Minimizing rules of fuzzy logic system by using a systematic approach

Conference Paper · February 1993

DOI: 10.1109/FUZZY.1993.327466 · Source: IEEE Xplore

Benito Fernández Rodríguez
The Whisper Company & Machine Essence Corporation

Minimizing Rules of Fuzzy Logic System by Using a Systematic Approach

Chuan-Chang Hung*and Benito Fernández R.†
NeuroEngineering Research & Development Laboratory
Mechanical Engineering Department
The University of Texas at Austin

Abstract— **Fuzzy logic systems have been extensively applied for control and decision systems. The control strategy may be viewed as a rule based design. All fuzzy rules contribute to some degree to the final inference or decision, however, some rules fired weakly do not contribute significantly to the final decision and may be "eliminated" (reduced). It is desired to minimize the rules in order to reduce the computation time to make a faster decision. Karnaugh maps have provided systematic methods for simplifying switching functions in logic design of binary digital systems. Based on this idea, this paper will present a novel method to help us reduce fuzzy rules. Comparisons will be made between systems utilizing reduced rules and original rules to verify the outputs. As a practical example of a nonlinear system, an inverted-pendulum will be controlled by minimum rules to illustrate the performance and applicability of this proposed method.**

I. Introduction

During the past years, rule-based controllers combined with fuzzy logic and approximate reasoning have emerged as one of the most promising application of fuzzy set theory [1]. The rule-based system has been used to emulate and even surpass the decision-making of human reasoning in control processes. However, the design of a rule set of a rule-based system is initially developed to form an approximate or incomplete knowledge base for control strategy. Therefore, it is possible to design more rules than that required by the actual process, but still cannot simulate the entire control process. As a result, it will decrease the effectiveness of control processes due to the time required for firing all the rules and incomplete knowledge of the rules. It is desired to use the most effective rule set instead of implementing all possible rules, and provide a compensation for incomplete knowledge. Then, the computation time will be reduced and a control decision will be made more precisely.

Karnaugh maps have provided systematic methods for simplifying switching functions in logic design of binary digital systems [3]. Given a word description of the desired behavior of a logic network, we can write the output of the network as function of the input variables. This function can be specified as an algebraic expression. Then, we can facilitate algebraic manipulation by using the Minimum maps with boolean algebra. By doing this, we can obtain the minimum sum-of-products or minimum product-of-sums form of a function. Although fuzzy rule-based systems have been used to solve problems that involve ill-defined entities between true and false, we intend to investigate the similarity of fuzzy logic systems and binary digital systems and present a novel method to help us reduce the fuzzy rules based on Minimum maps. The minimum rule base being reduced is expected to have a similar performance as that of the original rule base, even in the case of incomplete knowledge. Therefore, the time required for computing the control signal will be deceased. We will begin the presentation of this systematic approach for reducing the fuzzy rules. Next, the comparison is made between systems utilizing the reduced rules and the original rule-base to verify the outputs. Finally, an inverted-pendulum is controlled by the minimum set of rules to illustrate the performance and applicability of the proposed method.

II. A Systematic Approach

In order to derive the algebraic simplification of switching functions, we need to translate word descriptors of the rules into like algebraic equations. Fuzzy logic allows premises and conclusions of the rules to be fuzzy propositions. The truth values of propositions are expressed as linguistic variables with varying degrees of truth. Therefore, the linguistic value of fuzzy sets are converted into a binary representation in order to apply Minimum maps. After the above transformation, given a set of fuzzy rules (completely or incompletely specified), we can display

*Graduate Student
†Assistant Professor

0-7803-0614-7/93/$03.00 ©1993 IEEE

them on a *minimum fuzzy map*. The fuzzy rules may be represented as a minterm expression (standard sum of products), a maxterm expression (standard product of sums), or as an algebraic form. As a result, the minimum sum-of-products or minimum product-of-sums form rules that can be derived from the map.

In the following section, we will elaborate the procedure of reducing the fuzzy rules by our systematic approach based on minimum fuzzy maps.

A. The Procedure of Reducing Rules

1. In order to apply Karnaugh maps to represent fuzzy rules, the linguistic values (fuzzy sets) of input and output variables in [0,1] must be regarded as crisp values (0 or 1) without considering the overlap between each fuzzy set.

2. Discretize the input and output variables as n-bit integers, having 2^n possible values. Hence, each linguistic values can be represented as one binary value.

