Int. J. Human-Computer Studies (1999) **51**, 135-147
Article No. ijhc.1973.0303
Available online at http://www.idealibrary.com on IDEAL®

An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller

E. H. MAMDANI AND S. ASSILIAN
Queen Mary College, London University, U.K.

(*Received 2 November 1973*)

This paper describes an experiment on the "linguistic" synthesis of a controller for a model industrial plant (a steam engine). Fuzzy logic is used to convert heuristic control rules stated by a human operator into an automatic control strategy. The experiment was initiated to investigate the possibility of human interaction with a learning controller. However, the control strategy set up linguistically proved to be far better than expected in its own right, and the basic experiment of linguistic control synthesis in a non-learning controller is reported here.

Introduction

Many techniques for the synthesis of automatic controllers will be found in the control literature. The standard textbook approaches, however, all involve quantitative, numeric calculations based on mathematical models of the plant and controller. In recent years there have been many studies of self-organizing, or adaptive, controllers in which the control strategy is not synthesized in advance but is generated by optimization algorithms based on the controller's "experience". Such controllers exhibit some characteristics of human learning and in their more general forms are examples of "artificial intelligence".

However, the capability of self-organization and learning are only two aspects of the many attributes of human intelligence. A different facet, that has not been emulated in control studies, is the ability to comprehend *instructions* and to generate strategies based not on experience but on *a priori* verbal communication. Luria (1961) has given graphic illustrations of the way in which simple perceptual-motor skills are built on verbal foundations in young children. Pask (1971) has emphasized the linguistic nature of many aspects of intelligent behaviour. In artificial intelligence research Winograd (1973) has given an impressive demonstration of the possibility of direct linguistic control of a mechanical arm performing a building task with solid objects. Most control engineers would accept intuitively that the mathematical computations they perform in translating their concept of a

1071-5819/99/080135 +13 $30.00/0

control strategy into an automatic controller are far removed from their own approach to the manual performance of the same task, and that there seems to be a fairly direct relationship between the loose linguistic expression of a control strategy and its manual implementation. It was this direct path between a linguistic statement of a control strategy and its implementation that formed the subject of this investigation.

The full richness of linguistic structure exhibited so effectively by Winograd (1972) has no equivalent in the present study since we were primarily concerned with the translation of semantic expressions into control laws, and not with the recognition of the expressions themselves or their manipulation. To the control engineer quantitative languages supporting arithmetic are the natural ones. To support the translation of the vaguer, non-numeric statements that might be made about a control strategy we needed a semi-quantitative calculus. Zadeh's (1973) fuzzy logic seemed to provide a means of expressing linguistic rules in such a form that they might be combined into a coherent control strategy. In the case study reported here we have implemented a controller, a *fuzzy logic controller* based on Zadeh's calculus, and investigated its behaviour in the control of a small steam engine.

The Plant to be Controlled

The plant for which the controller was implemented comprises a steam engine and boiler combination. The model of the plant used has two inputs: heat input to the boiler and throttle opening at the input of the engine cylinder, and two outputs: the steam pressure in the boiler and the speed of the engine. Simple identification tests on the plant proved that it is highly nonlinear with both magnitude and polarity of the input variables. Therefore, the plant possesses different characteristics at different operating points, so that the direct digital controller implemented for comparison purposes had to be retuned (by trial and error) to give the best performance each time the operating point was altered.

The Controller

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

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

where Σ stands for union.

Three basic operators that are used in this application are defined next. The union of fuzzy subsets A and B is denoted $A+B$ and is defined by

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

where v stands for maximum (abbreviated to max). The union corresponds to the connective OR. Similarly, the intersection of A and B is denoted $A \cdot B$ and is defined by

$$A \cdot B = \sum_i \mu_A(u_i) \wedge \mu_B(u_i)$$

where $\wedge$ stands for minimum (abbreviated to min). The intersection corresponds to the connective AND. Finally, the complement of a set A is denoted $\neg A$ and is defined by

$$\neg A = \sum_i 1 - \mu_A(u_i)$$

Complementation corresponds to negation, i.e. 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:

(1) *PE*—Pressure Error, defined as the difference between the present value of the variable and the set point.

(2) *SE*—Speed Error, defined as in (1).

(3) *CPE*—Change in pressure error, defined as the difference between present *PE* and last (corresponding to last sampling instant).

(4) *CSE*—Change in speed error, defined as in (3).

(5) *HC*—Heat Change (action variable).

(6) *TC*—Throttle Change (action variable).

