Fourth European Congress on Intelligent Techniques and Soft Computing, Aachen, Germany, September 2 – 5, 1996, Proceedings, Volume 1, pp. 209 – 218. (Reconstructed from a latex file and hardcopy)

ROUGH CONTROL APPLICATION OF ROUGH SET THEORY TO CONTROL

Toshinori MUNAKATA
Computer and Information Science Department
Cleveland State University
Cleveland, OH 44115
rnunakata@cis.csuohio.edu

Zdzislaw PAWLAK
Institute of Computer Science
Warsaw University of Technology
ul. Nowowiejska 15/19, 00 665 Warsaw, Poland
e-mail: zpw@ii.pw.edu.pl
and
Institute of Theoretical and Applied Informatics
Polish Academy of Sciences
ul. Baltycka 5, 44 000 Gliwice, Poland

1 Introduction

The two most significant developments in the field of artificial intelligence (Al) since 1990 are real world practicality and diversification [14,15]. Fuzzy set theory, for example, has grown to become a major scientific technology, applied to a couple of thousand systems for everyday industrial and commercial settings worldwide [16]. The area of neural networks is another example where extensive practical applications are expected [27]. An interesting question would be determining what the next generation of new Al technologies are and their potentials for practical applications.

Rough set theory is a relatively new mathematical and Al technique introduced in the early 1980's by Pawlak. The technique is particularly suited to reasoning about imprecise or incomplete data, and discovering relationships in this data. The main advantage of rough set theory is that it does not require any preliminary or additional information about data - like probability in statistics, basic probability assignment in Dempster Shafer theory of evidence or the value of possibility in fuzzy set theory.

Rough set theory overlaps, to some extend, with many other theories used to reasoning about data, in particular with Dempster Shafer theory and fuzzy set theory. Despite of these connections it can be viewed as an independent discipline in its own right.

Recently there has been a growing interest in rough set theory among researchers in many Al related disciplines. The theory has found many real life applications in many areas. The primary application of rough sets so far have been in data and decision analysis, databases, knowledge based systems, and machine learning. Some information about application of the theory can be found in [20].

Although there has been some research on rough control [4,8,9,10,11,12,13,23,25,27,28,29] their number and domains have been relatively small, and they are all academic rather than real life applications. It is worthwhile to mention in this context however that two rough controllers have been implemented in hardware [4,13] - displaying very attractive features. First of all their speed of operation is much higher than speed of any other compatible controllers, based on different principles. The work is in progress and a final evaluation of the results needs more research and experiments.

Observing the importance of rough control applications to practical problems, an informal Rough Control Research and Development Group was initiated in March 1995, and Toshinori Munakata has been named as the first Chair. The major objective of this article is to discuss motivation, principle, implementation, potential application areas, advantages and future problems of rough control - applications of rough set theory to control problems.

Whether rough control will succeed is yet to be seen, but there are several reasons for its high potential. Control has been the most successful application domain for recently evolved Al areas, such as fuzzy sets [7,16] and chaos theory [5]. In case of fuzzy systems, although there are several major application types such as fuzzy knowledge based systems and fuzzy databases, control has been most dominant in terms of the number. Control is also one of the most practical application domains for neural networks [27]. Simply stated, control is a mapping problem from inputs to outputs. When compared with symbolic Al, the task of control is often more clearly defined and its effectiveness is often easier to prove than, say, symbolic expert systems. If a machine can operate at 5%, or even 1%, more efficiently than before, we do not need to elaborate words to explain its validity. There are some similarities between fuzzy and rough sets. The experience gained in fuzzy control probably can be tested for rough control. More importantly, the fundamental strengths of rough set theory, such as making sense out of a complex information system where some of the information is possibly vague, appear to be particularly useful for certain types of control problems.

