PROCEEDINGS
OF THE INSTITUTION OF ELECTRICAL ENGINEERS
Control & Science

Application of fuzzy algorithms for control of simple dynamic plant

E.H. Mamdani, B.E., M.Sc.(Eng.), Ph.D., Mem. I.E.E.E.

Indexing terms: Controllers, Control engineering applications of computers, Direct digital control, Steam plants

Abstract

The paper describes a scheme in which a fuzzy algorithm is used to control plant, in this case, a laboratory-built steam engine. The algorithm is implemented as an interpreter of a set of rules expressed as fuzzy conditional statements. This implementation on a digital computer is used on-line, to control the plant. The merit of such a controller is discussed in the light of the results obtained.

1 Introduction

Techniques from the field of artificial intelligence may be usefully employed to control a complex, nonlinear dynamic plant. Although such plants may be difficult to control manually, it may be possible to control them by means of a suitable heuristic program. The effectiveness of such programs has been demonstrated in chess playing and theorem proving etc. These programs may be very complex and hence difficult to construct, and may also take a long time to evaluate decisions. Thus, they have not often been applied to control a dynamic plant, although, in theory, it should be possible to do so. On the other hand 'learning' controllers have been widely studied, and some of these have indeed been based on fuzzy set theory.[1] These controllers are similar to pattern recognisers in that their structure is postulated first, and this is then evaluated with respect to its function and convergence properties. The purpose of a heuristic program is to implement a 'rule-of-thumb' function of a controller, and, consequently, it may lack structure and generality.

Zadeh's approach, based on fuzzy sets and fuzzy algorithms,[2] provides a general method of expressing linguistic rules so that they may be processed quickly by a computer. At the same time, it is usually possible for an experienced operator to express the strategy or protocol for controlling a plant, using linguistic variables, as a set of rules to be used in the different situations. Thus, a control algorithm may be constructed so that its operation does not depend on the rules being expressed exhaustively, and so that its performance is adequately logged, which may provide clues for subsequent addition to or change of the rules.

2 Control system

The pilot study, using the approach mentioned above, was carried out on simple dynamic plant – a model steam engine. The system can be described briefly with reference to Fig. 1. This highly interactive plant has quantised inputs with 32 possible settings for the heat and 10 settings for the throttle. The preprocessing algorithm compares the analogue-measured outputs of the plant with prespecified set points. It also quantises these error measurements and calculates the change in error from the previous reading. Thus, the controller algorithm computes the action on four quantised variables: errors and changes in errors in the pressure and speed values.

The controller is actually composed of two separate algorithms. One decides the 'change' (or correction) in the heat setting, and the other decides the change in the throttle setting. Each algorithm can base the decision on all four output variables, and thus copes with the interactive nature of the plant. Each algorithm is effectively an interpreter of a set of linguistic rules which have been previously specified. An example of such a rule to control the heat action may be 'If the pressure error is high positive, and if the change in this error is zero or small (positive or negative), and for any values of the speed error and change in speed error, apply a small negative heat change'. Obviously, a shorthand method of specifying these rules is used. They are specified by the human operator, based on his previous experience of the nature of the plant.

The relevant feature of the present study is that the decision algorithms are based on fuzzy-set-theory approach as explained in detail in Appendix 5. The main advantage of this is that, unlike most methods in the field of 'artificial intelligence', the decisions can be calculated within seconds, i.e. in time commensurate with the needs of most dynamic plants. Furthermore, a very small amount of storage space is used, so that mini- or microcomputers can be employed in the implementation.

In the case of the steam engine, the digital computer can make up to ± 7 (including 0) step changes to the heat setting. The algorithm mentioned above itself does not specify one of these 15 possible outcomes, but rather specifies a weight for each of the possible outcomes. This is resolved by the

Paper 7319 C, first received 12th April and in final form 24th September 1974
Dr. Mamdani is with the Department of Electrical & Electronic Engineering, Queen Mary College, Mile End Road, London E1 4NS, England

action-evaluation algorithm which takes the action with the maximum weight, or, where two maximum values occur, an action midway between the two is taken as indicated in Fig. 2.

