Fuzzy Control Educational Tutorial and Simulator

M. A. GAMAL, M. M. GOMAA, M. SOBH
Computer and Systems Engineering department
Faculty of Engineering, Ain Shams University
1 El_Sarayatte st., 11517 Abbassia
Cairo - EGYPT.

Abstract:- This paper presents a multipurpose fuzzy control support system. Fuzzy control is an intelligent control aimed to have an ultimate degree of autonomy in terms of reasoning and planning and to have the ability to extract the most valuable information from unstructured and noisy data. Conventional control versus intelligent control is discussed first in this paper. Then the theoretical background related to fuzzy logic and fuzzy control is given. Next, the proposed fuzzy control support system is introduced. This system includes an educational tutorial for fuzzy control, a design tool of a fuzzy controller, a simulation tool giving the total fuzzy controlled system evolution, a wizard that gathers all required information from user, and finally a possibility of fuzzy rule-base generation either from equivalence to a conventional three-term controller or from simulation. This rule-base generation feature helps to obtain an approximate rule-base that may be refined later on, according to the expected performance.

The proposed support system is developed using Visual C++ and run under PC platform. The animated, interactive, and graphical windows interfacing facilitates the usage of this system.

A simple fuzzy control system of two coupled tanks is handled as an illustrative example.

Key words:- Fuzzy Logic, Fuzzy Control, Intelligent Control, Modeling, Simulation.

1. Introduction

Conventional automatic control theory needs mathematical precise models of the process to be controlled. Feedback, feedforward, PID, minimum variance, minimum energy, ... control types are examples of conventional control. The basic problems encountered with this type of control, especially with the increase of processes complexity and the demand of reliability, are: unidentifiable processes, model uncertainty, incomplete data, completely changed processes, unexplainable behaviors, etc. [1]. The need for dealing with uncertainty, incomplete information, and with the process operator mentality, urged the researchers over the last ten years to concentrate on, and employ what is called *intelligent control*. This type of control applies *Artificial Intelligence* (AI) techniques; it is based on process models that are very similar to human mental models. These models are uncertain and depends on heuristics and common sense reasoning. There are three intelligent control approaches that have had significant impacts:

- *Expert control*: uses a symbolic reasoning approach, where the knowledge of experts is made available to the user.
- *Artificial Neural Networks (ANNs)_based control*: ANNs are learning systems capable of uncertain and/or nonlinear mappings.
- *Fuzzy control*: which is well suited to handling heuristic knowledge in controlling a system.

This paper takes into consideration the fuzzy control approach while other intelligent control approaches are out of its scope. This paper presents a fuzzy control multipurpose support system that provides an integrated environment to a process knowledge engineer through which she/he can makes it all. She/he can learn the basic concepts of fuzzy control, design a fuzzy controller, test a fuzzy controller, apply a fuzzy control to a process, and simulate the fuzzy controlled system. A <u>new approach</u> of fuzzy rule-base generation from simulation is also presented in this support system.

In section 2, the basic concepts of fuzzy logic are resumed. The principles of fuzzy control are presented in section 3. In section 4, the main features of the proposed fuzzy multipurpose support system are explained. Section 5 concentrates on the methodology by which a rule-base is generated. The application of two coupled tanks, as a fuzzy control system, is given in section 6. Final conclusions are given in section 7.

2. Fuzzy Logic

The basic concepts of the fuzzy sub-sets theory and the fuzzy modeling are presented in this section; this theory was lanced by L. A. Zadeh in 1965. The details can be found in [2], [3], [4], [5], [6].

2.1 Fuzzy Sub-Sets Theory

The fuzzy is related to the difficulty of realization of slicing distinction among information, which is the frequent case in the human mentality. A fuzzy information can be imprecise and/or uncertain: the imprecision concerns the information contents while the certainty concerns the truth or the falsity of the information. The theory of fuzzy sub-sets treats linguistic variables instead of numerical variables.

If E is a reference set (integers, reals, etc.) and x is a defined variable over this set, then A is said to be a fuzzy sub-set (or fuzzy part) of E, if there is an application μ_A called membership function, defined by: μ_A: E $\to$ [0, 1]; $\mu_A(x) \in [0, 1]$. The reference set E is transformed into a set of fuzzy sub-sets $\mathbb{F}(x)$. A fuzzy sub-set A_i, for I: 1 $\to$ m, is associated to a linguistic signification L_i. Therefor, a transformation τ exists such that: $\mathbb{F}(x) = \{A_1, A_2, ..., A_m\}$ (the set of all fuzzy parts), $\mathbb{L}(x) = \{L_1, L_2, ..., L_m\}$ (set of all linguistic significations), and τ: $\mathbb{F}(x) \to \mathbb{L}(x)$.

The membership function concerning the fuzzy sub-set A_i is written as $\mu_{L_i}(x)$. Consider for example a variable x defined on $\Re^+$, one can define three fuzzy sub-sets as shown in fig. 1 with the fuzzy significations (small), (middle), and (large). The definition of a membership function does not follow any rule and depends only on the human experience and statistic data. The form of a membership function the most used is whether triangular or trapezoidal.