3. While displaying the rules on the map, we can minimize the n-variable functions using Karnaugh maps based on different values of only one variable ($Y(True) + Y'(False) = 1$). For instance, the minimum sum-of-products expressions can be derived by using the theorem

$$XY' + XY = X \tag{1}$$

However, each variable in a fuzzy rule has multiple linguistic values in terms of the multiple fuzzy sets. In addition, the simplified form is indicated by looping the corresponding output value on the map, not just the corresponding $1's$ on a conventional map that has only a true or false value. Therefore, the effects of the variables are deleted according to the changes of all linguistic values in the loop. Here, the eq.(1) is modified as follows for simplifying the fuzzy rules:

$$X_1Y_1 + X_1Y_2 + ... + X_1Y_n = X_1 \tag{2}$$

where, X_1 is the one of linguistic values of X variable. $Y_1, Y_2, ..., Y_n$ are n linguistic values of Y variable

Meanwhile, outputs with the same binary value can be reduced into sets formed by the minimum sum-of-products or minimum product-of-sum. In what follows, there are two cases that show this simplification:

(a) Completely-specified rules:

			A			
			A_1	A_2	A_3	A_4
			00	01	10	11
	B_1	00		10		
B	B_2	01	01	01	01	01
	B_3	10		00		
	B_4	11		1 0		

$$\begin{aligned} C &= A_2B_1 + A_1B_2 + A_2B_2 + A_3B_2 \\ &\quad + A_4B_2 + A_2B_3 + A_2B_4 \\ &= A_2B_1 + B_2 + A_2B_3 + A_2B_4. \end{aligned} \tag{3}$$

where, A and B are input variables, C is the output variable. A_1, A_2, A_3, and A_4 are linguistic values of variable A, they are represented as binary numbers such as $A_1 = 00, A_2 = 01, A_3 = 10$, and $A_4 = 11$. B_1, B_2, B_3, and B_4 are linguistic values of variable B, they are represented as binary numbers such as $B_1 = 00, B_2 = 01, B_3 = 10$, and $B_4 = 11$. In the entity of the map, $00, 01, 10$, and 11 are output linguistic values for C_1, C_2, C_3, and C_4 respectively.

(b) Incompletely-specified rules:

			A			
			A_1	A_2	A_3	A_4
			00	01	10	11
	B_1	00		10		
B	B_2	01	01	$\times$	01	01
	B_3	10		10		
	B_4	11		10		

$$\begin{aligned} C &= A_1B_2 + A_3B_2 + A_4B_2 + A_2B_2(\times) \\ &\quad + A_2B_1 + A_2B_3 + A_2B_4 \\ &= B_2 + A_2. \end{aligned} \tag{4}$$

where, the required minterm are indicated by a binary number on the map,and the don't care minterms are indicated by $\times$'s. When choosing terms to form the minimum sum of products, all the same binary number must be covered, but the $\times$'s are only used if they will simplify the resulting expression.

4. Define the membership function to fuzzify the linguistic values of input and output variables for new expression.

5. Apply the concept of fuzzy logic [4] for new rules to execute the inference procedure and to defuzzify the output in order to obtain the output value.

III. Examples

A. One bit problem:Two fuzzy sets

The fuzzy system has two inputs A and B, and one output C. We discretize each input and output spaces into two regions in terms of two linguistic values which are defined as Positive Small (PS) and Negative Small (NS). The fuzzy rules are listed as followed:

If A is PS and B is NS, then C is PS;
If A is PS and B is PS, then C is PS;
If A is NS and B is PS, then C is NS;
If A is NS and B is NS, then C is NS.

First step:
The linguistic value of input and output variables are approximated as crisp values without considering overlapping. Each variable is 1 bit code, so the value of the variable can be represented as binary number: 1 is NS and 0 is PS.

