Fifth IFSA World Congress (1993), 887-890

Fuzzy Control as Self-Organizing Constraint-Oriented Problem Solving

Osamu KATAI†, Masaaki IDA†, Tetsuo SAWARAGI†, Kiminori SHIMAMOTO††, and Sosuke IWAI†

†Dept. of Precision Mechanics, Faculty of Engineering, Kyoto University, Sakyo-ku, Kyoto 606-01, JAPAN
TEL: +81-75-753-5201 (dial-in) FAX: +81-75-771-7286 (Dept. Office)
E-mail: D52754JPNKUDPC.BITNET, d52754@kuduts.kudpc.kyoto-u.ac.jp
††Nippon Steel Co. Ltd., Otemachi 2-6-3, Chiyoda-ku, Tokyo 100, JAPAN

Abstract: By introducing the notion of constraint-oriented fuzzy inference, we will show that it provides us ways of fuzzy control methods that has abilities of adaptation, learning and self-organization. The basic supporting techniques behind these abilities are "hard" processing by Artificial Intelligence or traditional computational framework and "soft" processing by Neural Network or Genetic Algorithm techniques. The reason that these techniques can be incorporated to fuzzy control systems is that the notion of "constraint" itself has two fundamental properties, that is, the "modularity" property due to its declarativeness and the "logicality" property due to its two-valuedness. From the former property, the modularity property, decomposing and integrating constraints can be done easily and efficiently, which enables us to carry out the above "soft" processing. From the latter property, the logicality property, Qualitative Reasoning and Instance Generalization by Symbolic Reasoning can be carried out, thus enabling the "hard" processing.

1. Introduction

We have already introduced a "constraint-oriented way of treating fuzziness" which can be applied to various kinds of problem solving including control and planning problems, etc [1], [2]. In this paper, we will first show the whole scope of our constraint-oriented approach for fuzzy information processing, particularly for the case of fuzzy control, and then show that this framework will provides us ways of fuzzy control that has abilities of adaptation, learning and self-organization. The basic supporting techniques behind these abilities are "hard" processing by Artificial Intelligence and traditional computational framework and "soft" processing by Neural Network and Genetic Algorithm techniques. The reason that these techniques can be incorporated to fuzzy control systems is that the notion of constraints itself has two fundamental properties, that is, the "modularity" property due to its declarativeness and the "logicality" property due to its two-valuedness. From the former property, the modularity property, decomposing and integrating constraints can be carried out quite easily and efficiently, which enables us to do the above "soft" processing. From the latter property, the logicality property, Qualitative Reasoning and Instance Generalization by Symbolic Reasoning can be carried out, thus enabling the "hard" processing.

2. The Whole Scope of Constraint-Oriented Way of Fuzzy Control

Fig. 1 shows the way how the above mentioned abilities can be substantiated for the case of fuzzy control. Suppose that we are given with experience of control, that is, a set of instances of observation-action pairs with their results and evaluation. Then, instance generalization by the methods of "Qualitative Reasoning (QR)" or "Instance Generalization (IG)" can be applied to yield generalized constraint regions on observation-action pairs. Their decomposition and further selection and refinement by Neural Network (NN) based techniques yield "betweenness rules" which are also introduced by the authors as a general kind of constraint-oriented rules and are the origin of constraint-oriented fuzzy inference [3]. More precisely, the decomposition of the betweenness rules yield "constraint-interval fuzzy inference rules" that are also introduced by the authors. Moreover, we have another route to derive the betweenness rules, that is, by the use of pairing of the instances which is assured to satisfy certain constraint conditiuons through certain methods of Qualitative Reasoning and Instance Generalization, and then by the use of Neural Network-based refinement, we have the betweenness rules [4], [5].