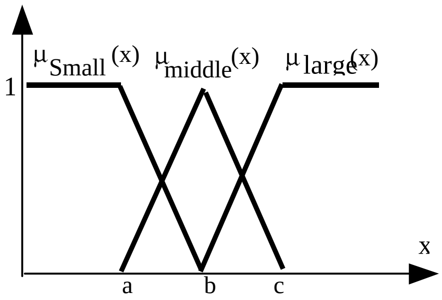


Fig. 1 Linguistic signification of a fuzzysub-set.

2.2 Fuzzy Modeling

A fuzzy proposition is a representation of knowledge expressed in a natural manner by a human. The proposition takes the form "x is Lx" where x is a numeric variable and Lx is a linguistic signification of a fuzzy sub-set; e.g. "z is small", "y is large", "temperature is high", etc. A fuzzy model of human knowledge is a set of fuzzy rules; a fuzzy rule is a conditional scheme of the form: $rule_i$:

IF ($Proposition_{i\_1}$) THEN ($Proposition_{i\_2}$). $Proposition_{i\_1}$ and $Proposition_{i\_2}$ are called antecedent and conclusion of the rule number i respectively. An example is: $rule_i$: IF "x is small" THEN "y is large". The antecedent proposition can be a combination of propositions via logical connectives AND, OR, and/or NOT. An example is: $rule_i$: IF "x is small" AND "y is middle" THEN "z is large".

2.2.1 Fuzzy Inference: Generalized Modus Ponens

Consider the fuzzy rule: IF "x is Lx" THEN "y is Ly"; it is given $\mu_{Lx}(x)$ (the membership function that x is Lx), and the observed fact "x is L'x" with $\mu_{L'x}(x)$ (it is 1 in case of precise input numeric measurment (x_0), and in case of imprecise measurement, it is a triangular membership function with its vertex at 1 and its base depends on the precision). The conclusion (in presence of the observed fact which is imperfectly convenient to the antecedent) is "y is L'y" with membership function $\mu_{L'y}(y)$ which is derived by the application of the combination-projection principle (generalized modus ponens):

$$\mu_{L'y}(y) = SUP_{x \in E} T\ (\mu_{L'x}(x),\ \mu_R(x, y)) \quad ......(1)$$

where E is the reference axis, and T is a triangular norm expressing a fuzzy conjunction. Many T-norms are defined:

$T1(u, v) = \min(u, v)$ (Zadeh)

$T2(u, v) = u.v$ (Probabilistic) …(2)

$T3(u, v) = \max(0, u+v-1)$ (Lukasiewicz)

where u and v are membership functions; $\mu_R(x, y)$ is a possibilistic distribution function describing the causal link, or the implication degree, between the antecedent and the conclusion of the rule. Note that the conversion from input numeric data into suitable linguistic variables and corresponding membership degrees or functions is carried out via a *fuzzification* procedure.

There are two cases of rules representation:

<u>Cas # 1 numeric representation of rules</u>: it is given the membership function $\mu_{Ly}(y)$ of the proposition "y is Ly"; $\mu_R(x, y)$ is calculated (in case of conjunctive reasoning) as follows:

$$\mu_R(x, y) = T(\mu_{Lx}(x), \mu_{Ly}(y))$$

$$\Rightarrow \mu_{L'y}(y)=SUP_{x\in E}T(\mu_{L'x}(x),T(\mu_{Lx}(x), \mu_{Ly}(y)) \quad \ldots\ldots\ldots(3)$$

On using T1, for example,

$$\Rightarrow \mu_{L'y}(y)=SUP_{x\in E}\min(\mu_{L'x}(x), \min(\mu_{Lx}(x), \mu_{Ly}(y)) =SUP_{x\in E}\min(\min(\mu_{L'x}(x),\mu_{Lx}(x)),\mu_{Ly}(y)). \quad \ldots\ldots\ldots(4)$$

This case can consider precise and imprecise (fuzzy) input numeric data within the fuzzification procedure.

<u>Cas # 2 symbolic representation of rules</u>: it is given $\mu_R(x, y) \in [0, 1]$ as a factor expressing the confidence of the rule, and often the rule is written under the form : IF "x is Lx" THEN "y is Ly" (CF = cf). CF is the confidence factor of the rule. This case considers only the precise input numeric data to be fuzzified, $x = x_0$.

$$\Rightarrow \mu_{L'y}(y) = T(\mu_{L'x}(x_0), cf) \quad \ldots\ldots\ldots(5)$$

The T-norm T2 is often used in this case and therefor:

$$\mu_{L'y}(y) = \mu_{L'x}(x_0)*cf. \quad \ldots\ldots\ldots(6)$$

2.2.2 Fuzzy Aggregation

There are two types of aggregation: aggregation of propositions at the level of a rule and aggregation of multiple rules, whose output variable in their conclusions is the same.

<u>aggregation of propositions</u>: the logical operator AND is replaced by a triangular T-norm (T1, T2, T3, etc.). The logical operator OR is replaced by a triangular T-conorm ($\perp$) expressing a fuzzy disjunction. Many T-conorms are defined:

$\perp 1(u, v) = \max(u, v)$ (Zadeh)

$\perp 2(u, v) = u+v-u.v$ (Probabilistic) …………..(7)

$\perp 3(u, v) = \min(1, u+v)$ (Lukasiewicz).

The fuzzy negation is made as:

$$n(u) = 1-u \quad \ldots\ldots\ldots(8)$$

If it is given for example the proposition ("x is Lx" AND ("y is Ly" OR "z is Lz")), then the membership function of this proposition is $T(\mu_{Lx}(x), \perp(\mu_{Ly}(y), \mu_{Lz}(z))$.

