Intelligent Control in Space Exploration: What Non-Linearity to Choose?

Hung T. Nguyen[1] and Vladik Kreinovich[2]

***Abstract*— This paper is a short overview of our NASA-supported research into the optimal choice of fuzzy control techniques for space exploration problems.**

I. Introduction: Intelligent Control is Necessary for Space Exploration

Control is necessary for space missions. For a space mission to be successful, it is vitally important to have a good control strategy for all possible situations. For example:

- For a *Space Shuttle*, it is necessary to guarantee the success and smoothness of docking, the smoothness and fuel efficiency of trajectory control, etc.
- For an *automated planet mission*, e.g., for a rover mission to Mars, it is important to control the spaceship's trajectory, and after that, to control the rover so that it would be operable for the longest possible period of time.

It is often difficult or impossible to apply methods of traditional control theory. In many complicated control situations, in particular, in many control situations related to space flights, methods of traditional control theory are difficult or even impossible to apply. The main reason for that difficulty is as follows:

- For *traditional control*, we must *know* (more or less *precisely*) the properties of the controlled system.
- However, *space missions* are usually sent to explore new phenomena, and must operate under extreme conditions. Therefore, our prior *knowledge* about the situation is *not complete*.

Even when we *do* know the system precisely, this description may be so complicated that computing the optimal control is computationally intractable [31].

The authors are with the [1]Department of Mathematical Sciences, New Mexico State University, Las Cruces, NM, 88003, USA, hunguyen@nmsu.edu and with the [2]Computer Science Department of the University of Texas at El Paso, El Paso, TX 79968, USA, vladik@cs.utep.edu. This work was supported by NASA Research Grant No. 9-757, and, in part, by NSF Grant No. EEC-9322370.

Intelligent control is needed. In general, in uncertain situations, where no routine methods are directly applicable, we must rely on the creativity and skill of the human operators. And, indeed, expert controllers are very good in controlling complicated processes (not only in space missions, but also in the chemical industry, in metallurgy, in business). These experts usually cannot explain their control strategy in precise mathematical terms, but they can describe their strategies in terms of natural language, by phrases like "If a Space Station is close, and the relative speed is medium, decelerate a little bit". So, in order to develop an *automated* controller,

we must somehow transform these informal rules into a precise control strategy.

The methodology of such transformations is called *intelligent control.*

Intelligent control is useful also for *non-automatic* control: For example, for manned space missions, there are astronauts who are the best in solving the control problems. We cannot clone the best controllers, but we want to have an automated device that would simulate the best experts, and thus help other astronauts to control the mission.

Intelligent control is possible. There exist several dozens of different methods for intelligent control. This activity started in the 1970's by L. Zadeh and Mamdani, and has now many important applications, ranging from the Japanese automated subway system to various appliances.

II. The choice of intelligent control methodology can drastically change the quality of the resulting control

The experience in applying intelligent control shows that sometimes a transformation method leads to unstable or non-smooth control. In such situations, a different transformation of rules into control would be more appropriate.

Usually the choice of an appropriate technique is made on a trial-and-error basis.

For space applications, it is necessary to have theoretical methods for selecting the best intelligent control methodology. Experiments performed at Johnson Space Center on the Shuttle and rover simulators, showed that intelligent control techniques really lead to high quality control of space missions and planet rovers.

However, most of the existing intelligent control techniques are semi-*heuristic*, in the sense that they rely partly on trial-and-error. This may be acceptable for an appliance, but it is definitely unacceptable to choose a technique on a trial-and-error basis for a billion-dollar project. So, we need *guaranteed* (theoretical) methods to choose an appropriate technique.

So, we must:

- *analyze the existing* intelligent control *techniques*, and
- *find out which* of these techniques *is the best* with respect to the basic optimality criteria: stability, smoothness, robustness, etc.;
- if for some problems, none of the existing techniques is of satisfactory quality, *design new, better intelligent control techniques.*

This paper is an overview. The results of our research are described in detail in the papers [1]–[32] and in the student thesis supported by this project [T1]. In this paper, we give a brief overview of these results.

Before we formulate these results, we will briefly remind the reader of the main features of the most widely used existing intelligent control methodology: fuzzy control.

III. Fuzzy control: in brief

Rules. Fuzzy control methodology starts with expert "if-then" rules, i.e., with rules of the following type:

If x_1 is A^j_1 and x_2 is A^j_2 and ... and x_n is A^j_n, then u is B^j,

where x_i are parameters that characterize the plant, u is the control, and A^j_i, B^j are the natural language terms that are used to describe the j^{th} rule (e.g., "small", "medium", etc).