These constraint-interval fuzzy inference rules together with the values of observation variables, enable us to carry out constraint-interval fuzzy inference to derive "interval constraints" on the values of the actions, i.e., the control variables, whose conjunctive and/or disjunctive integration together with defuzzification yield the values of the control variables through which generation of next action is done

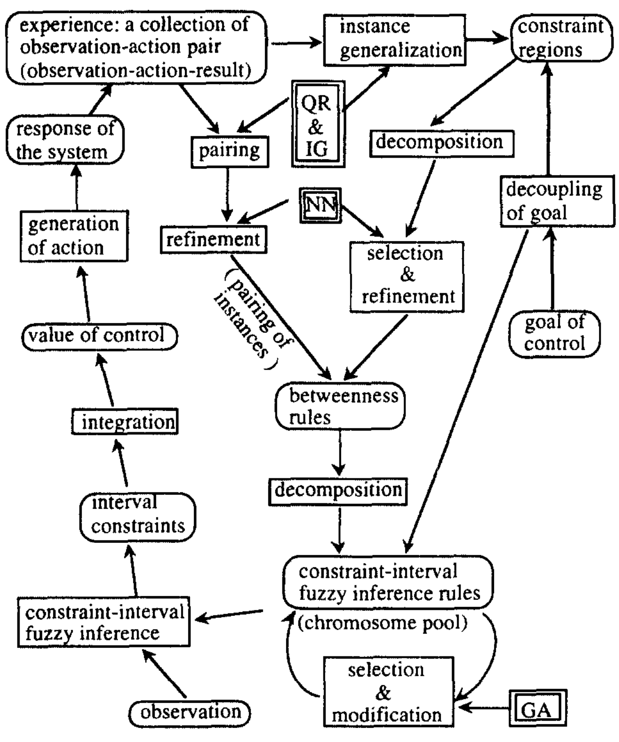

Fig. 1 Whole scope of constraint-oriented fuzzy control

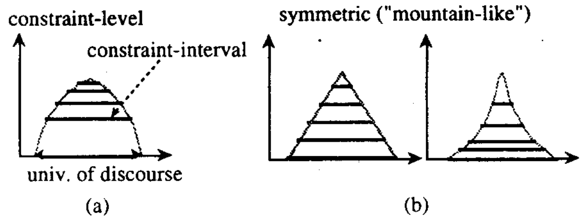

Fig. 2 A symmetric "mountain-like" constraint-interval fuzzy set

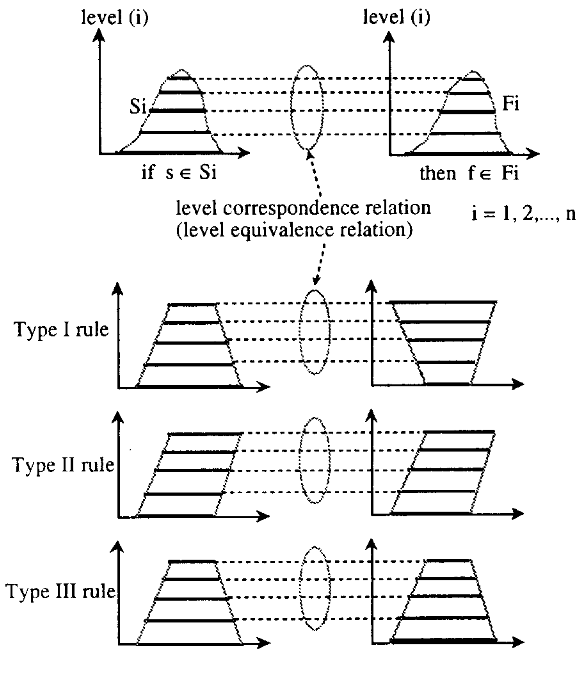

Fig. 3 Constraint-interval fuzzy inference rules ans their types

yielding another instance of observation-action-result triplet. Moreover, for complex control problems, we can decompose the goal of control yielding constraint regions on observation-action pairs from which we "decouple" the control problem into more simple subproblems.

Also, Genetic Algorithm (GA) techniques can be applied directly to the collection of constraint-interval rules, that is regarded to be constituting a pool of "chromosomes," i.e., a genetic pool, which are selected, modified and reproduced to refine the pool and the activated values of control variables by the observation are used to derive the value of control variables whose response from the environment is evaluated so as to be used in the GA-based selection, modification and refinement of the chromosomes [6].