<u>aggregation of multiple rules</u>: given m rules of the form: R_i : IF (x is Lx_i) THEN (y is Ly_j) ; $i:1 \rightarrow m$, then:

$$\mu_F(y)=MAX_{(i,j)\in I}SUP_{x\in E}T(\mu_{L'x}(x), T(\mu_{Lxi}(x),\mu_{Lyj}(y))) \quad ..(9)$$

where I is the set of the different combinational couples between Lxi and Lyj; $\mu_F(y)$ is the deduced fuzzy conclusion about y where for any value of y (e.g. y_0), the corresponding degree of confidence of its value is $\mu_F(y_0)$. If the T-norm T1 is used, then:

$$\mu_F(y)=MAX_{(i,j)\in I}SUP_{x\in E}\min(\min(\mu_{L'x}(x), \mu_{Lxi}(x)), \mu_{Lyj}(y))..(10)$$

If precise input numeric data is considered, then the inference is simplified by:

$$\mu_F(y)=MAX_{(i,j)\in I}\min(\mu_{Lxi}(x_0),\mu_{Lyj}(y)) \quad ...(11)$$

note that $\mu_{L'x}(x_0) = 1$, which is known as « *singleton* » fuzzy sub-set.

3. Fuzzy Control

The basic structure of a fuzzy controller is given in fig. 2. Each component is explained in the following:

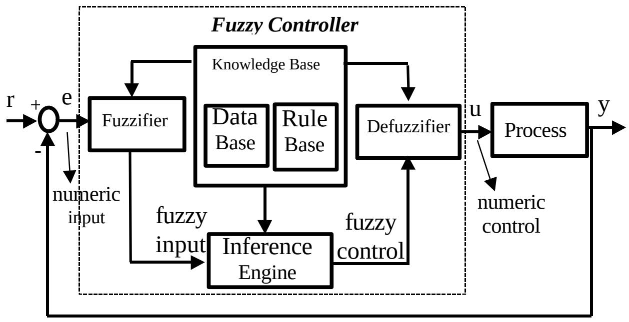


Fig. 2 Structure of a fuzzy controller.

Data base: defines the membership functions of fuzzy sub-sets of each variable over its universe of discourse. Possible fuzzy sub-sets (linguistic variables) of a control variable (error, change of error, control action, etc.) are: PB (Positive Big), PM (Positive Medium), PS (Positive Small), ZE (Zero), NS (Negative Small), NM (Negative Medium), and NB (Negative Big). A membership function may be triangular, trapezoidal, etc.

Rule base: fuzzy control rules are a set of imprecise conditional statements which constitute a set of linguistic decision rules. The form of a fuzzy control rule is: IF (conditions are satisfied) THEN (actions can be taken). An example of a fuzzy control rule is: If ((error in level is PB) AND (change of this error is NS)) THEN (control action is PB). The fuzzy control rule-base can be derived from human operator's experience, control engineer's knowledge, modeling the operator's control actions, and/or from a fuzzy model of the process [7].

Fuzzifier: is an interface that receives input control numerical variables (like error, change of error, etc.) and transforms them into their linguistic significations with appropriate membership degrees, using the data base. Note that a numerical value may correspond to multiple linguistic values with different membership degrees, e.g. error is ZE with 0.7 certitude and PS with 0.5 certitude (sum is not necessarily unity, as possibilistic approach is followed).

Inference engine: specifies at first the applicable fuzzy rules (from the rule-base) convenient to the current fuzzy control inputs, and then determines the fuzzy control action through a fuzzy inference (as explained above). In the literature [8], [9], [10], the one can find Mamdani, Larsen, Tsukomoto, and Takagi fuzzy inference (or reasoning). Each type of fuzzy reasoning differs from the others in the way that membership functions are assigned, the fuzzy rules are written, and the fuzzy operators are chosen. In the case of Mamdani's fuzzy inference, the T-norm T1 (which is the minimum) is used as explained in section 2. In the case of Larsen's fuzzy inference, the T-norm T2 (which is the product) is used. Tsukomoto's fuzzy inference is similar to that of Mamdani but two fuzzy sub-sets only are used for each variable : Positive (P) and Negative (N). In the case of Takagi's reasoning, a fuzzy rule is such that the antecedent is expressed in terms of fuzzy variables (as normal) but the conclusion expresses the control action as a function of numerical values of the input control variables ; the degree of certainty of a fuzzy rule influences the numerical function of that rule. Fig. 3 shows an example of Mamdani's fuzzy inference, assuming that the applicable fuzzy rules are:

Rule_1: If((e is PS)AND(ce is PM))THEN (u is PS)

Rule_2: If((e is PM)AND(ce is PS))THEN (u is PM)

where e is the error, ce is the change of error, u is the control action, e_0 and ce_0 are given precise numerical values of the error and its change. The inference law is (cf. section 2):

$$\mu_F(u)=MAX_{(i,j,k)\in I}\min(\min(\mu_{Lei}(e_0),\ \mu_{Lcej}(ce_0)),\ \mu_{Luk}(u)) \quad \text{......(12)}$$

$\mu_F(u)$ is the fuzzy control action (deduced fuzzy conclusion about u).

