A Software Package for Fuzzy Control Design on a Laptop Computer

Masae Kanda*, Shogo Tateishi**, Shinji Hayashi***
Mika Araki****, Takehiko Tomita*****

* Industrial Instrumentation & Control Systems Dept., Fuchu Works, Toshiba Corp., 1, Toshiba-cho, Fuchu-shi, 183, Tokyo, Japan
** System & Software Engineering Lab., Toshiba Corp.
*** Toshiba Research & Development Center, Toshiba Corp.
**** C & I Product Marketing Dept., Toshiba Corp.
***** Plant Systems Engineering Dept., Toshiba Corp.

Abstruct

In coming years fuzzy control techniques will be increasingly applied to plants in such fields as industry and public works. For this reason, it is extremely important that a high-quality support tool for systems configuration be developed.

In this paper we will describe a comprehensive, user-friendly design tool for configuration of fuzzy control systems, called FZtool, that runs on a compact and cost-effective laptop computer.

1. Introduction

The concept of fuzzy logic was proposed by Zadeh in 1965 [1]. In the 1970's, Mamdani applied it to control systems [2]-[6] and in 1980 the first practical application was tried by the F. L. Smidth Corp. [7]. Since then, many have studied the practical application of fuzzy control and fuzzy controllers in plants. See references [8] and [9] for a brief and detailed survey of control applications using the fuzzy logic concept.

The fascination of fuzzy logic exists in its facility as an interface between linguistic and digital information [10]. Fuzzy production rules, such as the IF-THEN operator, and the capability for approximate matching add to this fascination. Fuzzy control can be very useful in utilizing the know-how of operators in plant situations and thus contribute to the automation of manual plant operations. Moreover, fuzzy control is capable of dealing with nonlinear processes easily since it is not necessary to specify an accurate mathematical model of the process to be controlled.

A variety of products have been developed based on fuzzy logic, including a number of fuzzy controllers [12]. Now there is a compelling need for a fuzzy control configuration tool that is easy to use and that provides easy tuning. Two major requirements for easy configuration of fuzzy control systems are:

(1) A savings in engineering time and manpower resources
(2) The ability to realize a just-fit system for a particular specification.

To meet these needs we have developed FZtool, described in this paper. This section has introduced FZtool. Section 2 describes FZtool architecture, Section 3 concerns FZtool main functions and conclusions are presented in Section 4.

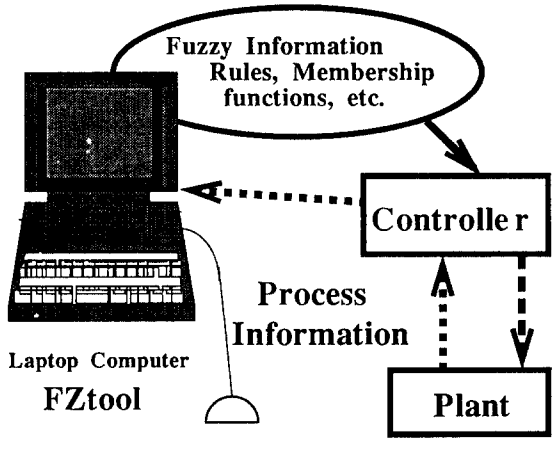

Fig. 2.1 FZtool standard system configration

2. System Architecture

FZtool is a software package that runs under MS-DOS on an IBM PC/AT compatible computer such as the J-3100 laptop series. The FZtool standard system configuration is shown in Fig. 2.1.

FZtool supports fuzzy control from design to testing. The main functions of this software package are:

(1) Design
(2) Simulation
(3) Monitoring
(4) Tuning

The design includes definition of I/O parameters, membership functions, rules and scaling factors. The design function includes compilation for generating executable code.

The simulation functions allows the rules composed with the design function based on a user process control model to be simulated.

The monitoring function collects process data for on-line monitoring of plant data and preparation of data for tuning.

The tuning function provides a means for modifying defined membership functions and scaling factors in order to tune the design to the application.

The FZtool System Window screen is shown in Fig. 2.2. It is divided into a command area at the top, a menu and data display area in the midsection, and a message and help area at the bottom.

Commands are expanded through pull-down menus, as shown in Fig. 3.1. For example, to access the subcommands associated with Design, just click the mouse on Design in the command area. A pull-down menu will appear in the menu area. (If a mouse is not available, FZtool can be operated from the keyboard.)

FZtool communicates with the controller through an RS-232C port. If only one serial port is available and is used to connect to the controller, FZtool can still be operated from the keyboard.