Mamdani's transformation. The value u is a proper value for the control if and only if one of these rules is applicable.

Thus, the property "u is a proper control" (which we will denote by $C(u)$), can be, therefore, described as follows:

$$C(u) \equiv (A^1_1(x_1) \,\&\, A^1_2(x_2) \,\&\, \ldots \,\&\, A^1_n(x_n) \,\&\, B^1(u)) \vee$$
$$(A^2_1(x_1) \,\&\, A^2_2(x_2) \,\&\, \ldots \,\&\, A^2_n(x_n) \,\&\, B^2(u)) \vee$$
$$\ldots$$
$$(A^K_1(x_1) \,\&\, A^K_2(x_2) \,\&\, \ldots \,\&\, A^K_n(x_n) \,\&\, B^K(u))$$

Membership function. The natural language terms are described by *membership functions*, i.e., we interpret $A^j_i(x)$ as $\mu^j_i(x)$, the number (called *degree of belief*) that describes the expert's degree of belief that a given value x satisfies the property A^j_i. Similarly, $B^j(u)$ is represented as $\mu_j(u)$.

"And" and "or" operations. The logical connectives & and $\vee$ are interpreted, in this context, as operations $f_\&$ and $f_\vee$ on degrees of belief. The most frequent choices of these operations are $\min(a,b)$ and $a \cdot b$ for $f_\&(a,b)$, and $\max(a,b)$ and $a+b-a\cdot b$ for $f_\vee(a,b)$.

After these interpretations, we can form the membership function for control:

$$\mu_C(u) = f_\vee(p_1, ..., p_K),$$

where

$$p_j = f_\&(\mu_{j,1}(x_1), \mu_{j,2}(x_2), ..., \mu_{j,n}(x_n), \mu_j(u))).$$

Defuzzification. The system must supply a control, so we must end up with a single value u of the control that will actually be applied. An operation that transforms a membership function into a single value is called a *defuzzification.* To complete the fuzzy control methodology, therefore, we must apply some defuzzification operator F to the membership function $\mu_C(u)$ and thus obtain the desired value $\bar{u} = f_C(\vec{x})$ of the control that corresponds to $\vec{x} = (x_1, ..., x_n)$. The most widely used defuzzification procedure is *centroid defuzzification*

$$\bar{u} = \frac{\int u \cdot \mu_C(u)\,du}{\int \mu_C(u)\,du}.$$

IV. Preliminary Research: Is the Existing Intelligent Control Methodology Reasonable? Universally Applicable? Optimal?

Before we start fine-tuning the existing intelligent control techniques, we must first make sure that this methodology is indeed good, i.e.,

- that this methodology is *reasonable*, i.e., consistent both with common sense and with other successful formalisms proposed to represent human reasoning;

- that this methodology is *universally applicable*, i.e., in principle, it can be used for an arbitrary control situation; and finally,
- that this methodology is indeed *optimal* in some reasonable sense, i.e., it can, potentially, lead to the best possible control.

In our preliminary research, we have shown that this is indeed the case.

A. Fuzzy control is reasonable

Let us first show that fuzzy logic is indeed consistent with other formalisms proposed to describe commonsense reasoning. All these formalisms can be viewed as modifications of *classical* (2-valued) *logic*, the logic that describes the ideal reasoning. Both in classical logic and in its commonsense modifications, we start with elementary (atomic) statements and combine them by using logical connectives (such as "and", "or", "not") and quantifiers (such as "for all" and "there exists") into more complicated logical statements.

The classical 2-valued logic can be characterized by the following features:

- in classical logic, every elementary statement is *either true, or false*;
- these statements can be combined by basic *logical connectives* and *quantifiers*;
- the truth value of the resulting complicated statements is determined by the *rules of logic*: e.g., "for all x, $A(x)$" is true if and only if the statement $A(x)$ is indeed true for all x.

At first glance, these features sounds perfectly reasonable. However, in real life, our reasoning does not always follow these rules:

- First of all, in real life, we are often *not sure* whether a certain statement is *true or false.*
  To describe this "un-sureness", it is desirable, in addition to the classical truth values "true" and "false", to have *intermediate degrees of belief.* If we add such degrees of belief, we get a modification of a classical logic that is called a *multiple-valued logic*:
  - Fuzzy logic is one particular case of this logic, in which we assume that there are *infinitely* many different degrees of belief that fill the entire interval $[0, 1]$.
  - Other multiple-valued logics, with *finitely* many different degrees of belief, have also been proposed to describe commonsense reasoning.