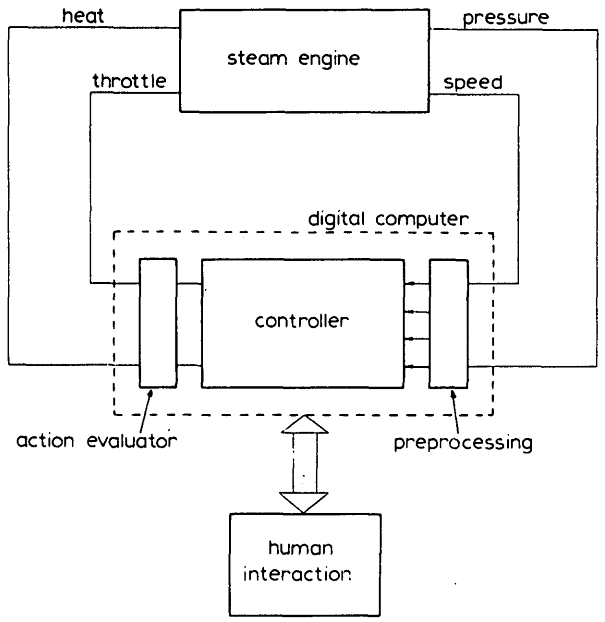

Fig. 1 *Schematic of system*

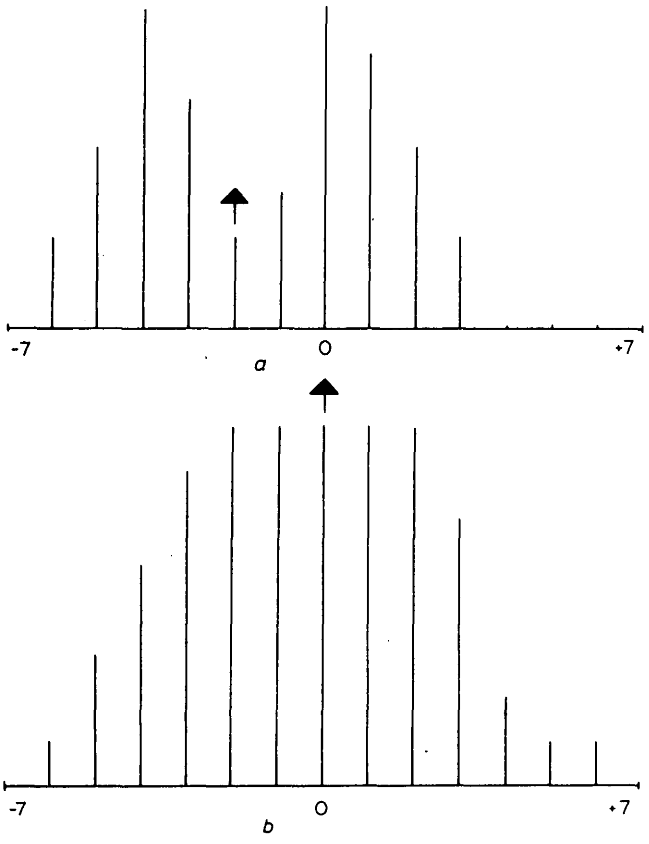

Fig. 2 *Possible fuzzy subsets of action variables resulting from algorithms*

3 Discussion and conclusion

Results have shown that this approach can give similar, if not better, results compared with classical controllers. Its simplicity can therefore make it viable in many practical situations. This is especially true when one considers other special features incorporated in the computer program to aid human interaction.

For example, it is possible to 'tune' the above controller online, by modifying weak or bad rules that give rise to unacceptable actions. To do this, a background routine continuously monitors the operation of the controller and, for each sampling instance, prints out the rules that contribute most to the action taken.

From the results of the present preliminary study, it would appear that this method is chiefly applicable in the control of plants that are difficult to model, such as those in the cement, chemical or iron and steel industries. Current further work is therefore aimed at using this procedure to control different types of system, some with their own special features, such as peculiar nonlinearities or long time delays. It is specifically expected that this further work will lead to generalisations regarding program implementation.