3. Main functions

3.1 Design

The Design function has six subfunctions. They are listed below with their associated menu commands shown in brackets:

(1) Definition of inputs/outputs [inputs-outputs].
(2) Definition of membership functions [membership].
(3) Rules creation [rules]
(4) Definition of scaling factors [scale]
(5) Compile [compile]
(6) Assemble [assemble]

The Design subcommand menu, accessed by clicking the mouse on Design in the command area, is shown in Fig. 3.1.

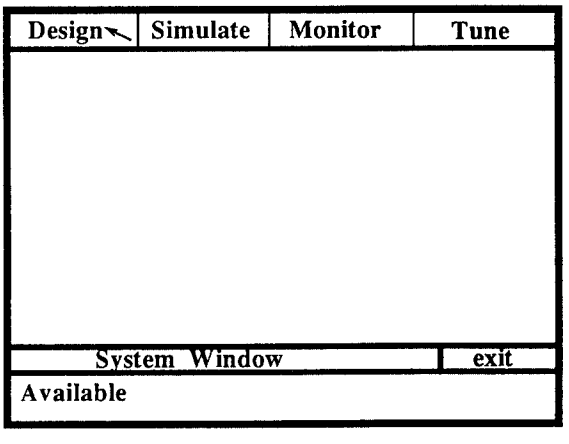

Fig. 2.2 System window screen

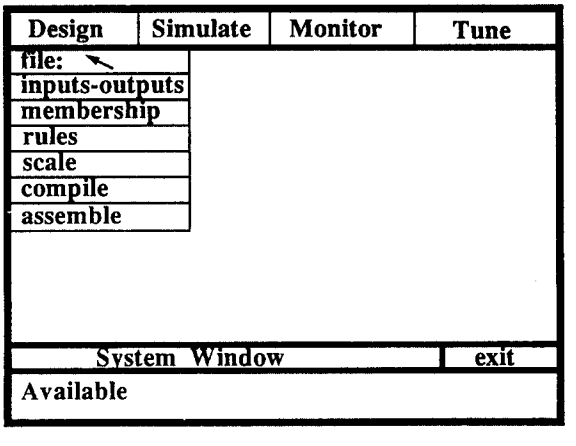

Fig 3.1 Sytem window after clicking "Design"

Before descibing the Design subcommands, a note about scaling factors in fuzzy control. The scaling factor is a kind of normalization parameter. Scaling factors for input variables match a range between input parameters and the controller. Similarly, scaling factors for output variables match a range between controller outputs and the object being controlled. In Fig. 3.2 the "k_is" need are the scaling factors for inputs, and the "k_os" are the scaling factors for outputs.

Fig. 3.2 Diagram of fuzzy controller

Inputs-outputs

Once a file name has been entered at the "file:" subcommand, as shown in Fig. 3.1, clicking the mouse on the "inputs-outputs" subcommand will call up the screen shown in Fig. 3.3. A shown in Fig. 3.3., inputs and outputs are defined by filling in the "Name," "Range" and "Method" columns of the display. Only outputs can have a method specified. This can be either defuzzification by the center-of-gravity method or by the simplified (weighted average) method. In the latter case, the THEN-clause membership function is given by a crisp value, so defuzzification is not necessary. Use of a crisp value as a membership function is often the case in control applications.

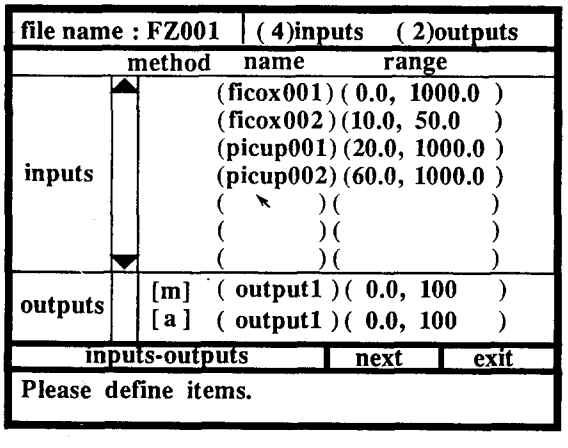

Fig. 3.3 Inputs-outputs screen

Membership

Fig. 3.4 shows the screen that is called up when the "membership" subcommand is selected. FZtool supports five membership function types. They are "value," "blinp, " "blinz," "blinn" and "trap." Each of these is shown in Fig. 3.4.

Selecting a type is easy. Just click one of the brackets in the "type" column. One of the membership types will appear within the bracket. Clicking the mouse will cycle through the five membership function types. When the type you want appears, select it by pressing the CR (carriage return) key.

The "value" type means a crisp value. If the simplified method was chosen for an output on the inputs-outputs screen (see Fig. 3.3), then only "value" can be selected as its membership function type. In general, FZtool won't allow invalid selections. For example, you cannot select and define inputs-outputs until you have specified a file name with the "file:" subcommand (see Fig. 3.1). This is an FZtool feature that helps make operation of this software tool simple and comprehensive, even for beginners.

