Book Review of

"Fuzzy Sets Engineering"

by Witold PEDRYCZ

CRC Press, Boca Raton, FL, 1995, 332 p.

This book belongs to a recent trend of ideas which some have named "soft computing". It is rather characteristic of the current interest in the systems engineering literature for new modeling methods based on graphical structures and nested functions, such as neural nets. This approach is particularly active in the area of fuzzy systems engineering. Pedrycz's volume develops a system paradigm in which the processing module that computes outputs from inputs is not necessarily expressed at the same level of granularity as the available observed inputs nor the outputs required by the environment where this processing model in involved. A typical example of such a paradigm is precisely the fuzzy controller which receives numerical information, delivers numerical information, but whose encoding is in terms of linguistic rules. In other words this book deals with the interface between numerical and symbolic information. It also heavily relies on the notion of networks of local processors (which many call neural nets), and tends to suggest that most fuzzy models can be expressed as a network of local processors. The advantage of such an architecture is that, once suitably parametrized, it is amenable to data-driven optimization procedures which are commonly dubbed "learning methods". The originality of the book from the point of view of distributed structured models, is that Pedrycz suggests that each such model underlies a logical skeleton that can be expressed in terms of conjunction, disjunction, implication and the like. It is then possible to look at the network of processors as a fuzzy logic-based model. Moreover Pedrycz seems to consider that the information flow in the model need not be precise numbers, but could as well be fuzzy numbers, and he uses basic notions of possibility theory and fuzzy pattern matching to cope with this situation.

The book is organized in 10 chapters. The first one is an exposition of the above philosophy for systems engineering. The next four have the form of a fuzzy controller. Namely Chapter 2 is the interface between inputs and fuzzy sets and deals with issues of linguistic term sets on numerical universes. The basic question is how to represent a numerical possibly imprecise or fuzzy input by means of a referential of fuzzy sets. Chapter 3 is about the processing unit, and advocates the idea that fuzzy models can be viewed as networks of logic-based neurons. Here a zoology of such neurons is given: OR neurons, AND neurons, OWA neurons (after Yager) inclusion, difference and equivalence neurons for matching purposes. In each case, weighted versions of fuzzy logic neurons are described where the weights are supposed to be exploited by a learning method. Various such learning methods are recalled, especially gradient-based methods (derivatives of the minimum and maximum operations are suggested) and genetic-like algorithms. Chapter 4 exemplifies this logico-neuronal methodology to the computations of various things: distances, medians,

vector quantization, computational memories, pattern recognition units, fuzzy number ranking methods. A common feature of the methods is that they are all data-driven. Chapter 5 deals with the reconstruction of the output of a fuzzy processor from a set of evaluations pertaining to a referential of fuzzy sets. A typical reconstruction method is the center of area defuzzification. Chapter 6 to 9 focus on the fuzzy controller, again viewing it as a neuro-logical network, and considering learning issues. Chapter 6 stresses the analogy between learning a fuzzy controller and Hebbian learning of associative memories. It also discusses robustness and validation issues, and the switching between fuzzy and PID controllers. Chapter 7 is one of the most informative of the book, and consists of it is a presentation of 9 software development tools for fuzzy models. Addresses, phone numbers, fax numbers and sometime email addresses are enclosed. Chapter 8 is about fuzzy controllers in action. It tries to represent notions of controllability, stability and the like, again in the form of neuro-logical processor networks. It points out the possibility of improving an expert-originated fuzzy controller via learning and sketches the way of learning both a fuzzy model and its fuzzy controller from data. The material contained in the two last chapters is seldom found in a book. Chapter 9 explains what becomes of a flip-flop circuit when its logical states are replaced by a continuum of states (the unit interval). Fuzzy flip-flops invented by Hirota define a special family of dynamic systems where the state at time t + 1 (continuously) depends on the state at time t and of two inputs. They are given a neuro-logical architecture, and applied to state-dependent pattern classifiers. Chapter 10 is devoted to fuzzy Petri nets, where tokens are replaced by fuzzy logical values between 0 and 1. The author claims that such a scheme can encode rules, can also encode fuzzy controllers, and finally it seems as if the fuzzy Petri net hides one more neuro-logical network of local processors.