"Control" in this article refers to control of the various physical, chemical, or other numeric characteristics, such as temperature, electric current, flow of liquid or gas, motion of machines, various business and financial quantities (e.g., flow of cash, inventory control), etc. A control system can be abstracted as a box for which inputs are flowing into it, and outputs are emerging from it. Parameters can be included as parts of inputs or within the box, i.e., the control system. For example, consider a system that controls a room temperature distribution by heat and possibly cooling sources. The inputs may be the current room temperature distribution and parameters representing a target temperature distribution. The outputs can be the amounts of the heat sources to be applied. The control problem in general is to determine the numeric values of the outputs for given values of the inputs. That is, the problem is to develop a formula or algorithm for mapping from the inputs to the outputs.

In case of fuzzy control, the control box includes components such as fuzzification, fuzzy inference using fuzzy if-then rules, and defuzzification procedures. In case of classic control, the control can be a PID process (proportional, integral, and differential). In our rough control model, three major types of rough set applications will be investigated. One is to create such a control box by simplifying existing input-output mapping rules or generating new rules. The second is, in simple words, to perform input-to-output mapping based on rough set theory, i.e., in a sense, to operate the control box where the fuzzy components are replaced with the ones derived from rough set theory. The third is a hybrid system, where rough control is integrated with other techniques such as fuzzy or neural network control. We expect that such rough control systems may be particularly useful when either input and output values or input-to-output mapping rules are imprecise or incomplete. Rough control will give the "best' output values based on its criteria.

Generally, bad control can be time consuming or lead to poor performance, due to inefficient and ineffective control rules. A good algorithm will be relatively simple yet it performs efficient and stable control. For easy problem, a simple mathematical formulation may be sufficient- When problems get harder, traditional control techniques such as PID may not work well. for difficult problems, conventional methods are typically expensive and depend on mathematical approximations (e.g., linearization of nonlinear problem), which may lead to poor performance- This is the type of problems where fuzzy control has been successful, and where our major target domain is centered for rough control as well. In essence, fuzzy control deals with the hard problems by allowing gradual and continuous transition of logic and humanlike descriptive or qualitative expressions such as “moderately slow.” Rough control, on the other hand, deals with the hard problems by identifying degrees of significance for input and output attributes and by means of "quantization" to discern vague information. Rough control also allows descriptive or qualitative expressions. Both techniques aim at exploiting the tolerance for imprecision and uncertainty to achieve tractability, robustness, and low cost in practical applications.

2 Rough Sets, Basic Concepts

Rough set philosophy is based on the assumption that, in contrast to the classical set theory, we have some additional information (knowledge, data) about elements of the universe. Elements that display the same information are indiscernible in terms of the available knowledge and form elementary granules (blocks, classes etc.) of knowledge. These granules are called elementary sets or concepts, and can be considered as elementary building blocks of our knowledge. Elementary concepts can be combined into compound concepts, i.e., concepts that are uniquely defined in terms of elementary concepts. Any union of elementary sets is called a crisp set, and any other sets are referred to as rough (vague, imprecise). With every set X we can associate two crisp sets, called the lower and the upper approximation of X. The lower approximation of X is the union of all elementary set which are included in X, whereas the upper approximation of X is the union of all elementary set which have non-empty intersection with X. In other words the lower approximation of a set is the set of all elements that surely belongs to X, whereas the upper approximation of X is the set of all elements that possibly belong to X. The difference of the upper and the lower approximation of X is its boundary region. Obviously a set is rough if it has non empty boundary region; otherwise the set is crisp. Elements of the boundary region cannot be classified, employing the available knowledge, either to the set or its complement. Approximations of sets are basic operation in the rough set theory and are used as main tools to deal with vague and uncertain data. Approximations are used to define all other operations needed in data analysis in the rough set approach.

Precise mathematical formulation of the above ideas can be found in the enclosed literature, in particular in [19]. In this paper we will limit ourselves to illustrate them intuitively by means of an example.

Rough set theory is mainly used to discover patterns in data, data reduction and evaluation, etc. Data are often available in a form of data tables, known also as *attribute-values tables* or *information systems*. An information system is a table column of which are labeled by attributes, rows - by objects and entries of the table are attribute values.

An example of a simple information system is presented in Table 1.

Store	E	Q	L	P
I	high	good	no	profit
2	med.	good	no	loss
3	med.	good	no	profit
4	no	avg.	no	loss
5	med.	avg.	yes	loss
6	high	avg.	yes	profit

Table I

In the table six stores are characterized by four attributes, shown below:

E - empowerment of sales personnel,
Q - perceived quality of merchandise,
L - high traffic location,
P - store profit or loss.