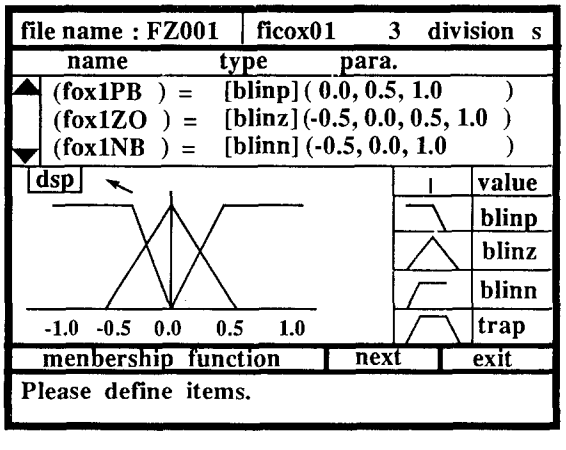

Fig. 3.4 Membership screen

Rules

Fig. 3.5 shows the screen called up when the "rules" subcommand is selected. This screen displays the composition rules for your design based on inputs-outputs and membership functions. Clicking the brackets for input variables as shown in Fig. 3.5 will cycle through the input variable names defined on the inputs-outputs screen (see Fig. 3.3). Similarly, clicking the brackets for membership functions will cycle through the defined membership functions and output variable names.

FZtool supports a "NOT" modifier. The smaller brackets shown in Fig. 3.5 is for selecting this NOT modifier.

Clicking the "compile" or "assemble" subcommands under the design command will produce as screen for each of these commands. The compiling generates macrocodes and the assembling generates microcodes.

A simulation example is shown in Fig. 3.6. Once the start and end time are set, trend graphs showing the input (process) variables are displayed at the left side of the screen. Two input variable can be displayed at a time. The results of two separate rule files can be compared using this function.

The outputs as produced by the compounded membership functions are displayed on the right side of the screen.

If the mouse pointer is placed on the time axis of one of the displayed input graphs, the input and output values at that time are displayed above all graphs.

The simulation function is useful for gaining a rough approximation of the results produced by a set of rules. Since the process model is known, design errors can be detected immediately by observing the input and output variables.

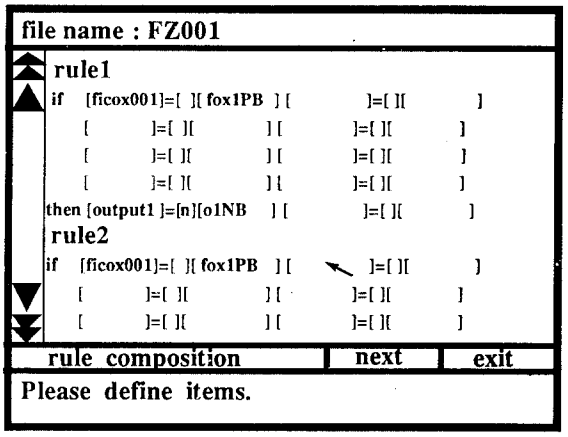

Fig. 3.5 Rule screen

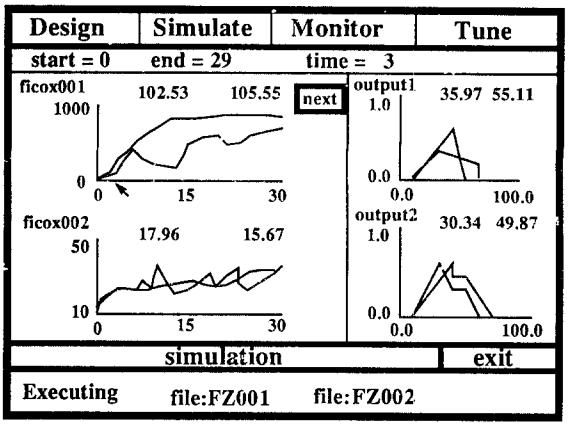

Fig. 3.6 An example of simulation

3.2 Simulation

The simulation function allows you to execute control of a given process model in software. To use this command, you must have specified the appropriate rules, inputs-outputs, scaling factors and simulation period under a file name, as described for the Design command in Section 3.1. Here you need only specify the file name here and the simulation period. The simulation results produced can be stored to a file.

3.3 Monitoring

The monitoring function collects data, such as that from process variables, from the controller. When this command is selected, FZtool monitors the requested data according to a specified sampling period and number of data. Three types of data can be collected. These are data from input variables, data from output variables and data from other variables necessary to analyze and tune the system for plant operation.

3.4 Tuning

The tuning function supports the preparation of tuning parameters for modification of those factors important to control, such as membership functions and scaling factors.