This book is in some sense very symptomatic of the difficulty to put together Systems Engineering and logic-based Artificial Intelligence. Traditional systems engineering relies on continuous blackbox models, function approximation, and is data-driven. On the contrary, traditional AI is based on knowledge representation, symbolic deduction methods, and avoids numbers. Fuzzy logic control in early times used to be knowledge driven, linguistically interpretable, but accepted numbers. With the arrival of neural networks the paradigm of fuzzy logic seems to have drifted away from artificial intelligence, because neural nets are essentially a new powerful tool for designing data driven blackbox models, and fuzzy rule-based systems look like some kind of exotic neural nets. By emphasizing the logical skeleton of neural networks Pedrycz tries to some extent to resist the current stream of engineering methods where the notion of fuzzy logic model tends to be reduced to just another universal approximator of numerical functions to be trained on data. Does this book succeed in demonstrating that the mathematics of fuzzy sets (fuzzy connectives, possibility theory, fuzzy arithmetic, ...) are useful for systems engineers? To the opinion of this reviewer, it does not completely succeed, perhaps because systems engineers actually don't need much of fuzzy set theory. To quote Pedrycz himself in his book (page 129): "Quite often the fuzzy controller is referred to as a fuzzy logic controller. At least the construct we are about to discuss does not strongly justify the use of any logical notions." To some extent one may equally wonder whether fuzzy controllers, as often used to-day, still justify the use of any *fuzzy set* notions. At least this the feeling one may also get by reading some chapters of this book. On the other hand it must be acknowledged that to-date, fuzzy control offers a systematic design vehicle

for building numerical control laws from linguistic information. The role of fuzzy sets is thus limited to the early stages of design, and the use of dedicated sotfware tools even makes an extensive study of fuzzy set theory almost unnecessary for practitioners who only want to synthetize a control law. This state of facts suggests that the fuzzy control methodology is mature enough and is no longer a topic for basic research for fuzzy set theory scholars.

The contribution of fuzzy sets to system science, if any, seems to be a methodology for systematically relating linguistic, knowledge-based representations to the language of continuous mathematics (numerical functions), recognizing that some numerical functions extend Boolean connectives. But this remark becomes useful in two types of problems only: those in which the available information is mainly human knowledge and those where the aim is to automatically produce knowledge in an easily understandable form. A (purely) data-driven fuzzy controller is thus a strange concept if the aim is to let a computer control a system, and if no expert control knowledge is available. A fuzzy rule-based function approximator is also something strange if the aim is to optimize the quality of fit to available data and not to qualitatively describe the behavior of a system. Indeed linguistically relevant fuzzy rules are partially incompatible with high precision of curve fitting (is this not Zadeh's incompatibility principle recalled on page 3 of this book). By emphasizing neural architectures and parameter identification via gradient methods, Pedrycz makes a contribution to systems engineering. Fuzzy logic is used as a language that enables a system engineer to construct a network of local processors interpreted as fuzzy logical connectives.

Some readers may find the contents of some chapters rather sketchy. Mathematical and technical details seem to be deliberately avoided and reference is made to published papers of the author instead. Besides Chapter 10 on fuzzy Petri nets is rather debatable. It is not clear that Petri net specialists will recognize their favorite tools. Especially, the main merit of Petri nets is to conjointly model non-determinism and asynchronism in concurrent discrete-event systems. But it is not clear where these two notions appear at all in a fuzzy controller. Of course, when implementing a fuzzy controller, data flow management questions may arise, that can be dealt with by means of a Petri net; but this issue would deserve a more extensive treatment. Quite another concept of fuzzy Petri nets appear in the work of Cardoso et al. (J. Cardoso, R. Valette, D. Dubois "Petri nets with uncertain makings", in: Advances in Petri Nets (G. Rozenberg, ed.), LNCS Vol. 483, Springer Verlag, Berlin, 1990, 64-78) where the fuzziness is due to a lack of knowledge about where tokens are.

Despite the above criticisms this book is worth reading because of its thought-provoking attempt to reconcile neural nets and Boolean logic. This book is about the mysterious borderline between data and knowledge, which nowadays still looks like a foggy area. It is a contribution to the huge task of making this borderline better understood. Unfortunately there is still a long way before neural nets speak in natural language due to the contribution of fuzzy logic, and anyway it is not clear that this very task should be carried out by systems engineers.

Didier DUBOIS