Many of the questions concerning this approach arise mainly from its nonnumerical nature, i.e. stability of the overall control system, completeness of the rules, lack of suitable general-purpose programming language etc. However, numerical and analytical methods have their own limitations, whereas this study shows that heuristic programs may profitably and easily be used by the control engineers. Only more applications can show whether this can always be done.

4 References

1 WEE, W.G., and FU, K.S.: 'A formulation of fuzzy automata and its application as a model of learning systems', *IEEE Trans.*, 1969, **SSC-5**, pp. 215–213

2 ZADEH, L.A.: 'Outline of a new approach to the analysis of complex systems and decision processes', *ibid.*, 1973, **SMC-3**, pp. 28–44

5 Appendix

A fuzzy subset A of a universe of discourse U is characterised by a membership function $\mu = U \rightarrow (0, 1)$, which associates with each element u of U, number $\mu\,(u)$ in the interval (0, 1) which represents the grade of membership of u in A. The fuzzy set A of $U = u_1, u_2, ..., u_n$ will be denoted by

$$A = \sum_{i=1}^{n} \mu_A\,(u_i)/u_i = \sum_i \mu_A\,(u_i)$$

where Σ stands for union.

Three basic operators that are used in this application are defined as follows:

(*a*) The union of fuzzy subsets A and B is denoted $A + B$, and is defined by

$$A + B = \sum_i \mu_A(u_i)\; V \mu_B(u_i)$$

where V stands for maximum (abbreviated to max). The union corresponds to the connective OR.

(*b*) The intersection of A and B is denoted A,B, and is defined by

$$A.B = \sum_i \mu_A\,(u_i)\;\Lambda\,\mu_B\,(u_i)$$

where Λ stands for minimum (abbreviated to min). The intersection corresponds to the connective AND.

(*c*) The complement of a set A is denoted ΓA and is defined by

$$\Gamma A = \sum_i 1 - \mu_A\,(u_i)$$

Complementation corresponds to negation NOT.

The definition of a fuzzy set permits one to assign values to fuzzy variables. In this application, six (four input and two output) fuzzy variables are used:

(i) PE – pressure error, defined as the difference between the present value of the variable and the set point
(ii) SE – speed error, defined as in (i)
(iii) CPE – change in pressure error, defined as the difference between present PE and last (corresponding to last sampling instant)

(iv) CSE – change in speed error, defined as in (iii)
(v) HC – heat change (action variable)
(vi) TC – throttle change (action variable)

These variables are quantised into a number of points corresponding to the elements of a universe of discourse, and values are assigned to the variables, using seven basic fuzzy subsets:

(i) PB – positive big
(ii) PM – positive medium
(iii) PS – positive small
(iv) NO – nil
(v) NS – negative small
(vi) NM – negative medium
(vii) NB – negative big

By using these basic subjects and the three operators defined earlier, values such as 'not positive big or medium' can be assigned to the variables. Even more complex values can be computed using linguistic hedges etc. but, in this study, no such attempt was made, to avoid complications.

The control rules were implemented by using fuzzy conditional statements, for example 'If PE is NB then HC is PB'. The implied relation between the two fuzzy variables PE and HC is expressed in terms of the cartesian product of the two subsets NB and PB. The cartesian product of two sets A and B is denoted $A \times B$ and is defined by

$$A \times B = \sum_i \sum_j min \underset{A}{\mu}(u_i), \quad \underset{B}{\mu}(v_i)$$

$$= \sum_{ij} min\, u_i v_j \qquad \text{(for short)}$$

where u and v are generic elements of the universes of discourse of A and B, respectively. The cartesian product can be conveniently represented by a matrix of m rows and n columns where m and n are the numbers of elements in the universes of A and B, respectively. That is, $i = 1, 2, ..., m$; and $j = 1, 2, ..., n$.