  Comment. To a certain extent, the third value — "unknown" — is already present in the classical logic system, in the sense that in a formal system, due to Gödel's theorem, a statement S can not only be *true* (when S is deducible from the theory) or *false* (when its negation $\neg S$ is deducible from a theory), but it can also be *unknown*, when neither the statement S itself, nor its negation can be deduced from the theory. In this sense, multiple-valued logics are not so much *replacing* the traditional logic, but they are *enriching* these logics by providing a finer structure of this "unknown". In particular, in [21], we show that traditional "paradoxes" of fuzzy logic, like the possibility of a statement to be (to some extent) true and, at the same time, (to some extent) false have natural analogies in classical logic.
- Second, in commonsense reasoning, the meaning of *connectives* is sometimes slightly *different* from its meaning in classical logic.
  For example, in classical logic, if one of the atomic statements $A_1, \ldots, A_n$ is false, then the compound statement $A_1 \& \ldots \& A_n$ is false. Not so in commonsense reasoning. For example, if the objective of a Space Shuttle's mission was to investigate a new geophysical area (A_1), to experiment with the signal transmission (A_2), to repair a satellite (A_3), etc., and to launch a new satellite (A_n), and the mission succeeded in all but one parts of this mission, then,
  - according to classical logic, we must say that the mission has failed, while
  - from the commonsense viewpoint, this mission was highly successful.

  There are crucial situations where all goals must be satisfied. Depending on the situation, the same word "and" can mean different operations. So, another approach to formalizing commonsense reasoning is to replace a *single*, say, "and" operation with *several* different operations that describe different commonsense meanings of "and".
  For better understanding, the difference between the classical logic and this new approach can be illustrated *graphically*: if we represent each statement by a point, then,
  - in classical logic, the combined statement "A and B" is well defined and thus, also represents a *point*, while
  - in this new approach, we get the whole *line* of different values depending on which interpretation of "and" we choose. In view of this interpretation, this approach is called *linear logic.*
- Finally, formulas of *classical logic* are known to be, in general, *algorithmically undecidable* in

the sense that there is no algorithm for finding the truth values of all composite logical statements.

Thus, in commonsense reasoning, when we want to estimate the truth values of these statements, we have to use some *heuristic* algorithmic techniques that, in general, only *approximate* the actual (non-algorithmic) truth values of these statements.

The corresponding approach to commonsense reasoning, in which we make *logical* reasoning algorithmic (i.e., implementable by a *program*), is called *logic programming.*

All three approaches turned out to be consistent with fuzzy logic:

- Formulas stemming from the *finite-valued logic* turn out to be exactly the formulas of *fuzzy logic* and fuzzy systems [23].
- Formalisms of *fuzzy* and *linear* logic are so close that we can justifiably call fuzzy logic "applied linear logic" [14, 22].
- Finally, logical equivalence stemming from *logic programming* turns out to be equivalent to the one that comes from *fuzzy logic* [18, 19, 20].

In all three cases, we not only show that fuzzy logic is *consistent* with the other formalisms, but we get a new *justification* of previously heuristic methods and formulas of fuzzy logic in terms of these other formalisms:

- *Finite-valued logic* helps to justify certain *and* and or operations of fuzzy logic (namely, min and max), the extension principle, fuzzy optimization, etc. [23].
- Axioms of *linear logic* justify the general properties of fuzzy *and* and *or* operations, such as associativity [14, 22].
- Finally, *logic programming* explains why in the most successful approach to fuzzy control — Mamdani's approach — implication is (weirdly) interpreted as "and" [8, 20].

B. Fuzzy control is universally applicable

Main result: fuzzy control is a universal approximation for (crisp) control strategies. It has been known that fuzzy control is a universal methodology for *traditional* control problems, with *one or several inputs*. To be more precise, it has been known that an arbitrary control can be, within an arbitrary accuracy, approximated by an appropriate fuzzy controller.

In [12, 26], this result is extended to *distributed* systems in which the state is described by a function, and to even more general control situations.

Auxiliary result: "fuzzy control"-type statements are a universal approximation for arbitrary fuzzy statements about control. The universality results mentioned above mean, in particular, that if we know a *crisp* (non-fuzzy) control strategy, then we can have a set of fuzzy control rules that approximate this strategy with any given accuracy.

In real life, however, we do not know this crisp strategy; instead, we have a (*fuzzy*) expert *knowledge* about it. In some cases, this knowledge is already formulated in terms of if-then fuzzy rules; in fuzzy control methodology, these rules are transformed into statements that only use connectives "and", "or", and "not".