Let us observe that each store has different description in terms of attributes E, Q, L and P, thus all stores may be distinguished (discerned) employing information provided by all attributes.

However, stores 2 and 3 are indiscernible in terms of attributes *E*, *Q* and *L*, since they have the same values of these attributes. Similarly, stores 1, 2 and 3 are indiscernible with respect to attributes *Q* and *L*, etc.

Each subset of attributes determines a partition (classification) of all objects into classes having the same description in terms of these attributes. For example, attributes *Q* and *L* aggregate all stores into the following classes {1, 2, 3}, {4}, {5, 6}.

Suppose we are interested in the following problem: what are the characteristic features of stores having profit (or loss) in view of information available in Table 1. In other words, the question is whether we are able to describe set (concept) {1, 3, 6} (or {2, 4, 5}) in terms of attributes *E*, *Q* and *L*. It can be easily seen that this question cannot be answered uniquely in our case, since stores 2 and 3 display the same features in terms of attributes *E*, *Q* and *L*, but store 2 makes a profit, whereas store 3 has a loss. Thus information given in Table I is not sufficient to answer this question. However, we can give a partial answer to this question. Let us observe that if the attribute *E* has the value *high* for a certain store, then the store makes a profit, whereas if the value of the attribute *E* is *low*, then the store has a loss. Thus, in view of information contained in Table 1, we can say for sure that stores I and 6 make a profit, stores 4 and 5 have losses, whereas stores 2 and 3 cannot be classified as making a profit or having losses. Therefore we can give approximate answers only. Employing attributes *E*, *Q* and *L*, we can say that stores I and 6 *surely* make a profit, i.e., *surely* belong to the set {1, 3, 6}, whereas stores 1, 2, 3 and 6 *possibly* make a profit, i.e., possibly belong to the set {1, 3, 6}. We will say that the set {1, 6} is the *lower approximation* of the set (concept) {1, 3, 6}, and the set {1, 2, 3, 6} - is the *upper approximation* of the set {1, 3, 6}. The set {2, 3}, being the difference between the upper approximation and the lower approximation is referred to as the *boundary region* of the set {1, 3, 6}.

If in an information system we distinguish two classes of attributes, called *condition* and *decision* (*action*) attributes, then such a table is called a *decision table*. In Table I the attribute *P* can be regarded as a decision attribute, whereas the attributes *E*, *Q* and *L*, as condition attributes. Decision attributes specify decisions which should be performed if conditions, specified by condition attributes, are satisfied.

Thus each row of a decision table can be regarded as a *decision rule* of the form “if ... then...” Object in a decision table are in this case are understood as labels of corresponding decision rules. For example in Table 1 we have the following decision rules:

If (*E* = *high*) and (*Q* = *good*) and (*L* = *no*), then (*P* = *profit*),
If (*E* = *no*) and (*Q* = *avg.*) and (*L* = *no*), then (*P* = *loss*).

Rules having the same conditions but different decisions, are called *conflicting* or *inconsistent*; otherwise the rules are *nonconflicting* or *consistent*. For example, in Table I rules 2 and 3 are inconsistent, whereas the remaining rules are consistent. Consistent decision rules enable us to make certain decisions, in contrast to inconsistent rules, which do not determine unique decisions) whatsoever. The ratio of consistent rules to all rules can be used as a *consistency measure* of the decision table. If the consistency measure is one the decision table uniquely determine the decision process, and if it is less then one the decision process in adequately determined. Rough set theory provides means to generate, analyze and optimize sets of decision rules obtained from data tables. The decision table can be obtained as a result of observations (measurements), computer simulation or knowledge of an expert. Many methods of optimal decision rule generations from decision tables (simplifying of decision tables) using rough set theory can be found in the enclosed references [1,21]. The proposed method of data analysis have several advantages. Some of them are listed below.

• Provides efficient algorithms for finding hidden patterns in data.

• Finds minimal sets of data (data reduction).