Second Step:
Translate the word description of the *fuzzy rules* into *algebraic expression* and display them in the Karnaugh map, e.g.

$$C = A_1B_2 + A_1B_1 + A_2B_1 + A_2B_2 \tag{5}$$

			(PS) A_1 0	(NS) A_2 1
(PS)	B_1	0	0	1
(NS)	B_2	1	0	1

where, A and B are input variables, C is the output variable. A_1 and A_2 are linguistic values of variable A, they are represented as binary numbers: A_1 = 0= PS, A_2 = 1= NS. B_1 and B_2 are linguistic values of variable B, they are represented as binary numbers: B_1 = 0= PS, B_2 = 1= NS. In the entity of the map 0 and 1 are output linguistic values for C_1 and C_2

Third Step:
A 0 in column A_1 row B_1 of above Figure indicates that A_1B_1 is a minterm of C. Similarly, a 1 in column A_1 row B_2 indicates that A_1B_2 is a minterm. Minterms in adjacent squares of the map can be combined since they differ in only one variable. Thus, A_1B_1 and A_1B_2 combine to form A_1. Likewise, the A_2B_1 and A_2B_2 can be combined to form A_2 by looping the corresponding $0's$ on the map. Therefore, the algebraic simplification of a completely-specified set of rules can be derived as:

$$C = A_1 + A_2 \tag{6}$$

Fourth Step:
Assign the triangle membership function in Fig. 1 to the linguistic values (PS, NS) of A variable and transfer the algebraic expression to word description of fuzzy rules. They are described as:

If A is PS, then C is PS; If A is NS, then C is NS

Fifth Step:
Apply the concept of fuzzy logic to get output value

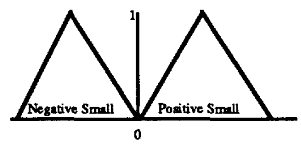


Fig. 1 The Membership Function of Input Variables

Validation:
Suppose the input A and B and C output spaces are the same, the real interval [-1,1]. The A, B and C are partitioned into the same two fuzzy sets:

$$\text{PS} = [0,1]$$
$$\text{NS} = [-1,0]$$

Consider the input data pair $A = 0.75$, $B = 0.5$ activates the original fuzzy rules. We combine the antecedent membership value with minimum with the conjunctive AND. Then, the correlation-minimum inference procedure [2] activates the output fuzzy set to some degree. We can then compute the fuzzy centroid with the following equation. To determine the specific output value of the fuzzy set, the control actions are weighted by their membership values and averaged. This weighted control action is known to provide a sluggish, yet more robust control.

$$\bar{C} = \frac{\sum_{j=1}^{p} w_j C_j}{\sum_{j=1}^{p} w_j} \tag{7}$$

where, $\bar{C}$ is the specific output value, and w_j is the scalar activation value of the j^{th} fuzzy rule's consequence which is the minimum of the n antecedent membership values, and C_j is the j^{th} value in the discretized output universe of discourse $C=\{C_1, C_2, ...C_j, ...C_n\}$.
The specific C value is calculated from the original rules to be 0.5. Similarly, the value of output is 0.5 from the new minimum rules. Hence, we can conclude that we can apply the minimum rules and obtain the same output values as when applying the original set of rules in complete knowledge base . However, the minimum set of rules provides less computation time than the original ones.

B. Two bits problem: Three fuzzy sets

The fuzzy system has the same input and output as the previous example. However, each input and output spaces are discretized into three regions in terms of three linguistic values which are defined as Zero (ZE), Positive Small (PS) and Negative Small (NS). The fuzzy rules are listed as follows:

If A is ZE and B is ZE, then C is ZE;
If A is ZE and B is PS, then C is NS;
If A is ZE and B is NS, then C is PS;
If A is PS and B is NS, then C is PS;
If A is NS and B is PS, then C is PS.

First step:
The linguistic value of input and output variables as crisp values without considering overlapping. Each variable is a two bit code, so the value of the variable can be represented as binary number: 00 is ZE, 01 is PS, 10 is NS and 11 are defined as don't care values (NIL) (NIL: not specified linguistic value in variable domain)

Second Step:
Translate the word description of the *fuzzy rule* into *algebraic expression* and display in the Karnaugh map, e. g.

$$C = A_1B_1 + A_1B_2 + A_1B_3 + A_2B_1 + A_3B_2 \tag{8}$$

			(ZE) A_1 00	(PS) A_2 01	(NS) A_3 10	NIL A_4 11
(ZE)	B_1	00	00			
(PS)	B_2	01	10	××	10	××
(NS)	B_3	10	01	01	××	××
NIL	B_4	11				

where, A and B are input variables, C is the output variable. A_1, A_2, A_3 and A_4 are linguistic values of variable A, they are represented as binary numbers such as A_1 = 00= ZE, A_2 = 01= PS, A_3 = 01= NS, A_4 = 11 = NIL. B_1, B_2, B_3 and B_4 are linguistic values of variable B, they are represented as binary numbers such as B_1 = 00= ZE, B_2 = 01= PS, B_3 = 10= NS, B_4 = 11= NIL. In the entity of the map 00, 01, 10 and ×× are output linguistic values for C_1, C_2, C_3 and C_4 respectively.