However, in many other real-life cases, the fuzzy knowledge about control can be of much more general type. Therefore, the question appears: can we approximation an arbitrary fuzzy knowledge, that uses *arbitrary* logical *connectives* (including different versions of fuzzy implication), by a knowledge described in terms of "and", "or", and "not" fuzzy connectives? In other word, can we approximate an arbitrary fuzzy logical connective by a combination of these three basic ones?

It may seem, at first glance, that different unusual connectives are purely mathematical constructions, but, as we show in [32], even those connectives that may appear this way actually result from very natural axioms. In view of this result, it is desirable to consider the approximability of *arbitrary* logical connectives.

It turned out [24, 27] that in general, such an approximation of an arbitrary connective *is* possible, but only when we, in addition to these three basic connectives, allow *modifiers* such as "very", "slightly", etc. (that without the modifiers, such an approximation is impossible, is also shown in [13]).

C. Fuzzy control is optimal

Main result: fuzzy control is optimal (in some reasonable sense). In many space-related problems, we need the control results really fast; in these situations, to speed up the computations, it is natural to use *several processors* working *in parallel.* In [17], we have shown that if we want the *fastest* possible parallel *universal* computer (i.e., a computer with the ability to approximate an arbitrary function), then we get an architecture that corresponds exactly to *fuzzy control* methodology.

Thus, fuzzy control methodology is indeed, in the above sense, *optimal.*

Fuzzy control and neural control: which is the best? The paper [17] also contains a comparison between the fuzzy control methodology and another widely spread area of intelligent control: *neural network* control. Namely:

- If we consider *digital* processors, then *fuzzy control* is the optimal methodology.
- However, if we consider *analog* processors, then the same optimality criterion leads to the selection of *neural network control* methodology.

On one hand, neural network control is somewhat similar to the fuzzy control:

- the description of a fuzzy controller consists of elementary objects *rules*;
- the description of a neural network controller consists of elementary objects: *neurons*.

This similarity shows itself in the fact that for a natural expert system application, both approaches lead to the same class of methods [10].

On the other hand, this analogy is not complete, there are major differences in these two methodologies:

- In fuzzy control, *rules* (i.e., the corresponding elementary objects) come *directly* from the experts.
- On the other hand, for neural network control, *neurons* and their weights usually come from a lengthy and extremely time-consuming training.

This specific neural problem of determining the weights of the neurons naturally leads to the following two questions:

- First of all, how uniquely are these weights determined by the control that we are trying to approximate? The answer to this question — "yes, uniquely" — is given in [29].
- Second, how many neurons do we need to approximate a given control? In general, we may need a lot, but in [15, 16], a general class of controls is described for which the number of necessary neurons remains quite feasible.

V. The Optimal Choice of the Fuzzy Control Technique

A. Criteria for choosing a control

What do we want of the control?

- First, the control must *control*. In other words, if some external force has shifted the controlled object from the desired trajectory, then the system must return to the desired trajectory as soon as possible. In control theory, this property is called *stability*.
- Second, the control must lead to a *smooth* trajectory. *Smoothness* is extremely important both for manned and for automated space missions, because abrupt accelerations can be very uncomfortable to human astronauts and damaging for the sensitive equipment.
- The input data for control usually comes from sensors, and sensors are not 100% accurate. As a result, the measured values of the input variables that are used by the controller may be different from the actual values of the measured quantities. The ideal control must, therefore, work well not only for the input values, but also for the values that are close to the input ones. In other words, the uncertainty in the final control value, that is caused by the uncertainty of the input data, must be the smallest possible. Such controls are called *robust*.
- Finally, we want the computations of the control value to be as fast as possible. This computation speed is important for many control situations, but it is especially important for space missions, where decisions often need to be made in no time. As a result, in such situations, we must select the fastest possible algorithms, i.e., in computer science terms, algorithms with the *smallest possible computational complexity*.

Ideally, we would like to have a control that is the best according to *all* of these criteria, but in reality, these criteria are often conflicting with each other: e.g., if we want the system to be the *stablest* possible, i.e., return to the original trajectory as fast as possible, then a small deviation would result in a fast jerk back, making the trajectory *non-smooth*. In different situations, different criteria are most appropriate:

- For example, when we *dock* a Space Shuttle to a Space Station, the main criterion is *smoothness*, because non-smooth docking can seriously damage both the Space Shuttle and the Space Station.
- When we *track* a satellite by a radio signal, then our main goal is *not to lose* it; in this case, if we have accidentally deviated from the satellite's position, we want to get the signal back as soon as possible. In terms of our criteria, in this situation, the main criterion is *stability*.