• Evaluates significance of data.

• Generates minimal sets of decision rules from data.

• It is easy to understand and offers straightforward interpretation of results.

The method is particularly suited for parallel processing , but in order to exploit this feature fully a new hardware solution are necessary.

Several software systems based on rough set theory have been developed, e.g., LERS Rough Das, Rough Class, DATALOGIC and others.

Decision tables can be also used to specify a control processes. In this case decision rule are to be understand as control rules, which determine actions to be taken if certain conditions are fulfilled. Consequently rough set theory can be used to derive optimal set of control rules from the decision table. The problem will be discussed in more details in the next section.

3 Main Ideas of Rough Control

Imagine we want to perform delicate room temperature control for a sophisticated experiment, perhaps for biomedical or solid state physics. In this scenario, we need fine temperature control: the allowable temperature deviation range is, say, within 0.02 °C of the target temperature throughout the room. Furthermore, the homogeneity of the temperature distribution is required, i.e., the temperature difference between any two points of one meter distance must be less than, say, 0.01 °C. The difficulty of temperature control is compounded because of the various boundary conditions. For example, the current level of robotics is not good enough to make robots perform the experiment. That is, human technicians must be in the room, which themselves are complicated heat sources.

Solving the problem theoretically, for example, by the Navier-Stokes' equation for air flow, associated with thermodynamic equations for heat conduction, convection, and radiation, under such complicated boundary conditions is out of question. A practical approach is to develop empirical formula for control from experimental data for temperature distributions and various heat/cooling sources. Rough control may be used in various stages of such development.

The major component of the inputs (attributes) is the measured temperatures throughout the room. Since the temperatures have to be measured three dimensionally, many sensors will be required at least initially. Other factors, such as the human body heat source, can be added as a part of the inputs.

The control may by described by an input-output mapping table, which may look as follows:

Input and Parameters			Output	
pl	p2	p3 . . .	ql	q2 . . .
. . .				
+.03	+.0l	-.04 . . .	-.6	+.2 . . .
+.02	+.02	-.05 . . .	-.7	+.1 . . .
. . .				

Such a table can be constructed initially in various ways. For example, if there are any existing methods to approximate the mapping, they can be used. Also, experiments can be conducted by human experts, possibly involving trial-and-errors.

The input-output table can now be treated like a decision table and by employing techniques offered by the rough set theory we can generate from it an optimal set of control rules, which can be put and executed by a rough set controller. Let us discuses this from the control point of view.

Since maintaining many temperature sensors is expensive, lesser sensors are desirable. During the first stage, rough control may be used to reduce the number of sensors required to achieve the]required control. Similarly, some of output elements may be found insignificant and thus deleted. Or, rough control may suggest other possible ways for achieving the same results.

After the initial construction of the input-output mapping table, the system becomes operational. However, the operations probably require much fine tuning. For example, there will be a certain limit to the number of the table entries because of the space and efficiency. In other words, all the possible combinations of input-output values may not be included in the table. Also, the data are incomplete and inaccurate. Rough control may be used for fine tuning of such a circumstance. For example, the closest table entries are used as "zeroth approximation." Rough control then finds 'superposing corrections" to the zeroth approximation. Rough control may compute the corrections by first determining the input-to-output dependencies, then taking "the center of gravity' as in case of defuzzification. Generally, the use of traditional method for zeroth approximation and rough control for fine tuning may be conservative but probably safer than relying the total control on a new technology. Many other variations and extensions for employing rough control would be possible, depending on the types of applications. The values of these variables can be descriptive, such as pl = moderately low, p2 = normal, and ql = moderately high. Each of these entities (rows) represents a control rule, e.g., when temperature = 25, and pressure = 1.05, we should apply heat source = 0.3.

For certain applications, multiple, rather than single, tables may be more appropriate for the system. Such situations may occur, for example, for the following cases.

(i) A table is so large that it is easier to develop and more efficient to operate and maintain sub-tables than dealing with only one.

(ii) From the characteristics of the control system, it is natural to divide into multiple tables (e.g., cases for acceleration, deceleration, and no acceleration).