These variables are quantized into a number of points corresponding to the elements of a universe of discourse, and values to the variables are assigned using seven basic fuzzy subsets (see appendix): (1) *PB*—Positive Big; (2) *PM*—Positive Medium; (3) *PS*—Positive Small; (4) *NO*—Nil; (5) *NS*—Negative Small; (6) *NM*—Negative Medium; (7) *NB*—Negative Big. Using these basic subsets 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

$$A \times B = \sum_i \sum_j \min\{\mu_A (u_i), \mu_B (v_j)\}$$

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

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, $i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n$.

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 by R the relation between two variables. 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

$$B' = \sum_j \max_i \min u_i' \min u_i v_j$$

Relations of higher order 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

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

In the present application A' and B' were chosen to be non-fuzzy vectors and with only one element equal to 1, all the rest being 0. In this case the above expression reduces to

$$C' = \sum_k \min u_a v_b w_k$$

where a and b indicate the elements at which the vectors A' and B' have the value 1.

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

$$C_1 = \max C_1' \; C_2', \text{ etc.}$$
$$= \sum_k \max \{\min u_{a1} \; v_{b1} \; z_k, \min u_{a2} \; v_{b2} \; z_k, \text{etc.}\}.$$

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, two algorithms were implemented in this application: one to compute the "heat change" (*HC*) control action and the other to compute the "throttle change" (*TC*) control action. Every rule in these algorithms is a relationship between the input variables *PE*, *CPE*, *SE*, *CSE* (in that order) and either *HC* or *TC*. The control actions are computed by presenting values for the input variables to the two algorithms. The input vectors are of course obtained by sampling the states of the steam engine at the sampling instants.

The output of either algorithm is obviously a fuzzy set which assigns grades (of membership) to the possible values of the control fuzzy variable. In order to take a deterministic action one of these values must be chosen, the choice procedure depending on the grades of membership. Various considerations may influence the choice procedure depending on the particular application and in our case effectively that action is taken which has the largest membership grade. It is possible of course that more than one peak or a flat peak is obtained as illustrated below:

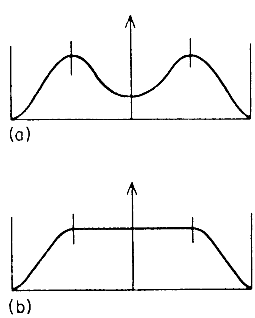

FIG. 1.

The particular procedure in our case takes the action indicated by the arrow, which is midway between the two peaks or at the centre of the plateau.

A spread as in (b) indicates the absence of a strong or good set of rules and is not as serious as the spread in (a). The latter indicates the presence of at least two contradictory rules and in such cases it is necessary to locate and modify these rules. This "tuning" process is facilitated in our application by a data-logging procedure which traces the contributing rules for every control action. Apart from resolving contradictions of this type, the above procedure also helps in modifying any weak or bad rules resulting in unacceptable control actions.

It might be useful to note here that it is the rules that are modified and not the seven basic, but subjective, definitions of the values assigned to the variables involved in the rules. Given a simple, say, 14-point universe of discourse, any subjective definition of "positive big" is hardly likely to be too contentious. Stated otherwise, a set of rules is rather insensitive to the definition of an individual fuzzy subset.

Results and Conclusions

The above scheme containing the 24 rules given in the appendix was implemented on the PDP-8 computer and applied to the steam-engine plant. A fixed digital controller was also implemented on the computer and applied to the same plant for the purpose of comparison. With the fixed controller many runs were required to tune the controller for the best performance. This tuning was done by trial and error. Results of many runs with different set points were taken. The quality of control with the fuzzy controller was found to be better each time than the best control obtained by the fixed controller. This is summarized in the figure below.

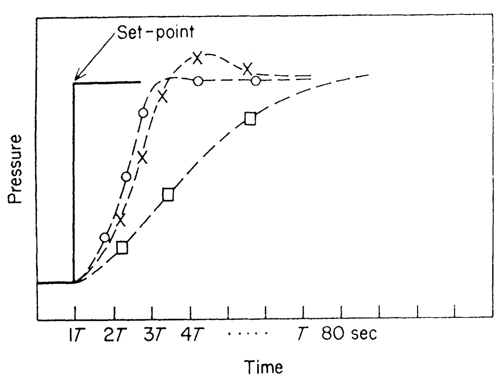

FIG. 2. Fixed controller (DDC algorithm), ×, □; Fuzzy controller, ⊙.

It is difficult to draw many conclusions from a single case study of this kind, apart from the obvious one that it demonstrates the excellent applicability of fuzzy set theory to the design of controllers. Questions may arise as to whether such a deterministic controller represents a non-trivial alternative to other approaches. It is difficult to see any direct relationship between such a controller and any other deterministic controller such as a DDC algorithm or some logic circuit with the same input–output capability. The power of the approach derives from the fact that it is possible to translate into an algorithm an entirely unstructured set of heuristics expressed linguistically.