In this report, we describe which techniques are the best w.r.t. these basic criteria.

Comment. In addition to the situations where one of the above-described criteria is the most appropriate, we may have more complicated situations in which the objective function is the result of a *trade-off* between different criteria. A critical survey of different methods of optimizing a (crisp) criterion under (possible fuzzy) constraints was published in

[4, 5, 6, 11].

Since we can have (potentially) infinitely many different combinations of criteria, we cannot explicitly describe the best control for all possible combinations. However, we hope that the general methods developed and used in this project can help in these more complicated situations as well.

B. How we solve the corresponding optimization problems

We must optimize under uncertainty. Fuzzy control is mainly used in situations when we do not have a complete knowledge about the controlled system; in other words, fuzzy control is mainly used in the presence of *uncertainty*. Hence, the problems of choosing the best technique (that we are interested in solving) are particular cases of *optimization under uncertainty*.

Optimization under uncertainty also occurs in fundamental physics. Choosing the best control technique is not the only real-life area where we must make conclusions in case of strong uncertainty. Strong uncertainty is also present in *fundamental* research, i.e., research in the areas where we have just started collecting data.

To handle such situations, theoretical physics has developed many useful approaches. One of these approaches that turned out to be one of the most successful in theoretical physics, is the so-called *group-theoretic* (*symmetry*) approach.

Group-theoretic (symmetry) approach in physics. It is well known that if a problem has a certain symmetry, then solving this problem becomes a much simpler task.

For example, if we are looking for a gravitational potential $\varphi(t, x_1, x_2, x_3)$ generated by several moving celestial bodies, then we must find a function of four variables by solving the corresponding partial differential equation. If, however, we know that there is only one body, and that this body is stationary and spherically symmetric, then we have two reasonable symmetries: invariance w.r.t. shift in time $t \to t + t_0$) and invariance w.r.t. rotations. In this situation, we must look for a solution that also has similar symmetries. If a function φ is invariant w.r.t. shift in time, this means that it does not depend on time at all. If a function is invariant w.r.t. rotations around the body's center O, this means that φ depends only on one variable: the distance r from a given point to the point O. Thus, the function φ turns into a function of 1 variable only: $\varphi(t, x_1, x_2, x_3) = \varphi(r)$, and a difficult-to-solve partial differential equation runs into a (much easier to solve) ordinary differential equation.

This idea of symmetries is used in physics not only to find solutions, but also to describe fundamental physical theories, the equations of most of which can be uniquely determined by the corresponding invariance requirements. This trend started with *special relativity* theory, whose main postulate was the postulate of *relativity*, i.e., invariance w.r.t. constant-speed motion. The notion of symmetry is so widespread that new physical theories are often formulated not in terms of different equations, but in terms of the corresponding symmetries.

Since symmetries are such a useful tool in physics, we want to use them for our problems as well.

Group-theoretic (symmetry) approach can be also used for selecting the best fuzzy control technique. In order to apply the ideas of symmetry to our problems, we must find out what the symmetries are in these problems.

There is a very natural symmetry here: namely, the very "fuzziness" of assigning *crisp* numbers to different "fuzzy" expert's degrees of belief means that different assignment procedures can be equally adequate. It is therefore natural to require the results of our processing the membership values (i.e., processing the results of this assignment) should not depend on which of the several possible equally adequate assignment procedures we choose. In other words, our processing algorithms must be invariant w.r.t *re-scaling*, i.e., w.r.t. moving from one scale of membership values to another possible scale.

It turns out that a natural formalization of this invariance can indeed solve the original optimization problems [4, 5]:

C. Results: the list of optimal methods

Optimal methods w.r.t. major optimality criteria. Part I. Choice of membership functions. The most *robust* membership functions are piecewise-linear ones [20, 25].

This result explains why the piecewise-linear membership functions are, at present, most frequently used.

Optimal methods w.r.t. major optimality criteria. Part II. Choice of "and" and "or" operations. (These results are (mainly) summarized in [4, 5, 20, 25].)

- If we are looking for the *most stable* control, then the best choice is to use $f_\&(a, b) = \min(a, b)$ and $f_\vee(a, b) = a + b - a \cdot b$ [13].
- If we are looking for the *smoothest* control, then the best choice is to use $f_\&(a, b) = a \cdot b$ and $f_\vee(a, b) = \min(a, b)$.