For even larger tables, we can have a multi-level table structure. When the size of the tables get large, how to efficiently search for specific rules will be a practically important issue. For fast system response, such as for real-time control, involving a large rule base, concurrent (parallel or distributed) computation may be necessary. Concurrent implementation of rough control should be relatively easy since all the control rules are represented in table form. For example, each of output may be determined independent of others by allocating an appropriate number of processors. In the following, we discuss different phases of rough control applications.

(1) Rule simplification and generation. One of strengths of rough set theory is data reduction and discovery. Given a set of control rules , rough sets can simplify the system by reducing certain rules to fewer but equivalent ones, or eliminating redundant input, output and rules Also, less important rules may be downgraded in priority or weighted by smaller factors. The target control systems can be existing ones by conventional methods such as described by human experts, PID or fuzzy control, or a new system to be created from scratch. Such refinement of input-to-output mapping is conceivable for systems used many years without realization of such possibility. These rule simplification and regarding processes may give the same result with possibly a smaller set of rules achieving higher system efficiency as a whole. Practical applications of traditional symbolic machine learning have been limited in comparison with, for example, knowledge based systems. Its recent growth rate after 1990, however, is one of the fastest among many Al techniques. Currently the most successful machine learning techniques appear to employ, for example, relatively straightforward rule induction method [2]. Observing these recent developments in traditional machine learning, we should not be over-optimistic but can expect reasonable successes in rule simplification and generation for rough control.

(2) Output generation. Under an assumption of control rules, output values are determined for "rough," i.e., imprecise or incomplete, input and parameter values and/or rules. The basic concept here is to explore the most typical feature of rough set theory. That is, in context of rough control, we perform input-to-output mapping under possibly an imprecise and incomplete environment. This type of rough control by output generation has high potentials when we look at proven successes of fuzzy control, the key elements contributed to those successes, and the similarities of the key elements in fuzzy and rough control. As stated earlier, the key elements for fuzzy control successes are the use of descriptive expressions and uncertain reasoning. Rough control also has these characteristics. We note that a combination of the

above (1) and (2) is also possible, i.e., acquisition of control rules is performed in the first phase, then applications of these rules may follow in the second phase.

(3) Derivation of numeric output values for any of the above.
Often, values of condition and decision attributes in rough sets are given in descriptive expression such as "slow." For control problems, we need to derive numeric output values. The situation is similar to the case of fuzzy control. Possible methods of deriving numeric output values include the following.

- Making the control rules fine enough to directly produce numeric values. For example, the closest match of the input table is determined, then the corresponding output values in the table give the answer.
- Depending on the number of input and output variables and their ranges, Method (i) may not give satisfactory answers. In such case, some kinds of a "weighted mean" of candidate output values can be computed. Given current input variable values, closest matching rules in the table are found, with a measure of "closeness" between the current and each of the rules. Then a weighted mean of the corresponding output values can be computed, with the higher closeness meaning the higher weights. This is similar to defuzzification process of evaluating the center of gravity in fuzzy control in a sense that a value is determined as a sort of a weighted mean of several values. This can also be viewed as a type of weighted interpolation, determining a desired point from several neighboring points.
- Use of consistency measure. When there are inconsistent control rules consistency measures may become useful to determine output values.

Hybrid systems can also implemented. The key concept of hybrid systems is to complement each other's weakness, thus creating new approaches to solve problems. This hybrid system concept is probably one of the most promising for practical applications. For example, fuzzy control has many established application cases while rough control has relatively few. Integrating rough control with successful fuzzy control cases could be relatively easy for accomplishing real world practicality of rough control. Possible hybrid systems can be categorized as follows.

(i) Rough + fuzzy systems. (A major rough control system is augmented by a fuzzy system.)
(ii) Fuzzy + rough systems. (A major fuzzy system is augmented by a rough control system.)
(iii) Rough + neural network systems.
(iv) Neural network + rough systems.
(v) Rough + other systems (e.g., PID, genetic algorithms, chaos).
(vi) Other + rough systems.

In addition, more than two systems, such as rough + fuzzy + neural network, can be integrated into one. Also, the components of a hybrid system such as rough and fuzzy control can simply coexist to help each other without major-augment distinction.