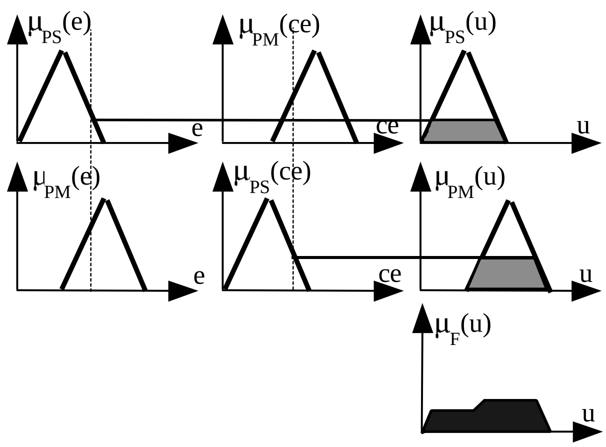


Fig. 3 Mamdani's fuzzy inference.

Deffuzifier: is an interface that performs the inverse operation of a fuzzifier; it receives the fuzzy control action $\mu_F(u)$ and gives a crisp numerical control action u_0. Two reliable methods of defuzzification are shown in fig. 4 [11]. The mean of maximum (MOM) method takes the average value of all control actions whose membership function attains a maximum. The center of gravity (COG) method takes the average of the control action values weighted by the grade of membership.

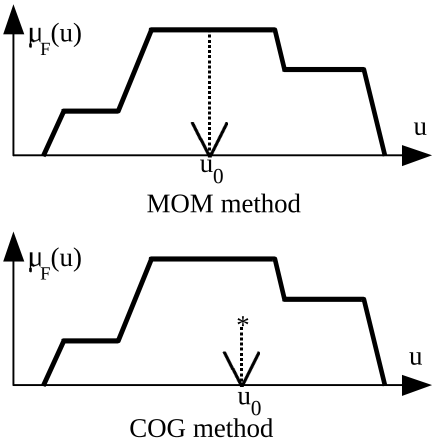


Fig. 4 Defuuzification methods.

4. Fuzzy Control Educational Tutorial and Simulator

In this section, the main functions and structures of the proposed fuzzy control multipurpose support system are presented. The main features of this system are:

- *Visual tool that simplifies design*: the simplicity from interactive visual interface, wizard modules that guide user, and validation that prevent logical errors.
- *On-line help*: in each application dialog, user will find help to understand each dialog item and the general operation of the dialog.
- *Capacity to test and tune designed fuzzy controller*: there is a capacity to test a designed fuzzy controller directly by applying crisp input and find the crisp output, or by connecting the fuzzy controller to control a system defined by state space equations and run the simulator then watch the response through a graph.
- *Wizard that gathers all required information from user*: the idea of the wizard is to teach the user the method or the steps of fuzzy control design. After using the wizard several times, user can use application modules separately.
- *Automatic rule-base generation*: this is done either in equivalence to a three-term controller or starting from simulation (cf. section 5).

The first session with the fuzzy support system provides the screen shown in fig. 5. All system's modules can be accessed directly from the main tool bar. The function of each tool bar icon is illustrated in fig. 6. Each module of the proposed system is explained in the following.

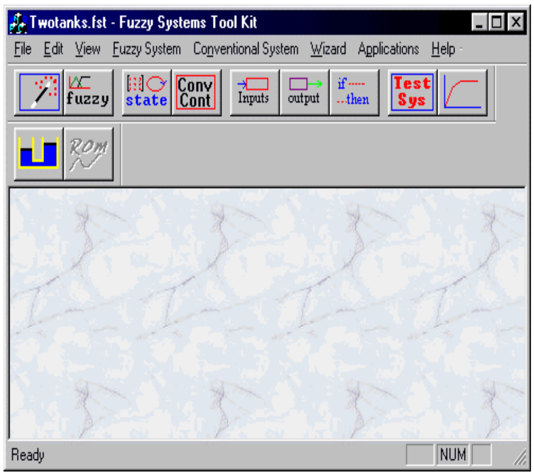
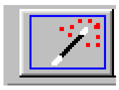
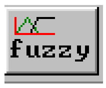


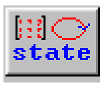
Fig. 5 First session with the fuzzy support system.



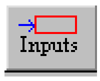
wizard module.



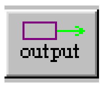
test of implemented fuzzy system.



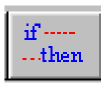
process model.



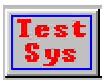
edit input fuzzy variables.



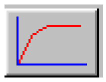
edit output fuzzy variables.



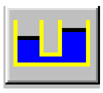
edit fuzzy rules.



apply control.



graph showing the result.



open double tank simulator module.

Fig. 6 Tool bar icon functions.

4.1 Fuzzy Controller Module

In this module, the one can define a fuzzy controller where its fuzzy rule-base, the membership functions of its input and output variables can be edited. The rule-base editor sub-module offers an easy way to edit a fuzzy control rule just by determining the linguistic signification of the input and output variables, using combo-box controls. The rule-base editor offers also an easy navigation through rules sequentially using spin control. Fig. 7 shows the interface of the rule-base editor. The membership function editor sub-module allows the definition of the membership functions of the linguistic significations of each variable whether it is input or output. This editor supports any number of membership functions, supports trapezoidal and triangular membership types, and it gives a graph to view fuzzy variable memberships and reflects any change in fuzzy variable data. Fig. 8 shows the interface of the membership function editor.