- If we are looking for the control that is *most robust*, then, depending on what we are looking for, we can get two different results:
  - if we are looking for the control that is the most robust *in the the worst case*, then the best choice is to use $f_\&(a,b) = \min(a,b)$ and $f_\vee(a,b) = \max(a,b)$ [20, 25];
  - if we are looking for the control that is the most robust *in the average*, then the best choice is to use $f_\&(a,b) = a \cdot b$ and $f_\vee(a,b) = a + b - a \cdot b$ [20, 25];
  - instead of minimizing the *average* error, we can try to minimize the corresponding *entropy* [11, 13]:
    - if we use the *average* entropy (in some reasonable sense), we get the same pair of optimal functions as for average error;
    - for an appropriately defined *worst-case* entropy (see also [30]) the optimal operations are $f_\&(a,b) = \min(a,b)$ and $f_\vee(a,b) = a + b - a \cdot b$.
- Finally, if we are looking for the control that is the *fastest to compute*, then the best choice is to use $f_\&(a,b) = \min(a,b)$ and $f_\vee(a,b) = \max(a,b)$.

Optimal methods w.r.t. major optimality criteria. Part III. Choice of defuzzification. In [11, 13], we show that the optimal defuzzification is given by the centroid formula.

Optimal methods w.r.t. additional optimality criteria: robustness w.r.t. possible computer malfunctions. *Robustness* can also mean robustness w.r.t. possible computer malfunctions. In principle, there are two possible types of malfunctioning:

- It can be a *temporary* malfunction, so all we need to do is *undo* the faulty operation and start again.
  In this case, we would like to have algorithms that make this "undoing" the easiest. In [3], we show that the possibility to undo is always present if and only if all membership functions are *fuzzy numbers*, i.e., if they have the simplest possible monotonicity structure (namely, they first increase, and then decrease).
- It can also be a *serious* malfunction, after which, for a certain period of time, further computations are impossible. In this case, we would like to have control implemented by an *interruptible* algorithm, i.e., by an algorithm that, if interrupted in the middle of the computations, still produces a reasonable control. In [1], it was shown that it is possible to transform every algorithm into an interruptible one without making its computation time much worse.

Optimal methods w.r.t. additional optimality criteria: optimal tuning in adaptive control. Similar optimization techniques have been applied to show that certain (fractionally linear) tuning formulas are the best in *adaptive* fuzzy control [T1].

Multi-criteria optimization. So far, we have considered situations in which we have a well-defined optimality criterion. However, in real life, we often have several *conflicting* criteria, especially when different participants of a project have slightly different aims. Optimization methods for such conflict situations are considered in [11].

VI. Additional method of improving fuzzy control

An additional method of improving the quality of fuzzy control was proposed in [7].

The main problem that this method is dealing with is that in traditional fuzzy control techniques, all rules are on equal standing. As a result, even when an expert explicitly says that for $x = 1$ the control should be exactly $u = 4$, the technique mixes this conclusion with other rules (like "when x is small, u should be small") and, as a result, returns the control $\bar{u}(1)$ that is often different from the desired $\bar{u} = 4$.

R. Yager and other researchers have proposed to remedy this situation by introducing the explicit *priorities* of different rules. In [7], we show that the same effect can be achieved without any additional information, simply by (slightly) modifying Mamdani's logical transformation.

A similar idea leads to a more adequate formalization of more complicated expert knowledge that includes *binary* properties like "x is approximately equal to y" [9].

VII. Auxiliary Results: Technical Diagnostics

Traditional fuzzy control techniques are designed mainly for the case when the controlled system functions well, and the question is only how to control it. In real life, however, and especially in space flights, malfunctions are quite possible. In this case, we have a problem of finding out which exactly component of the system is wrong.

If the system is simple and all its components are easily accessible, then we can simply test all its components. In space missions, however, systems are very complicated, and some components are difficult to access. As a result, we cannot simply test all the components, we need some intelligent algorithm to find the faulty component without testing all of them.

Similarly to fuzzy control, there are engineers who are very good in such a diagnosis, so it is natural to use their experience to diagnose the systems. Such technical diagnostic methods are developed for two possible types of malfunction:

- In [11], methods are described that find the faulty component for the case when the system stops functioning.
- In [2], methods are described that locate the faulty component in the situations when the system continues to function, but the value at least one of the critical parameters (that characterize the system's behavior) gets out of the interval of admissible values.

References

[1] M. Beltran, G. Castillo, and V. Kreinovich, "Algorithms That Still Produce a Solution (Maybe Not Optimal) Even When Interrupted: Shary's Idea Justified", *Reliable Computing* (to appear).

[2] M. Beltran and V. Kreinovich, "How To Find Input Variables Whose Influence On The Result Is The Largest, or, How To Detect Defective Stages In VLSI Manufacturing?", [28], pp. 34–37.