3. Modularity and Logicality of the Constraint-Oriented Fuzzy Control Schemes

In this section, we will explain several key concepts in our framework. The first one is the notion of "constraint-interval fuzzy set." As shown in **Fig. 2**(a), it is given as an ordered collection of crisp intervals on the universe of discourse each of which represents a constraint called "constraint-interval". Namely, the grade axis (in the traditional Fuzzy Set Theory) is now regarded to be an ordinal scale axis, hence the sets in Fig. 2(b) are regarded to be the same as the one in Fig. 2(b).

Fuzzy inference rules whose "if (antecedent)" and "then (consequent)" parts are both regarded to be constraint-interval

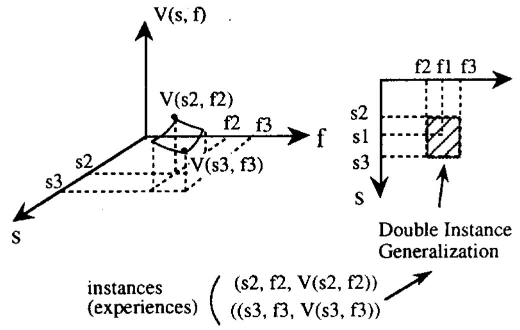

Fig. 4 Betweenness rule and its acquisition by Double Instance Generalization (DIG)

fuzzy sets are called "constraint-interval fuzzy inference rule" which says that if a variable s is in constraint-interval Si, then a variable f should be in the constraint-interval Fi for i = 1, 2, ..., n, as shown in **Fig. 3**. In this kind of rule, we have three typical rules, that is, type I, type II and type III rules. The former two types of rules are used to confine the permissible area of the values of control variables and hence are called "confining" rules. The latter type of rule, i.e. type III rule, is used to suggest the desirable area of control and is called "goal-seeking" rule.

More general constraint-oriented rules are given as betweenness rules such as if s is between s1 and s2, then f should also be between f1 and f2, which is written as

$$bet(s1: s2, s3) \dashrightarrow bet(f1: f2, f3).$$

This kind of rule can be derived by generalizing instance information, particularly by the method of "Double Instance Generalization (DIG)" introduced by the authors [3]. For instance, if we are informed of the qualitative shape or tendency of the values V(s, f) of the pair of the values of s and f as shown in **Fig. 4**, we can derive the following kinds of general rules from the triplets (s2, f2, V(s2, f2)) and (s3, f3, V(s3, f3)) such as

$$bet(s1: s2, s3)\ \&\ bet(f1: f2, f3)\ \&\ (s2 < s3\ \&\ f2 < f3)$$
$$\dashrightarrow bet(V(s1, f1): V(s2, f2), V(s3, f3)).$$

Thus, for assuring the least value V(s1, f1) of the observation-action pair (s1, f1), we will arrive at the following rule of control:

if bet(s1: s2, s3) and s2 < s3 & f2 < f3 hold,
then set f1 such that bet(f1: f2, f3) holds,

where < represent the partial order on s, which is sometimes given as a product of the orders on componential variables such as

$$s < s' \equiv s_j < s_j' \quad for\ j = 1, 2, ..., n.$$

Also, the betweenness relation is sometimes given as

$$bet(s1: s2, s3) \equiv s2 < s1 < s3\ \&\ s3 < s2 < s1.$$

Thus the betweenness rule given above is reduced to the following constraint-interval fuzzy inference rule:

$$if\ s2_j < s1_j < s3_j,\ j = 1, 2, ..., n\ and\ f2 < f3\ hold,$$
$$then\ set\ f1\ such\ that\ f2 < f1 < f3\ hold.$$

Suppose that we are given with a crisp constraint region on the pair of s and f as shown in **Fig. 5**. Then, decomposition of this region into rectangular areas yields type I and type II rules as shown in the figure.

