of running its own control algorithm, along with serving the kernel subsystem. It is designed to extend the control capabilities of the kernel as well as performing stand-alone non- fuzzy control on certain parts of the plant. The control algorithm for the PLC is coded in simplified statement language (STL) form. At the programmer's disposal are logic, basic arithmetic and trigger functions, as well as timers and two independent analog inputs. The PLC program is loaded into the subsystem by the kernel unit and is stored in a battery backed- up memory.

The **interface subsystem** provides the user interface to the system. It consists of 4 line LCD display and a control keypad.

The interface subsystem implements insulated RS-232 interface to personal computer, which is used to perform process monitoring, system optimization and control. A front-end PC application for system maintenance is in a process of development.

III. CONCLUSION

The prototype system, presented in this paper is still under development, but most of its hardware modules are designed and are in process of manufacturing. The real- time operating system, which the kernel and the peripheral subsystems will run is developed and tested on a prototype controller, built on the ATMega8 microcontroller unit.

After its complete development, the system will have the following basic features:

1. Capability of control and monitoring of various electrothermal processes, spreading from domestic heaters to induction – heating systems.
2. Capability of control and monitoring of distributed technological processes by means of autonomous intelligent peripheral units, connected in industrial network.
3. Capability of on-line and off-line monitoring of the process from host device such as personal computer or network concentrator. The process data will be exported in format, suitable for further analysis by means of dedicated software applications such as MATLAB.
4. The basic goal of the system is to verify the applicability of the object – oriented hierarchical model of the fuzzy set theory and fuzzy logic that was developed and to test its efficiency, compared to the conventional control methods.
5. The presumable possibilities for integration of module for self- optimization in the model implementation will be tested upon the system.
6. The possibilities for application of this control system architecture to other types of processes (such as motion control) will be tested.
7. The system will allow for testing of various control algorithms of a certain plant, with the possibility of off-line analysis of the results. This will help in optimizing the control algorithm to minimize the computational overhead and hardware requirements. The results of this optimization can be used in simpler controllers for commercial purposes.
8. As a result of the previous item, the system can be of use in the education process in control engineering, data acquisition and microprocessor systems.

REFERENCES

[1] Driankov D., H. Hellendoorn, M. Reinfrank, *An Introduction to Fuzzy Control*, Springer-Verlag, 1996 (2nd ed.)

[2] Babatov I., *Object- Oriented Model of the Fuzzy Set Theory for Direct Implementation of Embedded Control Systems*, UNITECH'04, *Confence Proceedings - Volume I*, pp. I-213 - I-218, Gabrovo, Bulgaria, 2004.

[3] *ATMega128 - 8-bit Microcontroller with 128K Bytes In-System Programmable Flash*. Datasheet. 2467E–AVR–05/02. Atmel Corporation, 2002.