The only comparable experiment in the literature appears to be that of Gaines (1973, p. 303) who reports a procedure for "priming" an adaptive-threshold-logic-based controller by giving it instructions which it interprets by "mentally rewarding" itself when "imagining" itself carrying out the instructions. Gaines is concerned to demonstrate that learning behaviour shown by human subjects in a series of experiments is also exhibited by a single learning system undergoing the same experiments, and his procedure was put forward as a possible emulation of the effects of instructions on his human subjects. In an earlier paper Gaines & Andreae (1966) propose the use of general-purpose learning controllers as an alternative to classical controller synthesis, and suggest that techniques of *priming* (linguistic communication of an initial control strategy), *coding* (organization of information and control actions to match plant and controller) and *training* (variation of the environment through a sequence of tasks of increasing difficulty) as synthesis techniques appropriate to learning controllers. In terms of this classification we have demonstrated the power of fuzzy logic as a basis for priming and shown that priming alone may be a powerful synthesis technique.

Another way of saying the same thing is that the fuzzy controller has been derived from a fuzzy internal model of the plant identified by the human being. This may suggest a further investigation into a fuzzy identification and control procedure. It would also be interesting to use the approach with other dynamic systems possibly more complex ones (with more input and output variables) where it could be considerably more difficult to specify the rules.

Appendix

The *PE* (Pressure Error) and *SE* (Speed Error) variables are quantized into 13 points, ranging from maximum negative error through zero error to maximum positive error. The zero error is further divided into negative zero error (*NO*—just below the set point) and positive zero error (*PO*—just above the set point). The subjective fuzzy sets defining these values are:

	−6	−5	−4	−3	−2	−1	−0	+0	+1	+2	+3	+4	+5	+6
PB	0	0	0	0	0	0	0	0	0	0	0·1	0·4	0·8	1·0
PM	0	0	0	0	0	0	0	0	0	0·2	0·7	1·0	0·7	0·2
PS	0	0	0	0	0	0	0	0·3	0·8	1·0	0·5	0·1	0	0
PO	0	0	0	0	0	0	0	1·0	0·6	0·1	0	0	0	0
NO	0	0	0	0	0·1	0·6	1·0	0	0	0	0	0	0	0
NS	0	0	0·1	0·5	1·0	0·8	0·3	0	0	0	0	0	0	0
NM	0·2	0·7	1·0	0·7	0·2	0	0	0	0	0	0	0	0	0
NB	1·0	0·8	0·4	0·1	0	0	0	0	0	0	0	0	0	0

The *CPE* (Change in Pressure Error) and *CSE* (Change in Speed Error) variables are similarly quantized without the further division of the zero state. The subjective definitions are:

	−6	−5	−4	−3	−2	−1	0	+1	+2	+3	+4	+5	+6
PB	0	0	0	0	0	0	0	0	0	0·1	0·4	0·8	1·0
PM	0	0	0	0	0	0	0	0	0·2	0·7	1·0	0·7	0·2
PS	0	0	0	0	0	0	0	0·9	1·0	0·7	0·2	0	0
NO	0	0	0	0	0	0·5	1·0	0·5	0	0	0	0	0
NS	0	0	0·2	0·7	1·0	0·9	0	0	0	0	0	0	0
NM	0·2	0·7	1·0	0·7	0·2	0	0	0	0	0	0	0	0
NB	1·0	0·8	0·4	0·1	0	0	0	0	0	0	0	0	0

Apart from the above definitions a further value *ANY* is allowed for all four variables, i.e. *PE*, *SE*, *CPE*, *CSE*. *ANY* has a membership function of 1·0 at every element.

The *HC* (Heat Change) variable is quantized into 15 points ranging from a change of −7 steps through 0 to −+7 steps. The subjective definitions are:

	−7	−6	−5	−4	−3	−2	−1	0	+1	+2	+3	+4	+5	+6	+7
PB	0	0	0	0	0	0	0	0	0	0	0	0·1	0·4	0·8	1·0
PM	0	0	0	0	0	0	0	0	0	0·2	0·7	1·0	0·7	0·2	0
PS	0	0	0	0	0	0	0	0·4	1·0	0·8	0·4	0·1	0	0	0
NO	0	0	0	0	0	0	0·2	1·0	0·2	0	0	0	0	0	0
NS	0	0	0	0·1	0·4	0·8	1·0	0·4	0	0	0	0	0	0	0
NM	0	0·2	0·7	1·0	0·7	0·2	0	0	0	0	0	0	0	0	0
NB	1·0	0·8	0·4	0·1	0	0	0	0	0	0	0	0	0	0	0

Similarly the *TC* (Throttle Change) variable is quantized into 5 points

	−2	−1	0	+1	+2
PB	0	0	0	0·5	1·0
PS	0	0	0·5	1·0	0·5
NO	0	0·5	1·0	0·5	0
NS	0·5	1·0	0·5	0	0
NB	1·0	0·5	0	0	0

The two control action algorithms are given below. These algorithms are based on the non-interactive control principle, i.e. the pressure-heat and speed-throttle loops are separated. Interactive algorithms have also been implemented but these are not discussed in this report.