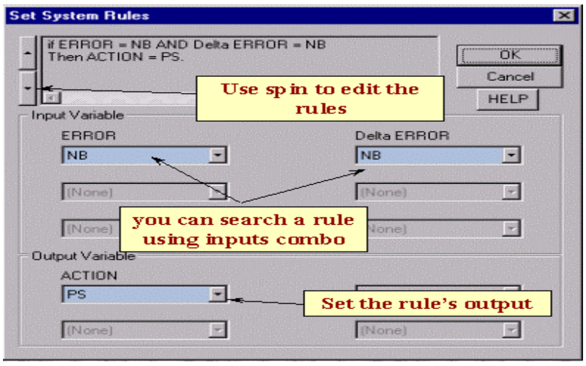


Fig. 7 Rule-base editor.

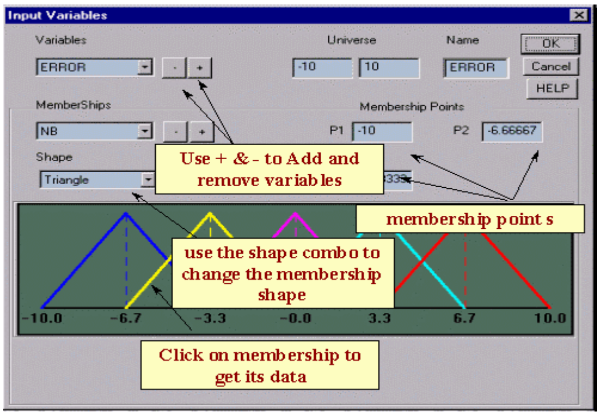


Fig. 8 Membership function editor.

4.2 Test Module

This module allows the test and experimentation of a fuzzy controller after its definition. At first, the user determines the method of defuzzification to be followed. Next, the user is allowed to enter crisp values of the fuzzy controller input variables (error and change of error). This is done by using slider control or by entering the variable crisp value as a number; the system prevents user from entering wrong data (i.e. out of the universe of discourse). The system shows the position of the crisp value on the membership functions graph, as shown in fig. 9. Pressing « Next » button allows moving to the next input variable. After giving the input variables crisp values, the system can show the set of firable rules, and then can give the crisp control action value after defuzzification.

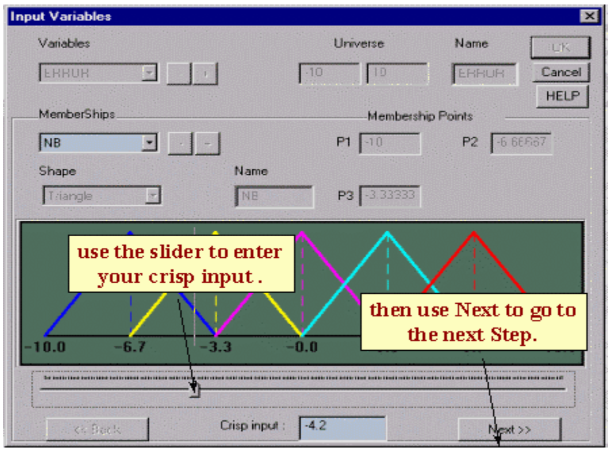


Fig. 9 Entering input variables crisp values.

4.3 Process Module

In this module, a process can be defined. This definition is made by giving the process state space model of the form:

$$X^{\bullet} = AX + BU$$
$$Y = CX + DU \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(13)}$$

where X, U, and Y are the state, input, and output vectors respectively ; A, B, C and D are the state, input, and output matrices respectively. Fig. 10 shows the interface of the process editor module.

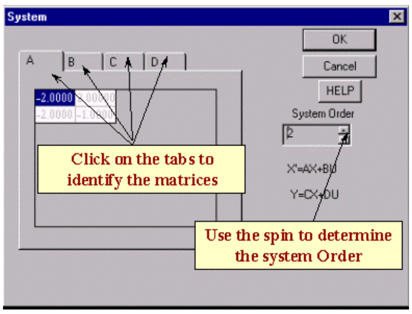


Fig. 10 process model editor.

4.4 Simulation Module

This module connects a fuzzy controller to a process that have been defined (within the corresponding modules) and run a simulator to find out the response of the fuzzy controlled process, as will be shown in section 6.

4.5 Wizard Module

This module enables the user to define complete fuzzy control system in a set of steps:

- define control system (through state space model);
- define fuzzy variables (input - output);
- define fuzzy control rules either manually or automatically. Automatic rule-base generation is explained in the next section.