[3] B. Bouchon-Meunier, O. Kosheleva, V. Kreinovich, and H. T. Nguyen, "Fuzzy Numbers are the Only Fuzzy Sets That Keep Invertible Operations Invertible", *Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'96)*, Granada, Spain, July 1–5, 1996, pp. 1049–1054.

[4] B. Bouchon-Meunier, V. Kreinovich, A. Lokshin, and H. T. Nguyen, "On the formulation of optimization under elastic constraints (with control in mind)", *Université Paris VI et VII, Institut Blaise Pascal, Laboratoire Formes et Intelligence Artificielle LAFORIA*, Technical Report 94/19, October 1994.

[5] B. Bouchon-Meunier, V. Kreinovich, A. Lokshin, and H. T. Nguyen, "On the formulation of optimization under elastic constraints (with control in mind)", *Fuzzy Sets and Systems,* 1996, Vol. 81, No. 1, pp. 5–29.

[6] B. Bouchon-Meunier, H. T. Nguyen, V. Kreinovich, and O. Kosheleva, "Optimization with soft constraints: case of fuzzy intervals", In: L. Hall, H. Ying, R. Langari, and J. Yen (eds.), *NAFIPS/IFIS/NASA'94, Proceedings of the First International Joint Conference of The North American Fuzzy Information Processing Society Biannual Conference, The Industrial Fuzzy Control and Intelligent Systems Conference, and The NASA Joint Technology Workshop on Neural Networks and Fuzzy Logic, San Antonio, December 18–21, 1994*, IEEE, Piscataway, NJ, pp. 177–179.

[7] B. Friesen and V. Kreinovich. "How to improve Mamdani's approach to fuzzy control", *International Journal of Intelligent Systems*, 1995, Vol. 10, No. 11, pp. 947–957.

[8] O. Kosheleva, V. Kreinovich, and H. T. Nguyen, "Mamdani's Rule: a "weird" use of "and" as implication justified by modern logic", *Sixth International Fuzzy Systems Association World Congress*, San Paulo, Brazil, July 22–28, 1995, Vol. 1, pp. 229–232.

[9] V. Kreinovich. "Strongly transitive fuzzy relations: a more adequate way to describe similarity", *International Journal of Intelligent Systems*, 1995, Vol. 10, No. 12, pp. 1061–1076.

[10] V. Kreinovich, "S. Maslov's Iterative Method: 15 Years Later (Freedom of Choice, Neural Networks, Numerical Optimization, Uncertainty Reasoning, and Chemical Computing)", a chapter in *Problems of reducing the exhaustive search*, American Mathematical Society, Providence, RI, 1997, pp. 175–189.

[11] V. Kreinovich, "Random sets unify, explain, and aid known uncertainty methods in expert systems", in J. Goutsias, R. Mahler, and H. T. Nguyen (eds.), *Applications and Theory of Random Sets*, Springer-Verlag, 1996 (to appear).

[12] V. Kreinovich, H. T. Nguyen, O. Sirisaengtaksin, "On approximation of controls in distributed systems by fuzzy controllers", In: B. Bouchon-Meunier, R. R. Yager, and L. A. Zadeh (eds.), *Fuzzy Logic and Soft Computing*, World Scientific, 1995, pp. 137–145.

[13] V. Kreinovich, H. T. Nguyen, and E. A. Walker, "Maximum entropy (MaxEnt) method in expert systems and intelligent control: new possibilities and limitations", In: Ken Hanson and Richard Silver, Eds., *Maximum Entropy and Bayesian Methods, Santa Fe, New Mexico, 1995*, Kluwer Academic Publishers, Dordrecht, Boston, 1996 (to appear).

[14] V. Kreinovich, H. Nguyen, and P. Woj-

ciechowski, “Fuzzy Logic as Applied Linear Logic”, *BUlletin for Studies and Exchanges on Fuzziness and its AppLications (BUSEFAL)*, No. 67, July 1996, pp. 4–13.

[15] V. Kůrková, P. C. Kainen, and V. Kreinovich, “Dimension-independent rates of approximation by neural networks and variation with respect to half-spaces”, *Proceedings of World Congress on Neural Networks, WCNN'95, Washington, DC, July 1995*, INNS Press, NJ, 1995, Vol. I, pp. 54–57.

[16] V. Kůrková, P. C. Kainen, and V. Kreinovich, “Estimates of the Number of Hidden Units and Variation with Respect to Half-spaces”, *Neural Networks*, 1996 (to appear).