HEATER ALGORITHM

If $PE = NB$
and $CPE =$ not (NB or NM)
and $SE = ANY$
and $CSE = ANY$
then $HC = PB$

Else
If $PE = NB$ or NM
and $CPE = NS$
and $SE = ANY$
and $CSE = ANY$
then $HC = PM$

Else
If $PE = NS$
and $CPE = PS$ or NO
and $SE = ANY$
and $CSE = ANY$
then $HC = PM$

Else
If $PE = NO$
and $CPE = PB$ or PM
and $SE = ANY$
and $CSE = ANY$
then $HC = PM$

Else
If $PE = NO$
and $CPE = NB$ or NM
and $SE = ANY$
and $CSE = ANY$
then $HC = NM$

Else
If $PE = PO$ or NO
and $CPE = NO$
and $SE = ANY$
and $CSE = ANY$
then $HC = NO$

Else
If $PE = PO$
and $CPE = NB$ or NM
and $SE = ANY$
and $CSE = ANY$
then $HC = PM$

Else
If $PE = PO$
and $CPE = PB$ or PM
and $SE = ANY$
and $CSE = ANY$
then $HC = NM$

Else
If $PE = PS$
and $CPE = PS$ or NO
and $SE = ANY$
and $CSE = ANY$
then $HC = NM$

Else
If $PE = PB$ or PM
and $CPE = NS$
and $SE = ANY$
and $CSE = ANY$
then $HC = NM$

Else
If $PE = PB$
and $CPE =$ not (NB or NM)
and $SE = ANY$
and $CSE = ANY$
then $HC = NB$

Else
If $PE = NO$
and $CPE = PS$
and $SE = ANY$
and $CSE = ANY$
then $HC = PS$

Else
If $PE = NO$
and $CPE = NS$
and $SE = ANY$
and $CSE = ANY$
then $HC = NS$

Else
If $PE = PO$
and $CPE = NS$
and $SE = ANY$
and $CSE = ANY$
then $HC = PS$

Else
If $PE = PO$
and $CPE = PS$
and $SE = ANY$
and $CSE = ANY$
then $HC = NS$

THROTTLE ALGORITHM
If $PE = ANY$
and $CPE = ANY$
and $SE = NB$
and $CSE =$ not (NB or NM)
then $TC = PB$

Else
If $PE = ANY$
and $CPE = ANY$
and $SE = NM$
and $CSE = PB$ or PM or PS
then $TC = PS$

Else
If $PE = ANY$
and $CPE = ANY$
and $SE = NS$
and $CSE = PB$ or PM
then $TC = PS$

Else
If $PE = ANY$
and $CPE = ANY$
and $SE = NO$
and $CSE = PB$
then $TC = PS$

Else
If $PE = ANY$
and $CPE = ANY$
and $SE = PO$ or NO
and $CSE = PS$ or NS or NO
then $TC = NO$

Else
If $PE = ANY$
and $CPE = ANY$
and $SE = PO$
and $CSE = PB$
then $TC = NS$

Else
If $PE = ANY$
and $CPE = ANY$
and $SE = PS$
and $CSE = PB$ or PM
then $TC = NS$

Else
If $PE = ANY$
and $CPE = ANY$
and $SE = PM$
and $CSE = PB$ or PM or PS
then $TC = NS$

Else
If $PE = ANY$
and $CPE = ANY$
and $SE = PB$
and $CSE =$ not (NB or NM)
then $TC = NB$

References

GAINES, B. R. & ANDREAE, J. H. (1966). A learning machine in the context of the general control problem. *Proc. 3rd Int. IFAC Congress, London.*

GAINES, B. R. (1973). The learning of perceptual-motor skills by men and machines and its relationship to training. *Instructional Science*, 263.

LURIA, A. (1961). *The Role of Speech in the Regulation of Normal and Abnormal Behaviour.* Oxford: Pergamon Press.

PASK, G. (1971). A cybernetic experimental method and its underlying philosophy. *Internatl J. Man–Machine Studies*, **3**, 279.

WINOGRAD, T. (1972). *Understanding Natural Language.* Edinburgh: Edinburgh University Press.

ZADEH, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans.* **SMC-3,** 28.