Third Step:
When choosing terms to form the minimum sum-of-products, all the outputs with similar values must be converted, but the ××'s are only used if they simply the resulting expression. A_1B_2 and A_3B_2 with don't-cares combining to form B_2 by looping the corresponding $10's$ on the map. Likewise, the A_1B_3 and A_2B_3 can be combined to form B_3 by looping the corresponding $01's$ on the map. Therefore, the algebraic simplification of an incompletely-specified rule can be derived as:

$$C = A_1B_1 + B_2 + B_3 \tag{9}$$

Fourth Step:
Assign the triangle membership function with overlapping between fuzzy sets in Fig. 2 to the linguistic values (ZE, PS, NS) of variable A and transfer the algebraic expression to word description of fuzzy rules. They are described as:

If A is ZE and B is ZE, then C is ZE;
If B is PS, then C is NS; If B is NS, then C is PS.

Fifth Step:
Apply the concept of fuzzy logic to get output value.

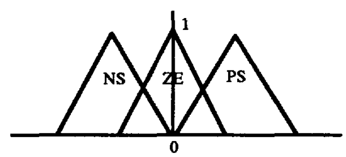


Fig. 2 The Membership Function of Input Variables

Validation
Suppose the input A and B and C output spaces are the same, the real interval [-1,1]. The A, B and C are partitioned into the same two fuzzy sets:

$$\mathrm{ZE} = [-0.2,\ 0.2]$$
$$\mathrm{PS} = [0,1]$$
$$\mathrm{NS} = [-1,0]$$

Consider the input data pair $A = 0.1$, $B = 0.01$ activating the original fuzzy rules. The specific C value is calculated at -0.0833 from Eq. 7. Similarly, the value of output is -0.0833 from the new minimum rules. Moreover, consider the values of input data to $A = 0.5$, $B = 0.3$. The specific C value cannot be obtained from the original fuzzy rules, because total weights of all activated rules are zero in the denominator of Eq. 7. However, the value of output can be calculated to be -0.5 from the new minimum rules. Hence, we conclude that we can apply the minimum rules in getting more complete results while the original rules lack complete information to compute the output value.

IV. Control Case

In this section, the control problem for the inverted-pendulum is used to interpret how the 3-bits concept can be used to reduce the fuzzy rules by forming a Karnaugh map. This control system has two inputs e(the error of the pendulum's angle from the balanced position) and $\dot{e}$ (the error of the pendulum's angular velocity) and one output u(torque). We discretize each input and output spaces into five regions in terms of five linguistic values which is defined as ZERO (ZE), Positive Small (PS), Positive Large (PL), Negative Small (NS) and Negative Large (NL). The fuzzy rules are listed as follows:

If e is ZE and $\dot{e}$ is NS, then u is PS;
If e is ZE and $\dot{e}$ is NL, then u is PL;
If e is ZE and $\dot{e}$ is PS, then u is NS;
If e is ZE and $\dot{e}$ is PL, then u is NL;
If e is ZE and $\dot{e}$ is ZE, then u is ZE;
If e is NS and $\dot{e}$ is ZE, then u is PS;
If e is NL and $\dot{e}$ is ZE, then u is PL;
If e is PS and $\dot{e}$ is ZE, then u is NS;
If e is PL and $\dot{e}$ is ZE, then u is NL;
If e is PS and $\dot{e}$ is NS, then u is NS;
If e is PL and $\dot{e}$ is NS, then u is NL;
If e is PL and $\dot{e}$ is PL, then u is NL;
If e is NS and $\dot{e}$ is PS, then u is PS;
If e is NL and $\dot{e}$ is PS, then u is PL;
If e is NL and $\dot{e}$ is PL, then u is PL.

First step:
The linguistic value of input and output variables as crisp value without considering overlapping. Each variable is three bit code, so the value of the variable can be represented as binary numbers: 010 is ZE, 011 is PS, 100 is PL, 001 is NS and 000 is NL. The other binary numbers 101, 110 and 111 are defined as don't care values (NIL) of the linguistic variable.