[17] R. N. Lea and V. Kreinovich, “Intelligent Control Makes Sense Even Without Expert Knowledge: an Explanation”, [28], pp. 140–145.

[18] H. T. Nguyen, O. M. Kosheleva, and V. Kreinovich, “Is the success of fuzzy logic really paradoxical? Or: Towards the actual logic behind expert systems”, *International Journal of Intelligent Systems*, 1996, Vol. 11, No. 5, pp. 295–326.

[19] H. T. Nguyen and V. Kreinovich, “On Logical Equivalence in Fuzzy Logic”, *Sixth International Fuzzy Systems Association World Congress*, San Paulo, Brazil, July 22–28, 1995, Vol. 1, pp. 209–212.

[20] H. T. Nguyen and V. Kreinovich, “Towards theoretical foundations of soft computing applications”, *International Journal on Uncertainty, Fuzziness, and Knowledge-Based Systems*, 1995, Vol. 3, No. 3, pp. 341–373.

[21] H. T. Nguyen and V. Kreinovich, “Classical-Logic Analogue of a Fuzzy ‘Paradox’ ”, *Proceedings of the 1996 IEEE International Conference on Fuzzy Systems*, New Orleans, September 8–11, 1996, Vol. 1, pp. 428–431.

[22] H. T. Nguyen and V. Kreinovich, “Fuzzy Logic, Logic Programming, and Linear Logic: Towards a New Understanding of Common Sense”, *Proceedings of NAFIPS'96, Biennial Conference of the North American Fuzzy Information Processing Society*, Berkeley, CA, June 20-22, 1996, pp. 546–550.

[23] H. T. Nguyen and V. Kreinovich, “Nested Intervals and Sets: Concepts, Relations to Fuzzy Sets, and Applications”, In: R. B. Kearfott et al (eds.), *Applications of Interval Computations*, Kluwer, Dordrecht, 1996, pp. 245–290.

[24] H. T. Nguyen and V. Kreinovich, “Kolmogorov's Theorem and its impact on soft computing”, In: Ronald R. Yager and Janusz Kacprzyk, *The Ordered Weighted Averaging Operators: Theory, Methodology, and Applications*, Kluwer, Norwell, MA, 1997 (to appear).

[25] H. T. Nguyen, V. Kreinovich, B. Lea, and D. Tolbert. “Interpolation that leads to the narrowest intervals, and its application to expert systems and intelligent control”, *Reliable Computing*, 1995, Vol. 3, No. 1, pp. 299–316.

[26] H. T. Nguyen, V. Kreinovich, and O. Sirisaengtaksin, “Fuzzy control as a universal control tool”, *Fuzzy Sets and Systems*, 1996, Vol. 80, No. 1, pp. 71–86.

[27] H. T. Nguyen, V. Kreinovich, and D. Sprecher, “Normal forms for fuzzy logic – an application of Kolmogorov's theorem” *International Journal on Uncertainty, Fuzziness, and Knowledge-Based Systems,* 1996, Vol. 4, No. 4, pp. 331–349.

[28] *Reliable Computing*, 1995, Supplement (Extended Abstracts of APIC'95: International Workshop on Applications of Interval Computations, El Paso, TX, Febr. 23–25, 1995).

[29] O. Sirisaengtaksin, P. Kainen, V. Kreinovich, and V. Kurkova, “For neural networks, even approximate function determines form”, *Proceedings of the First International Conference on Neural, Parallel, and Scientific Computations*, Atlanta, GA, May 28–31, 1995, Vol. 1, pp. 424–426.

[30] L. M. Rocha, V. Kreinovich, and R. B. Kearfott, “Computing uncertainty in interval based sets”, In: R. B. Kearfott et al (eds.), *Applications of Interval Computations*, Kluwer, Dordrecht, 1996, pp. 337–380.

[31] S. Smith and V. Kreinovich, “In Case of Interval Uncertainty, Optimal Control is NP-Hard Even for Linear Plants, so Expert Knowledge is Needed”, [28], pp. 190–193.

[32] I. B. Turksen and V. Kreinovich, “Fuzzy implication revisited: a new type of fuzzy implication explains Yager's implication operation”, In: Vladimir Dimitrov and Judith Dimitrov (eds.), *Fuzzy Logic and the Management of Complexity (Proceedings of the 1996 International Discourse)*, UTS Publ., Sydney, Australia, 1996, Vol. 3, pp. 292–295.

[T1] S. Hullahalli, *Why fractionally linear formulas work well in adaptive fuzzy control?*, Master Thesis, Department of Computer Science, University of Texas at El Paso, 1994.