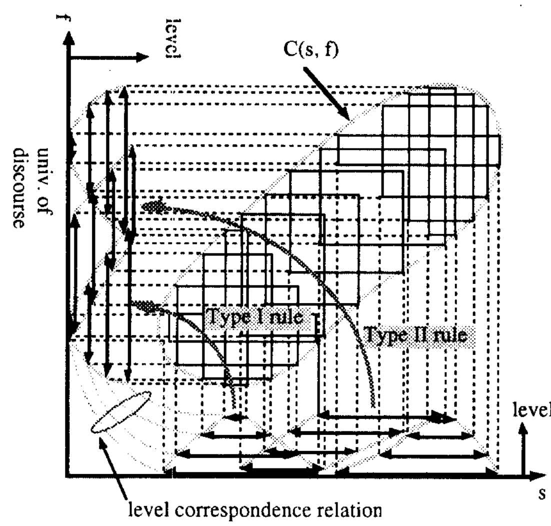

Fig. 5 Approximation of a crisp constraint by a collection of cofining (Type I & Type II) rules

Along with the ways mentioned above, we obtain the following procedure for deriving fuzzy control which assures the least value of the result of control.

(1) Select the instances of experience which have sufficiently good results.
(2) Delete the instances which are subsumed by other instances.
(3) Search for the pairing of the instances to compose constraint areas.
(4) If the constraint is of multiple levels, that is graded, we also have to organize the obtained area among different levels.

This selection can be carried out by Hopfield Network or Boltzmann Machine with an energy function evaluating the area of coverage with preferably less number of rectangles, that is intervals, having better continuity among different levels of constraints [4].

4. Decoupled Fuzzy Control Scheme for Cart-Pole Systems

The modularity of constraints enables the "decomposition" and "integration" of constraints. For instance, the control of a cart-pole system can be decomposed into that of the pole and that of the cart. The respective fuzzy inference on each side is then integrated by using "AND" composition to derive the constraint interval fuzzy set on the control variable, the external force to the cart, which is then defuzzified to yield the exact value of the control variable.

We set two kinds of goals on the cart and the pole, i.e., confining rule and goal-seeking rule.

Gc:
(Cg): goal-seeking for the cart,
(Cc): confining the cart,

Gp:
(Pg): goal-seeking for the pole,
(Pc): confining the pole,

The confining rule limit the move of the cart or the pole in a certain prespecified region, hence it permits their swinging motions, while the goal-seeking rule insists on their convergent behavior to the origin of the rail or to the vertical position of the pole. Thus we will have the following four cases of the goal of control:

case (1): (Cg) & (Pg)
case (2): (Cg) & (Pc)
case (3): (Cc) & (Pg)
case (4): (Cc) & (Pc)

The result of control is shown in **Fig. 6**. In case (1), the pole and the cart respectively insist on their own goal-seeking activities, hence the compromise between them results in their swinging motions from left to right as shown here. In this case, the behavior of the system is rather stable; it is seldom that the cart runs out of the rail or the pole falls down unless the initial condition is too severely set. In case (2), the pole insists on its own goal-seeking activity and the cart accommodates or adapts itself to the pole's behavior, hence the pole is held vertically and the cart swings smoothly right and left by using the full range of the rail. In case (3), on the contrary, the pole accommodates itself to the activity of the cart which results in so tight a condition that it is impossible to prevent the pole from falling down. Hence both of them move rather rapidly at first but are stopped suddenly by the falling down of the pole in the early stage. In case (4), both of them accommodate themselves to each other's behavior, hence they move very smoothly. The range of the cart movement is a bit small compared to that of case (3).