5. Rule-Base Generation

The control of a process using conventional methods, represents itself an experience; this experience can be converted into fuzzy control rules in an automatic fashion. Conversion to a fuzzy control is in need to deal with uncertain and incomplete information. This conversion may be done either in equivalence to a PID controller using the fuzzy PID method [12], or, as proposed in this paper, from simulation as explained in the following:

5.1 Fuzzy PID Method

This method assumes the membership functions of the error (e), change of error (δe), change of change of error ($\delta^2 e$), and change of control (δu) as shown in fig. 11 (note that the control action in the case of PID controller is a change of control signal). The e_i, δe_j, $\delta^2 e_k$, and δu_l, for i, j, k, and l : ..., -1, 0, 1, ... are called modal values. At these modal values, the outputs of both fuzzy and conventional PID controllers must be equal to each other. It can be shown that the conventional PID control law is:

$$\delta u = T.K_I.e + K_P.\delta e + (K_D/T).\delta^2 e \ \ldots\ldots\ldots..(14)$$

where T is the sampling period; K_P, K_I, and K_D are the proportional, integral, and derivative gains respectively. This control law is valid at modal points of the fuzzy controller, so:

$$\delta u_l = T.K_I.e_i + K_P.\delta e_j + (K_D/T).\delta^2 e_k \ \ldots\ldots.(15)$$

$$\Rightarrow l.\Delta D=T.K_I.i.\Delta A+K_P.j.\Delta B+(K_D/T).k.\Delta C \ \ldots.(16)$$

Therefore, if it is given K_P, K_I, K_D, and T of a conventional PID controller, the fuzzy equivalent controller is specified by:

$$l = i + j + k,$$
$$\Delta D/\Delta B=K_P,$$
$$\Delta D/\Delta A=T.K_I,$$
$$\Delta D/\Delta C=K_D/T \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots....(17)$$

If the value of ΔD is assumed, then ΔA, ΔB, and ΔC can be got and hence the membership functions are specified. The law that l = i + j + k represent the fuzzy control rules. If, for example i=1, j=0, and k=1, then l=2; this is the rule that: IF ((error is PS) AND (change of error is Z) AND (change of change of error is PS)) THEN (change of control is PB) and so on. Equivalent fuzzy controller of P, PI, or PD controller can be got easily from the general case of a three-term controller explained above.

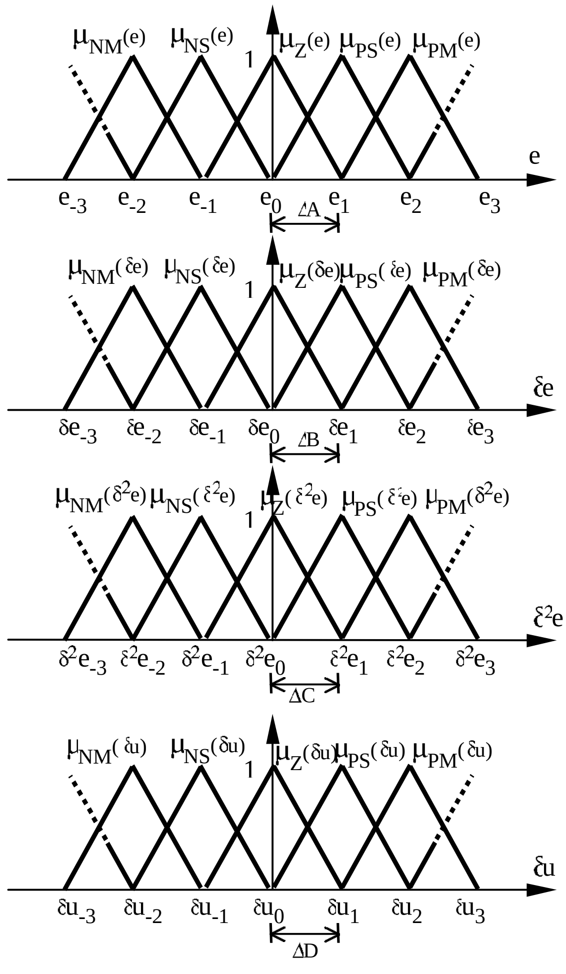


Fig. 11 membership functions of the variables of a fuzzy PID controller.

5.2 Simulation Method

An equivalent fuzzy controller to a conventional one (whether PID or not) can be got from simulation as proposed in this paper as follows:

- input variables of the fuzzy controller are assigned, e.g. error and change of error ;
- membership functions of these input variables and the control action are assigned ;
- simulation of the conventionally controlled process is run for a time t_S ;
- within t_S, find the periods within which the control action is ..., NS, Z, PS, ... (note that there will be overlaps) ;
- for each membership of the controller output, a group of rules can be generated. The generation of the rule-base is illustrated by the following example: suppose that the control action is PS within the periods t1, t2, and t3; it is NS within the periods t4, and t5; it is Z within the period t6 with the overlap and period relativity as shown in fig. 12. Within t1, the error signal verifies the membership functions Z and PS; the change of error verifies NS and Z; other membership functions verified by the error and the change of error for the other periods are indicated in fig. 15. For the membership PS of the output,the generated rules are :

R1_PS: IF ((error is Z) and (change of error is NS)) THEN (control action is PS)
R2_PS: IF ((error is Z) and (change of error is Z)) THEN (control action is PS)
R3_PS: IF ((error is PS) and (change of error is NS)) THEN (control action is PS)
R4_PS: IF ((error is PS) and (change of error is Z)) THEN (control action is PS)
R5_PS: IF ((error is Z) and (change of error is NB)) THEN (control action is PS)
**R6_PS*: IF ((error is PS) and (change of error is NB)) THEN (control action is PS)
***R7_PS*: IF ((error is PB) and (change of error is NB)) THEN (control action is PS)

For the membership NS of the output, the generated rules are:

**R1_NS*: IF ((error is PS) and (change of error is NB)) THEN (control action is NS)
***R2_NS*: IF ((error is PB) and (change of error is NB)) THEN (control action is NS)
R3_NS: IF ((error is PB) and (change of error is PS)) THEN (control action is NS)
R4_NS: IF ((error is PB) and (change of error is PB)) THEN (control action is NS)