Second Step:
Translate the word description of the *fuzzy rule* into and *algebraic expression* display in the Karnaugh map, e.g.

$$\begin{aligned} u &= e_1\dot{e}_3 + e_1\dot{e}_4 + e_1\dot{e}_5 + e_2\dot{e}_3 + e_2\dot{e}_4 + e_3\dot{e}_1 \\ &+ e_3\dot{e}_2 + e_3\dot{e}_3 + e_3\dot{e}_4 + e_3\dot{e}_5 + e_4\dot{e}_2 + e_4\dot{e}_3 \\ &+ e_5\dot{e}_1 + e_5\dot{e}_2 + e_5\dot{e}_3 \qquad (10) \end{aligned}$$

	(NL) e_1 000	(NS) e_2 001	(ZE) e_3 010	(PS) e_4 011	(PL) e_5 100	nil e_6 101	nil e_7 110	nil e_8 111
(NL) $\dot{e}_1$ 000	x x x	x x x	100	x x x	000			
(NS) $\dot{e}_2$ 001	x x x	x x x	011	001	000			
(ZE) $\dot{e}_3$ 010	100	011	010	001	000			
(PS) $\dot{e}_4$ 011	100	011	001	x x x	x x x			
(PL) $\dot{e}_5$ 100	100	x x x	000	x x x	x x x			
nil $\dot{e}_6$ 101								
nil $\dot{e}_7$ 110								
nil $\dot{e}_8$ 111								

where, e and $\dot{e}$ are input variables, u is output variable. e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8 are linguistic values of variable e, they are represented as binary numbers such as e_1 = 000 = NL, e_2 = 001=NS, e_3 = 010 = ZE, e_4 = 011 = PS, e_5 = 100 = PL, e_6 = 100 = NIL, e_7 = 110 = NIL and e_8 = 111 = NIL. $\dot{e}_1$, $\dot{e}_2$, $\dot{e}_3$, $\dot{e}_4$, $\dot{e}_5$, $\dot{e}_6$, $\dot{e}_7$, $\dot{e}_8$ are linguistic values of variable $\dot{e}$, they are represented as binary numbers such as $\dot{e}_1$ = 000=NL, $\dot{e}_2$ = 001=NS, $\dot{e}_3$ = 010 = ZE, $\dot{e}_4$ = 011 = PS, $\dot{e}_5$ = 100= PL, $\dot{e}_6$ = 100 = NIL (don't care value), $\dot{e}_7$ = 110 = NIL and $\dot{e}_8$ = 111 = NIL. In the entity of the map 000, 001, 010, 011, 100, and 101 are output linguistic values for u_1= NL, u_2= NS, u_3 = ZE, u_4 = PS and u_5 = PL respectively and $\times\times\times$ is the don't care value.

Third Step:
To simplify the expression, $e_1\dot{e}_3$, $e_1\dot{e}_4$ and $e_1\dot{e}_5$ with don't cares combining to form e_1 by looping the corresponding same output value $100's$ on the map. Likewise, the $e_2\dot{e}_3$ and $e_2\dot{e}_4$ can be combined to form e_2, $e_4\dot{e}_2$ and $e_4\dot{e}_3$ can be combined to form e_4 and the $e_5\dot{e}_1$, $e_5\dot{e}_2$ and $e_5\dot{e}_3$ can be combined to form e_5 by looping the corresponding $010's$, $001's$ and $000's$ on the map respectively. Therefore, the algebraic simplification of an incompletely-specified rule can be derived as:

$$\begin{aligned} u &= e_1\dot{e}_3 + e_2\dot{e}_3 + e_3\dot{e}_1 + e_3\dot{e}_2 + e_3\dot{e}_3 + e_3\dot{e}_4 \\ &+ e_3\dot{e}_5 + e_4\dot{e}_3 + e_5\dot{e}_3 \qquad (11) \end{aligned}$$

Fourth Step:
Assign the triangle membership function with overlapping between fuzzy sets in Fig. 3 to the linguistic values (ZE, PS, NS) of variable A and transfer the algebraic expression to word description of fuzzy rules. They are described as:

If e is ZE and $\dot{e}$ is NS, then u is PS;
If e is ZE and $\dot{e}$ is NL, then u is PL;
If e is ZE and $\dot{e}$ is PS, then u is NS;
If e is ZE and $\dot{e}$ is PL, then u is NL;
If e is ZE and $\dot{e}$ is ZE, then u is ZE;
If e is NS, then u is PS; If e is NL, then u is PL;
If e is PS, then u is NS; If e is PL, then u is NL.