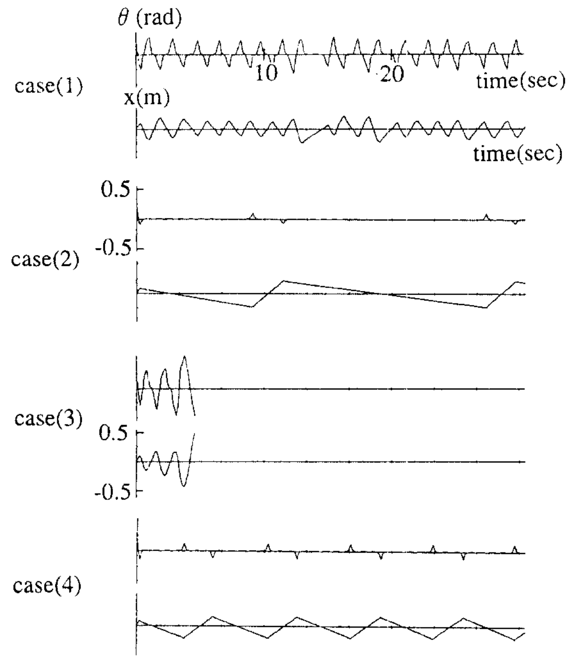

Fig. 6 The beheaviors of the cart-pole system by decoupled control

5. Application of Genetic Algorithm Techniques for Constructing Fuzzy Control Systems

As mentioned in Section 2, the collection of constraint-interval fuzzy inference rules can be refined by the use of Genetic Algorithm due to the modularized structures of fuzzy rules. In Genetic Algorithm, trial and error experiences are used to refine the population of solution candidates which are coded into sequences of symbols called chromosomes. The refinement of the population is done through the process which is derived from analogy to the evolutional genetic processes in creatures. These chromosomes are evaluated to calculate their degree of fitness to the environment (that is, the given problem). These chromosomes are then selected and reproduced by referring to their fitness values. A crossover

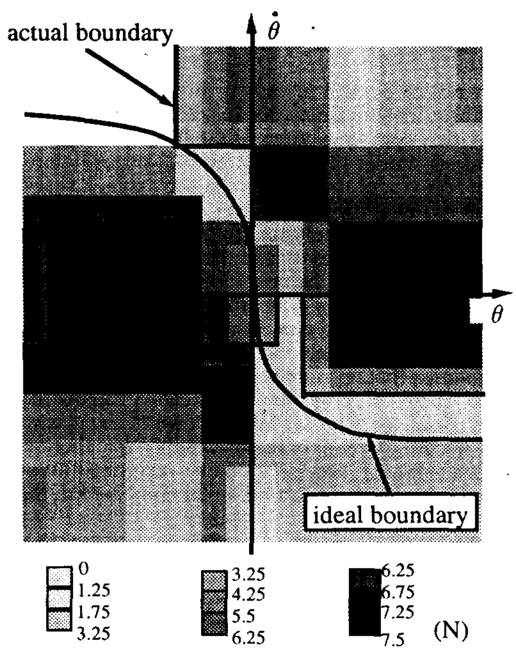

Fig. 7 The obtained value of control

between selected chromosomes subsequently takes place yielding the chromosomes, some of which are expected to be better than the original ones. Finally, mutation on the chromosomes is done to yield novel chromosomes. The whole process of this evolution can be regarded as a multi-point search for the optimal solution. The crossover operation shifts the points of search in a global fashion, while the mutation operation does so in a local fashion [6].

We regard each componential crisp constraint-interval rule as a chromosome, that is, we will adopt the Michigan approach instead of the Pittsburgh approach . In Pittsburgh approach, the whole inference system itself is coded as a chromosome, thus the size of the chromosomes becomes huge compared to that in the former approach. The evolutional operations, on the contrary, becomes complex in the Michigan approach compared to the the Pittsburgh approach. The main reason for adopting the former approach, the Michigan approach, is that we are searching for a method that will yield self-organizing mechanisms in the constraint-interval fuzzy inference systems. The self-organization is carried out by linking componential constraint-interval rules among different levels to yield a constraint-interval fuzzy rule.

In the Michigan approach, we have to evaluate the contribution of each fragmental componential rule making up the evolutional operations in the Michigan approach more difficult to be carried out than that of the Pittsburgh approach. Particularly, for the case of production systems for control, the evaluation on the whole control actions is usually done after a sequence of actions is applied, hence very complicated evaluation algorithms such as Bucket Brigade Algorithm are used. However, in our case, we use instantaneous evaluation of the control.