Similarly, neural network + rough system may enhance the system. One major bottleneck of the backpropagation model in terra of practicality is the slowness of the learning process. Rough sets may be used as a front-end of the neural network system to preprocess the training patterns or learning process, or as an aid for pre-designing the neural network architecture.

Genetic algorithms are computer models based on genetics and evolution. Their basic idea is the genetic programs work toward finding better solutions to problems, just as species evolve to better adapt to their environments. One common problem in genetic algorithms is the slowness of the evolution. As in case of neural networks, rough sets may be used to "clean up" parameter or gene values, especially there are certain vagueness involved in these values.

Similar hybrid systems are conceivable for rough + other type systems.

4 Conclusions

This article has presented a preliminary study on principles and potential applications of rough control. The most important current problem is to further conduct a more detailed feasibility study for specific applications for which tough control fits the best and to actually implement prototypes and real world fielded applications.

The types of control for which rough control may be particularly useful are as follows:

(1) A large complex system: For example, the number of input, parameter, and output variables, or the number of rules is so large that it is difficult to simplify the system by hand or Other techniques. Also, when a system has been built over years, adding more and more rules to the existing rule set, it may be difficult to find redundant or simple but equivalent rules. For these situations, rough control will improve the system efficiently by refining the variable sets and rules.

(2) A system that involves incompleteness or impreciseness for input, parameter, output, or rules. Fuzzy control also deals with vagueness for these attributes, but in a different way.

For future development, major steps are:

(1) Selection of specific applications where rough control fits the best.

(2) Planning. e.g., how to tailor the given control problem to fit the rough control approaches described in this article.

(3) Design. e.g., how to structure a large amount of input-to-output control information, possibly using multi-level tables. How to search for these tables to find matching rules.

(4) Implementation and testing.

(5) Problems such as performance measurement and stability.

(6) Effective integration of hybrid systems.

These steps for each application may require significant development efforts. We hope this article serves as a stepping stone for future extensive applications of rough control.

References

[1] J. Bazan, A. Skowron, and P. Synak, “Discovery of decision rules from experimental data,” Conference Proceedings (RSSC'94) The Third International Workshop on Rough Sets and Soft Computing, San Jose State University, CA, November 10-12, 1994, 526-535.

[2] C.C. Chan, “Incremental learning of production rules from examples under uncertainty: a rough set approach,” Int. J. of Software Engineering and Knowledge Engineering, Vol. 1, No. 4, 439-461, 1991.

[3] M. R. Chmielewski, J. W. Grzymala-Busse, N. W. Peterson and S. Than, “The rule induction system LERS-A version for personal computers,” Foundations of Computing and Decision Sciences 18, 181-212, 1993.

[4] E. Czogala, A. Mr6zek, and Z. Pawlak, “The idea of a rough fuzzy controller and its application to the stabilization of a pendulum-car system,” Fuzzy Sets and Systems 72, 61-73, 1995.

[5] W. Ditto and T. Munakata, “Chaotic systems: Principles and applications,” Communications of the ACM, Vol.38, No. 11, 96-103, 1995.

[6] X. Hu, and N. Cercone, “Mining Knowledge Rules from Databases: A Rough Set Approach,” 12th International Conf. on Data Engineering.

[7] C.C,. Lee, “Fuzzy logic in control systems: fuzzy logic controller, Parts I and II”, IEEE Trans. Sys. Man. Cyb., 20, 2, 404-435, 1990.

[8] T.Y. Lin, “Fuzzy Logic Controller and Rough Logic”, Soft Computing: Rough Sets, Fuzzy Logic, Neural Network, Uncertainty Management and Knowledge Discovery, Society of Computer Simulation, 125-129, 1995.

[9] T.Y. Lin, “Fuzzy Controllers- An Integrated Approach Based on Fuzzy Logic, Rough sets, and Evolutionary Computing”, CSC'95 Workshop on Rough Sets and Database Mining, 1995.