For the membership Z of the output, the generated rules are:

R1_Z: IF ((error is Z) and (change of error is PS)) THEN (control action is Z)
R2_Z: IF ((error is Z) and (change of error is PB)) THEN (control action is Z)
R3_Z: IF ((error is PS) and (change of error is PS)) THEN (control action is Z)
R4_Z: IF ((error is PS) and (change of error is PB)) THEN (control action is Z)

Rules of the same antecedents and different conclusions constitute ambiguous situations which are resolved using the relativity of membership periods as follows: consider rules *R6_PS* and *R1_NS*; they are of the same antecedent that error is PS and change of error is NB, this antecedent gives that the control action is PS for period equals to t2+t3 and it is NS for period t4. Since t2+t3 > t4, then rules *R6_PS* and *R1_NS* leads to one rule with the same antecedent and with the conclusion that the control action is PS. Similarly, rules *R7_PS* and *R2_NS* leads to one rule with the same antecedent and with the conclusion that the control action is PS (as t3 > t4). Any ambiguous situation is resolved in the same way.

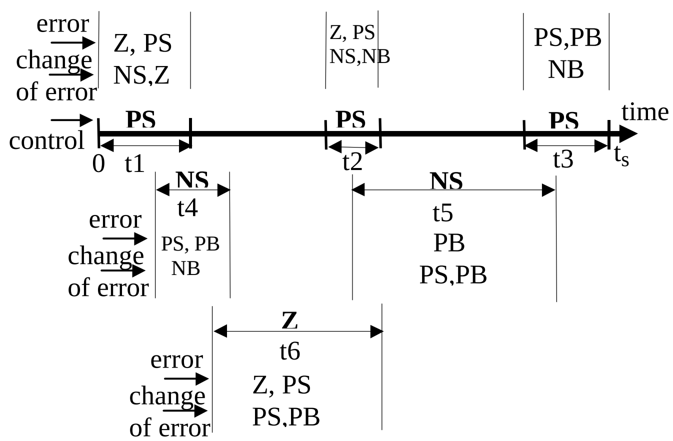


Fig. 12 Rule-base generation via simulation method.

6. Simulation Example

The simulation example considers the fuzzy control of a process of two coupled tanks. The input of the first tank is coming from stream of water that can be manipulated by a valve (input valve opening is the manipulated variable). The tanks are connected together from the bottoms. The head of the first tank affects the flow between the tanks. The controlled variable is the head of the second tank (cf. fig. 13). In this example, the state space model of the process is defined to the fuzzy support system; the system order is 2, the number is inputs is 1 and the number of outputs is 1, with:

$$A = [-2 \quad 2\,; 2 \quad -1],$$
$$B^T=[1 \quad 0],$$
$$C = [0 \quad -3],$$
$$D = 0 \;\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (18)$$

Then, a conventional PI controller is defined with a proportional band P=150 % and an integral time I=1 sec. The step response of the PI controlled process with a step value of 6 level units, is shown in fig. 14. Then an equivalent fuzzy rule-base is generated by the support system via the simulation method; the definition of membership functions of the error and the change of error as input variables, and of the control action, is made as shown in fig. 15. The set of generated fuzzy rules in this example is given in table 1. The step response of the equivalent fuzzy controlled process with a step value of 6 level units, is shown in fig. 16.

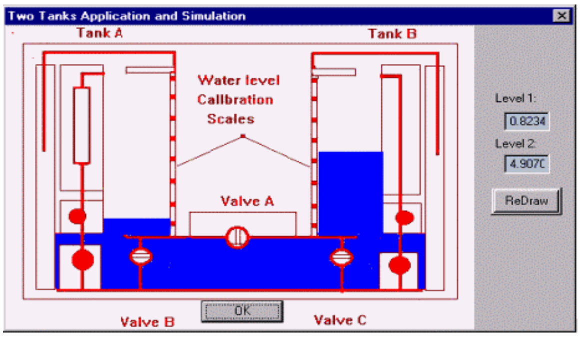


Fig. 13 Two coupled tanks system.

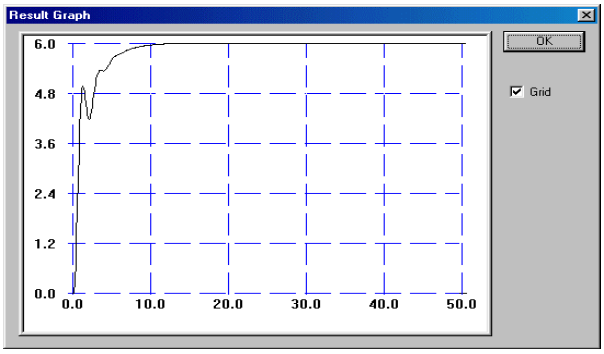


Fig. 14 Step response of the PI controlled system.

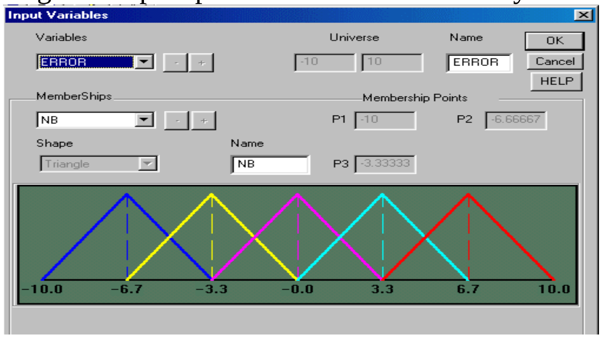


Fig. 15 Membership functions of control input and output variables of the double tank system.