We have applied this method to the cart-pole control problem where the cart position is disregarded. The crossover probability and mutation probability are set as 0.6 and 0.01, respectively, and the number of chromosomes is set to be 90. The initial condition is set randomly in the region: $-0.4 < \theta < 0.4$, $-0.4 < \dot{\theta} < 0.4$.

We obtained various constraint-interval fuzzy inference rules ranging from type I to type III, that is, we have a large amount of diversity of chromosomes. This means that trial and error experience is still insufficient for converging to the optimal control rules. However, this diversity is also the origin of the adaptability of this system. **Fig. 7** shows the obtained value of control (external force to the cart) versus various values q and q. It is observed that there still remains the area which has to be learned. This happens due to that the advance of learning incidentally hides this dangerous area from the learning system.

In Genetic Algorithm process, the fitness value of each chromosomes is calculated according to value V(s, f) of the pair of s and f which is estimated by incremental revision by the following way:

$$V(s, f) = V(s) - C(f) - V(s(s,f)),$$

$$V(s) = \min_{f}\{V(s, f) + C(f)\},$$

where V(s), C(f), and s(s, f) stand for the value of of being at s, the cost of using f, and the next state of s after applying f, respectively. **Fig. 8** shows the estimated value of V(s), i.e., $V(\theta, \dot{\theta})$.

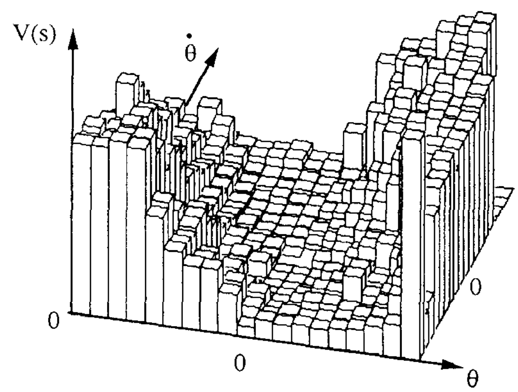

Fig. 8 GA-based estimation of $V(\theta, \dot{\theta})$, the minimum cost to attain the origin (0,0)

6. Conclusion

We have shown the reason and ways why and how the modularity and logicality properties of our constraint-oriented fuzzy control can be incorporated to efficient and effective ways of fuzzy control which can be constructed in a top-down and/or bottom-up manner by utilizing its learning and self-organizing and adaptivity abilities.

References

[1] Katai, O., Ida, M., Sawaragi, T., and Iwai, S.: Treatment of Fuzzy Concepts by Order Relations and Constraint-oriented Fuzzy Inference, Proc. of NAFIPS '90, A Quarter Century of Fuzziness (1990)

[2] Katai, O., Ida, M., Sawaragi, T., and Iwai, S.: Dynamic and Context-Dependent Treatment of Fuzziness from Constraint-Oriented Perspectives, Proc. of IFSA '91, Vol. on Artificial Intelligence, (1991)

[3] Katai, O., Sawaragi, T., and Iwai, S.: A Framework for learning and Reasoning via Order Relations, in Z. W. Ras (ed.), Methodologies for Intelligent Systems, Vol.4, Elsevier, (1989)

[4] Katai, O., Ida, M., Sawaragi, T., and Iwai, S.: Fuzzy Inference Rules and Their Acquisition from Constraint-oriented Perspectives, Proc. of IIZUKA '90 (1990)

[5] Katai, O., Ida, M., Sawaragi, T., Iwai, S.: Extracting Fuzzy Knowledge from Rank Ordered Sample Data via Instance Generalization and Connectionistic Computation, Proc. of IIZUKA '92, Vol. 2 (1992)

[6] Goldberg, D. E.: Genetic Algorithms, in Search, Optimization & Machine Learning, Addison Wesley (1989)