Fifth Step:
Apply the concept of fuzzy logic to get output value.

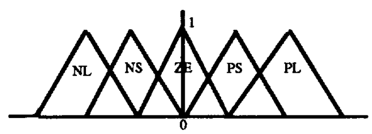


Fig. 3 The Membership Function of Input Variables

Comparison:
This nonlinear system is simulated to compare the performance of using the original rules and the minimum rules in controlling the balance of the inverted pendulum. The initial condition for the pendulum is set on e = 0.2 rad and $\dot{e}$ = 0.0 rad/sec. The response is shown in Fig. 4 and Fig. 5 respectively. The figures show that the controller of the pendulum from the original 15 rules and the reduced 9 rules have similar response. Next, we change the initial condition to e = 0.5 rad and $\dot{e}$ = 0.0 rad/sec. The results show that the response of the reduced rules become slightly faster than the original rules. Based on these results, we can see that the reducing rules can provide similar performance as that of the original rules in the control process. Moreover, the computation time is obviously shorter by using the reduced rules and it alleviates the complexity of implementation for the rule base.

We plotted the 3-D control surface to visualize the controller of the two set of rules.The control surfaces show the control u (vertical axes) corresponding to all combinations of values of the two input state variables error and error rate (horizon plane). We find the control surface of the reduced set of rules smoother, that is, it reflects that fact that the rule-based controller uses fewer rules than the original rule-based controller.

V. Conclusion

In this paper, we interpret and demonstrate the applicability of using Karnaugh maps to reduce rules base controller. We compare the output between the reduced set and the original set rules. We conclude that the expensive computation time will be reduced by using the minimum rules. Meanwhile, the reduced rules have similar performance as the original rules. As a result, this approach can provide a low-cost and robust means of design of the fuzzy rule-based controller. We would like to cautious that the results presented here are preliminary and a formal treatise of the matter is forthcoming. Limitations are obvious in the case of incomplete knowledge of the rules. The main goal of the study was to venture into "Conceptual" generalization using fuzzy logic.

References

[1] Gupta, M. M. and T. Yamakawa, "Fuzzy Logic in Knowledge-Based Systems, Decision and Control", North-Holland, 1988.

[2] Kosko, B.,"Neural Networks and Fuzzy Systems ", Printice Hall, Englewood Cliffs, NI07632, 1992.

[3] Roth, Charles H.,"Fundamentals of Logic Design", Third Edition, West Publishing Co. 1985.

[4] Zadeh, L. A. "Outline of a new approach to the analysis complex systems and decision processes',*IEEE Transactions on Systems, Man, and Cybernetics*, pp. 28-44, 1973.