Error / Delta Error	NB	NS	Z	PS	PB
NB	PS	PS	PB	PB	PS
NS	PS	PS	PS	PB	PS
Z	PB	PS	PB	PB	PB
PS	PS	PB	PB	PB	PB
PB	PS	PS	PB	PB	PB

Table 1 Set of generated fuzzy rules via simulation.

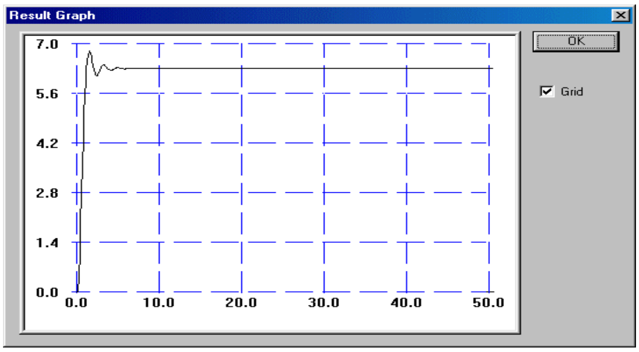


Fig. 16 Step response of the fuzzy controlled system.

The obtained fuzzy controller, in equivalence to a conventional one via the simulation, is not guaranteed to give a better response than the conventional one; it may give a better response or may not, depending on:

- level of noise, perturbation, and incomplete data;
- coverage of the simulation data to all the regions of the membership sets. Hence, during simulation, several set points with different values should be applied to cover all input-output ranges;
- fuzzy membership function shapes and their numbers of the control input-output variables.

In all cases, a generated rule-base can be refined by the process knowledge engineer according to the expected performance; this rule-base guides the process engineer to have a satisfactory final rule-base instead of starting from nothing.

7. Conclusion

No doubt that the recent emerge of the intelligent control techniques has solved many problems encountered during dealing with industrial process in a conventional way. Fuzzy control, as one of the intelligent control techniques, tries to deal with processes as the human operator deals with them as possible. Hence, there is no need for complex precise models but rather simple set of fuzzy rules are in need that code the human experience in a natural way.

This paper has reminded the importance of intelligent control, and then has given a fuzzy control support system including powerful features helping a process engineer to learn and experiment fuzzy control. A new method has been presented in this paper, to generate fuzzy control rules starting from a process model and a conventional controller. This method has been tested on a double tank system, as illustrative example. The response of the fuzzy controlled system approaches to the response of the conventionally controlled system. Further refinement of the generated fuzzy control rules can be done based on the expected fuzzy control system behavior.

References:

[1] M. G. Rodd, H. B. Verbruggen, A. J. Krijgsman," Artificial Intelligence in Real-time Control", *Eng. Applic. Artif. Intell.*, Vol. 5, No. 5, 1992, pp. 385 – 399.

[2] L.A. Zadeh, "Fuzzy Sets as a Basis for a Theory of Possibility ", *International Journal of Fuzzy Sets and Systems*, N° 1, 1978, pp. 3-28.

[3] D. Dubois, H. Prade, "Order of Magnitude Reasoning with Fuzzy Relations " *Revue d’Intelligence Artificielle*, Vol. 3, No. 4, 1989, pp. 69-94.

[4] H.J. Zimmermann, "Fuzzy set Theory and its Applications", *Second Edition, Kluwer Academic Publishers*, 19991, pp. 57-67.

[5] L. Foulloy, "Vers une Méthode d’écriture des Contrôleurs flous", *Rapport Interne LAMII*, N° 91/010, Octobre 1991.

[6] A. M. Ibrahim, "Introduction to Applied Fuzzy Electronics", *Prentice-Hall*, 1997.

[7] E. H. Mamdani, H. J. Efstathiou, K. Sugiyama, "Developments in Fuzzy Logic Control", *Proc. 23rd IEEE CDC, Las Vegas*, 1984.

[8] Y. Tsukamoto, "An Approach to Fuzzy Reasoning Method", *Advances in Fuzzy Sets Theory and Application, North Holland*, 1979.

[9] T. Takagi, M. Sugeno, "Derivation of Fuzzy Control Rules From Human Operator’s Control Actions ", *Proc. of the IFAC Symposium on Fuzzy Information, Knowledge Representation and Decision Analysis, Marsilles, FRANCE*, 1983, pp. 55-60.

[10] C. C. Lee, "Fuzzy Logic in Control Systems : Fuzzy Logic Controller",*Part II, IEEE Trans. Syst.Man.Cybern.*, Vol. 20, No. 2, Mars/April 1990, pp. 419-435.

[11] C. J. Harris, C. G. Moore, M. Brown, "Intelligent Control Aspects of Fuzzy Logic and Neural Nets", *World Scientific Publishing Co.*, 1993.

[12] L. Foulloy, S. Galichet, "Fuzzy Control: Representations, Equivalence, and Comparative Study ", *CNR Project, Report # 92-2, CNRS Automatic Control Research Group, July*, 1992.