[10] T.Y. Lin, “Rough-Fuzzy Controller for Complex Systems”, The Fourth Annual International Conference on Fuzzy Theory and Technology, Proceedings of Second Annual Joint Conference on Information Science Wrightsville Beach, North Carolina, 18-21, Sept. 28-Oct. 1, 1995.

[11] A. Mr6zek, “Comparative analysis of the human-experts inference models according the theory of rough sets,” Podstawy Sterowania, Vol. 14, No. 3, 197-213, 1984.

[121 A. Mr6zek, “Use of rough sets and decision tables for implementing rule-based control of industrial processes,” Bulletin of the Polish Academy of Sciences, T.Sc., Vol. 34, No. 5-6, 357-371, 1986.

[13] A. Mr6zek and L. Plonka, “Rule-based stabilization of the inverted pendulums”, Computational Intelligence, Vol. 11, No. 2, 1995.

[14] T. Munakata (Guest Ed.), “Commercial and Industrial Al,” Communications of the ACM, Vol. 37, No. 3, 23-119, 1994.

[15] T. Munakata (Guest Ed.), “New Horizons of Commercial and Industrial Al,” Communications of the ACM, Vol. 38, No. 11, 1995.

[16] T. Munakata and Y. Jani, “Fuzzy Systems: An Overview,” Communications of the ACM, Vol. 37, No. 3, 69-76, 1994.

[17] T. Munakata, “Rough control: Basic ideas and applications,” the Proceeding of the Second Annual Joint Conference on Information Sciences, Wrightsville Beach, NC, 340-343, 1995.

[18] T. Munakata, “Rough control: A perspective,” Proceedings of CSC'95, 23rd Annual ACM Computer Science Conference Workshop on Rough Sets and Database Mining, Nashville, TN, 1995.

[19] Z. Pawlak, “Rough sets: Theoretical aspects and reasoning about data,” Dordrecht, Netherlands: Kluwer Academic, 1991.

[20] Z. Pawlak, J. Grzymala-Busse, R. Slowi6ski and W. Ziarko, “Rough sets,” Communications of the ACM, Vol. 38, No. 11, 88-95, 1995.

[21] A.Skowron: “Extracting laws from decision tables,” Computational Intelligence, 11(2), 371-388, 1995.

[22] A. Skowron, and Z. Suraj, “Discovery of concurrent data models from experimental data tables : A rough set approach,” Proceedings of the First International Conference on Knowledge Discovery and Data Mining, Montreal, August, 1995, AAAI Press, Menlo Park CA, 288-293, 1995.

[23] A.J. Szladow and W. Ziarko, “Adaptive process control using rough sets,” Paper 93-384, Instrument Society of America, 1421-1430, 1993.

[24] A. Szladow, and W. Ziarko, "Development of reaction engineering model by discovering rules from process data," Proceedings of the Workshop on Knowledge Discovery in Databases, AAAI National Conference on Artificial Intelligence, Anaheim, 1991.

[25] A. Szladow, and W. Ziarko, "Knowledge-based process control using rough sets." In R. Slowinski (ed.), Intelligent Decision Support by Experience: Rough Sets Approach. Kluwer Academic Publishers, 49-60, 1992.

[26] B. Widrow, D.E. Rumelhart, and M.A. Lehr, "Neural networks: Applications in industry, business and science," Communications of the ACM, Vol. 37, No. 3, 93-105, March, 1994.

[27] W. Ziarko, "Generation of control algorithms for computerized controllers by operator supervised training," in Modeling, Identification and Control, Proceedings of the Eleventh IASTED International Conference, M.H. Hamza, (Ed.), Innsbruck, Austria, 510-513, Feb., 1992.

[28] W. Ziarko, and J.D. Katzberg, "Rough sets approach to system modeling and control algorithm acquisition," IEEE WESCANEX 93. Communications, Computers and Power in the Modern Environment Conference Proceedings (Cat. No.93CH3317-5), 154-164, May, 1993.

[29] W. Ziarko, and J.D. Katzberg, "Control algorithm acquisition, analysis and reduction: A machine learning approach," Knowledge-based system diagnosis, supervision, and control, S. G. Tzafestas (Ed.), Plenum Press, 167-180, 1989.