Having thus expressed the relation between two fuzzy variables, it can now be used to infer the value of the second variable given a value for the first. For example, 'PE is NM. What is HC?' Suppose we denote the matrix of relation between two variables by R. Then, if x is the given value of the first variable, the value y of the second variable is inferred by forming the composition $y = x \; 0 \; R$. Composition is interpreted as the max–min product of x and R. Therefore, if $A \times B = R = \sum_{ij} min\, u_i v_j$ as defined above, then a subset A' induces a subset B, given by $B' = \Sigma \max_i min\, u_i'$ $min\, u_i v_j$. Relations of order greater than two can be similarly defined. For example, 'If A then (if B then C)' is given by the cartesian product $A \times B \times C$. And now, given A' and B', the value of C' is inferred to be

$$C' = \sum_k \max_{ij} min\, u_i' u_i v_j' v_j \omega_k$$

In the present application, A' and B' were chosen to be nonfuzzy vectors and with only one element equal to unity, all the rest being zero. In this case, the above expression reduces to

$$C' = \sum_K min\, u_a v_b \omega_k$$

where a and b indicate the elements at which the vectors A' and B' have the value 1. The relationship between these two equations is crucial as it leads to the algorithm actually implemented.

Finally, two or more rules can be combined using the connective ELSE, which is interpreted as the *max* operation, to give an algorithm for the control action. For example,

If A_1 then (If B_1 then C_1)

ELSE If A_2 then (If B_2 then C_2) etc.

yields the resultant control action C', given A_1', A_2', B_1', B_2' etc. as

$$C' = max\, C_1, C_2' \text{ etc.}$$

$$= \Sigma\, max\, min\, u_{a1} v_{b1} z_k,\, min\, u_{a2} v_{b2} z_k \text{ etc.}$$

Therefore, more than one rule may contribute to the computation of the control action. This, of course, is because of the fuzzy nature of the rules. To recapitulate, each algorithm is the application of the above equation in several steps:

At each sampling instant find the quantised levels a, b, c and d for each of the ordered set of the state variables of the plant. Then for each rule i (by referring to defined values) find the quantities u_{ai}, v_{bi}, w_{ei} and x_{di}.
Step 1 Form $min\,(u_{ai}, v_{bi}, w_{ei}, x_{di}) = \alpha_i$
Next, by referring to the defined subset of the action variable in rule 1, form the subset:
Step 2 $C_i' = min\,(\alpha_i, z_k)$ (z_k are the values in the defined subset)
Step 3 Form $C' = max\,(C', C_i')$
Repeat step 1 for the next value of i. The procedure is started with the initial value of $C' = 0$. The final value of C' is the fuzzy subset for that action variable (see Fig. 2).

The PE (pressure error) and SE (speed error) variables are quantised into 13 points, ranging from maximum negative error, and is further divided into negative zero error (NO – just below the set point) and positive zero error (PO – just above the set point). The CPE (change in pressure error) and CSE (change in speed error) variables are similarly quantised without the further division of the zero state.

Apart from the seven primary fuzzy subsets, defined from the four variables mentioned above, a further value ANY is allowed for all these variables. ANY has a membership function of 1 at every element. The membership function in other fuzzy subsets would vary between 1 and 0 at each quantisation level, depending on the definition. For example, for a given variable, say PE, PB – positive big is defined as

0 0 0 0 0 0 0 0 0 0 0 0·1 0·4 0·8 1

and PM – positive medium is defined as

0 0 0 0 0 0 0 0 0 0·2 0·7 1 0·7 0·2 0

From this, a secondary subset, such as NOT PB or PM, will be

1 1 1 1 1 1 1 1 1 0·8 0·3 0 0·3 0·3 0

The two action variables HC (heat change) and TC (throttle change) are quantised in a similar way. There is a control-action algorithm for each of these action variables. The algorithms consist of a set of fuzzy conditional statements. For example, in the case of the heater algorithm two such statements are:

If PE = NB

and CPE = NOT (NB or NM)

and SE = ANY

and CSE = ANY

then HC = PB

ELSE

If PE = NB

and CPE = NS

and SE = ANY

and CSE = ANY

then HC = PM

ELSE

etc.

Note that both these statements assume that the action on heat is a function only of the pressure variables. Thus it is possible to implement a noninteractive control.