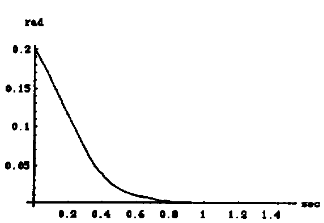


Fig. 4 The angle response from the original rules

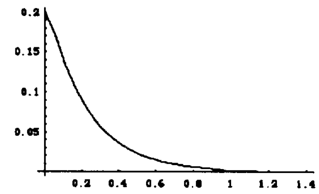


Fig. 5 The angle response from the reduced rules

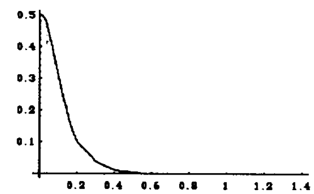


Fig. 6 The angle response from the original rules

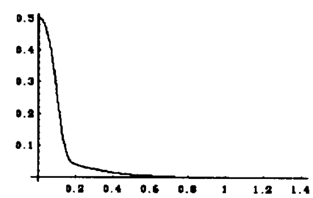


Fig. 7 The angle response from the reduced rules

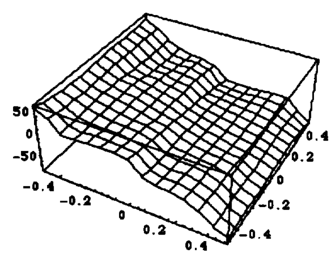


Fig. 8 The control surface from the original rules

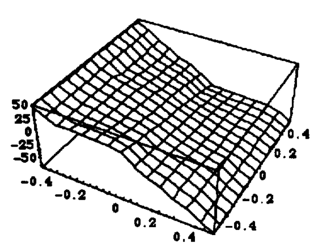
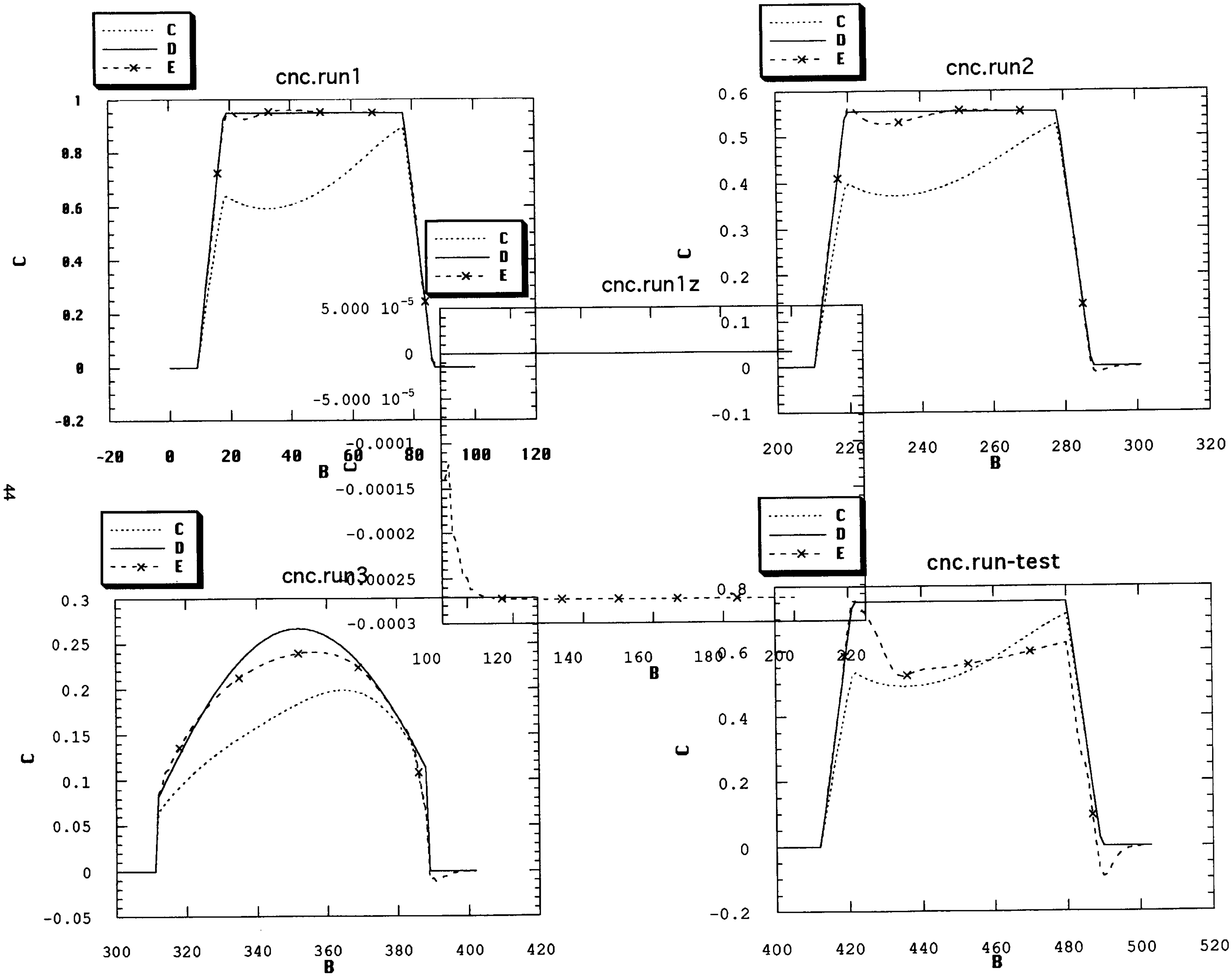


Fig. 9 The control surface from the reduced rules



cnc.run1
cnc.run2
cnc.run1z
cnc.run3
cnc.run-test
C
D
E
B