There are four subcommands under the Tuning command. "Response" is used to display response data which are gathered by the monitoring. "Reasoning" displays reasoning results. "Firing" displays information about rules firing and "charact." shows input and output characteristics.

Fig. 3.7 shows an example of a rule firing display where the graph under "freq." shows the firing frequency of every rule. (Firing frequency is a summation of certainty factors at every processing period.) The information displayed under "trend" on this screen shows the rule number for which firing frequency is maximum at every processing period.

The "correct" command bar just below the data display area allows you to call up the appropriate design screen for making corrections. Click "scales," "rules" or "membership" to call up the appropriate screen.

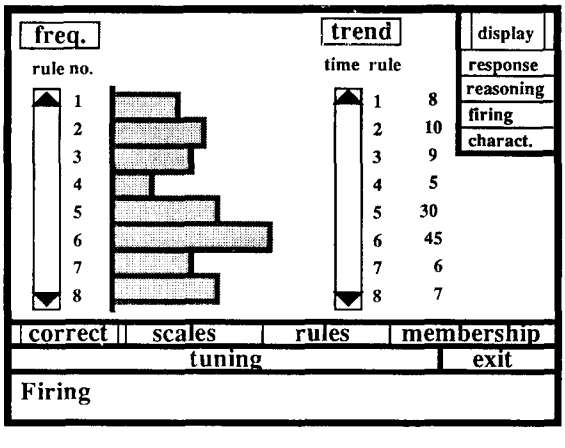

Fig. 3.7 Screen showing rule firing

4. Conclusions

The FZtool software package for configuring fuzzy control systems has been introduced in this paper. FZtool has four primary functions. FZtool supports fuzzy control from design to testing (simulation and monitoring).

The main features of FZtool are as follows:

(1) Comprehensive support functions
- Fuzzy control design
  Rule file allocation, input/output variables definition, membership function creation, rule description, and scaling factor definition.
- Compiling and assembling
  Macrocode and microcode generation
- Parameter tuning
- Testing
  Simulation associated with on-line monitoring.

(2) User-friendly operation

Operation of FZtool is easy with mouse operation, windowed displays, pull-down menus and a help function. Invalid selections and entries are locked out. These features make FZtool easy to use, even for beginners.

(3) Open to users

No programming or fuzzy logic expertise is required to use FZtool. All procedures are executed through a function/module selection method. FZtool running under MS-DOS on a commercial IBM PC/AT compatible laptop computer provides a de facto standard user environment.

In conclusion, FZtool contributes to a savings in engineering time and manpower by realizing just-fit systems for applications at low cost.

References

[1] L.A.Zadeh, "Fuzzy sets," Informat. Control, vol. 8, pp.338-353, 1965.

[2] E. H. Mamdani, "Applications of fuzzy algorithms for simple dynamic plant," Proc. IEE, vol. 121, no. 12, pp.1585-1588, 1974.

[3] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller,"Int. J. Man. Mach. Studies, vol. 7, no. 1, pp.1-13, 1975.

[4] E.H.Mamdani, "Advances in the linguistic synthesis of fuzzy controllers," Int. J. Man. Mach. Studies, vol.8, no. 6, pp.669-678, 1976.

[5] P. J.King and E. H. Mamdani, "The application Automat., vol.13, no.3, pp.235-242, 1977.

[6] E.H.Mamdani, "Application of fuzzy logic to approximate reasoning using linguistic synthesis," IEEE Trans. Computer , vol. C-26, no. 12, pp. 1182-1191, 1977.

[7] L.P.Holmbland and J.J.Ostergaard, "Control of Cement Kiln by Fuzzy Logic," Fuzzy Information and Decision Processes" (M.M.Guputa, E.Sanchez eds), North-Holland, pp.389-399, 1982.

[8] C.C.Lee, "Fuzzy logic in control systems: fuzzy logic controller-part I," IEEE Trans. Systems, Man and Cybernetics, vol. 20, no.2, pp.404-418, 1990.

[9] C.C.Lee, "Fuzzy logic in control systems: fuzzy logic controller, part I," IEEE Trans. Systems, Man and Cybernetics, vol.20, no.2, pp.419-435, 1990.

[10]Y.Tukamoto, "On Applications of Fuzzy Theory to Soft Sciences," J.SOFT, vol.2, no.3, 1990[In Japanese].

[11]K.Hirota, "The Current State of Fuzzy Control," Proc. Inst. Control. Engineering conf., vol.19, no.5, pp.1-13, 1988[In Japanese].

[12]O.Katai and I.Hayashi edt, "A special edition of products and techniques applied fuzzy logic," J.SOFT, vol.3, no.2, 1991[In Japanese].