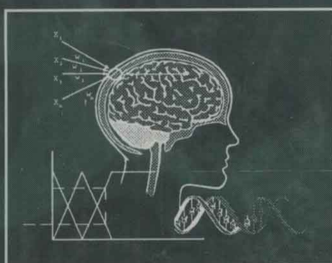
Edited by

Lakhmi C. Jain
Clarence W. de Silva



INTELLIGENT ADAPTIVE CONTROL

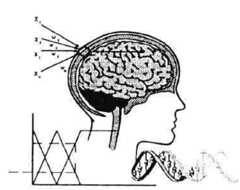
Industrial Applications

INTELLIGENT ADAPTIVE CONTROL

Industrial Applications

Edited by

Lakhmi C. Jain
Clarence W. de Silva

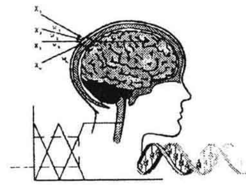


CRC Press
Boca Raton London New York Washington, D.C.

Library of Congress Cataloging-in-Publication Data

Catalog record is available from the Library of Congress.

International Standard Book Number 0-8493-9805-3
Printed in the United States of America 1 2 3 4 5 6 7 8 9 0
Printed on acid-free paper



The CRC Press

International Series on Computational Intelligence

Series Editor
L.C. Jain, Ph.D.

L.C. Jain, R.P. Johnson, Y. Takefuji, and L.A. Zadeh
Knowledge-Based Intelligent Techniques in Industry

L.C. Jain and C.W. de Silva
Intelligent Adaptive Control: Industrial Applications

L.C. Jain and N.M. Martin
Fusion of Neural Networks, Fuzzy Systems, and Genetic Algorithms: Industrial Applications

H.N. Teodorescu, A. Kandel, and L.C. Jain
Fuzzy and Neuro-Fuzzy Systems in Medicine

C.L. Karr and L.M. Freeman
Industrial Applications of Genetic Algorithms

L.C. Jain and Beatrice Lazzerini
Knowledge-Based Intelligent Techniques in Character Recognition

L.C. Jain and V. Vemuri
Industrial Applications of Neural Networks

PREFACE

Fuzzy logic, neural networks, and evolutionary computing have provided important tools and techniques for system control. Specifically, the field of Intelligent Control is fertile with techniques of computational intelligence; particularly, soft computing, that are, by and large, based on a fuzzy-neural-evolutionary framework. Intelligent control seeks to achieve good performance in machines, industrial processes, consumer products, and other systems, by using control approaches that, in a loose sense, tend to mimic direct control by experienced humans. Many of these techniques can learn, adapt to compensate for parameter changes and disturbances, and are able to provide satisfactory control even in incompletely-known and unfamiliar situations. With this backdrop, the main purpose of this book is to present some important techniques, developments, and applications of computational intelligence in system control.

Intelligent control is rapidly becoming an established field of education and research, not simply because of its pedagogical importance but also in view of its tremendous success in industrial applications. Most of the books in this field, however, are written by university educators and researchers, in the form of textbooks, research monographs, or collections of research papers. People who implement the techniques of intelligent control in industrial systems and products are often reluctant to publish their work due to proprietary restrictions, and the lack of direct professional benefit and opportunity. Of course, there are some university-industry collaborative activities where intelligent control techniques have been developed, designed, implemented, and evaluated, with the end result of successful commercial products and systems. Such collaborative work has a greater probability of entering into archival literature than those that are exclusively industrial developments. Facing this situation, we have made a serious attempt, in this book, to report truly industrial developments and applications of intelligent control.

This book consists of fourteen chapters. Chapter 1 provides an introduction to the fundamentals of neural networks, fuzzy logic, and evolutionary computing. This material forms the foundation upon which the remaining chapters are based. Chapters 2 through 7 fall into the category of theory and applications. In particular, these chapters will provide a more rigorous treatment of intelligent control and also will demonstrate some applications of the associated techniques. The remaining seven chapters primarily present industrial applications of intelligent control and soft computing. Applications presented in these chapters come from a variety of industries including transportation, petroleum, motor drive, and fish processing, The emphasis of the book is on the application of fuzzy logic, neural networks, and evolutionary computing. It is these three general approaches that are utilized either separately or cooperatively, in most applications presented in the book. Other knowledge-based techniques are also used in some applications, albeit in a lii manner; specifically, in Chapters 13 and 14.

The book provides a state-of-the art treatment in practical applicatior computational intelligence in system control. Introductory and advanced th design, practical implementation, and industrial use are covered in a coh manner. The book will be useful to engineers, scientists, researchers, and gra students, as a practical guide to theory and implementation of intelligent co particularly employing fuzzy logic, neural networks, and evolutionary computii

ACKNOWLEDGMENTS

We are very grateful to the contributors of this book for their aptitude, rigor professionalism. The significance of their individual contributions is reflected i overall utility of the book. We wish to thank Jan Vogel, Pieter Grimmerink Jerry Papke for their assistance in the preparation of the book. Our gra students and research colleagues have made important contributions to the co and development of the book, both directly and indirectly. In this context, we to particularly mention here, with thanks, Min-Fan Ricky Lee of National Res Council and the University of British Columbia, Scott Gu of Precix, Inc., Kurnianto of Tantus Electronics, Corp., Fakhri Karray of University of Wat and Jonathan Wu of National Research Council.

Lakhmi Jain
University of South Australia, Adelaide

Clarence de Silva
University of British Columbia, Vancouver.

THE EDITORS

Lakhmi C. Jain is the Founding Director of the Knowledge-Based Intelligent Engineering Systems (KES) Centre, Faculty of Information Technology, University of South Australia, Adelaide, Australia. His Ph.D. degree is in Electronic Engineering. A Fellow of the Institution of Engineers, Australia, he has received a number of awards for his papers including the Sir Thomas Ward Memorial Medal in 1996, and the Best Paper Award from the Electronics Association of South Australia, in 1995. Jain holds a patent (with Udina) for Skylight Light Intensity Data Logger. He is Founding Editor-in-Chief of the *International Journal of Knowledge-Based Intelligent Engineering Systems* and serves as an Associate Editor of the *IEEE Transactions on Industrial Electronics.* He has organized several international conferences and was appointed as Vice-President (International Liaison) by the Electronics Association of South Australia An expert in knowledge-based systems, artificial neural networks, fuzzy systems, and genetic algorithms with application to practical engineering problems, he has authored or co-authored over 100 books and research papers in his field. He is the Editor-in-Chief of the International Series in Computational Intelligence of CRC Press.

Clarence W. de Silva, Fellow ASME and Fellow IEEE, is Professor of Mechanical Engineering at the University of British Columbia, Vancouver, Canada, and occupies the NSERC Chair Professorship in Industrial Automation since 1988. He obtained his first Ph.D. from the Massachusetts Institute of Technology in 1978 and, twenty years later, another doctorate from the University of Cambridge, England. De Silva has served as a consultant to several companies including IBM and Westinghouse in the U.S.A., and has led the development of many industrial machines. He is a recipient of the Meritorious Achievement Award of the Association of Professional Engineers of BC, the Outstanding Contribution Award of the IEEE Systems, Man, and Cybernetics Society, and the Outstanding Large Chapter Award of the IEEE Industry Applications Society. He has authored 13 technical books, 10 edited volumes, over 100 journal papers, and a similar number of conference papers and book chapters. He has served on the editorial boards of twelve international journals and is the Senior Technical Editor of *Measurements and Control*, and Regional Editor, North America, of the *International Journal of Intelligent Real-Time Automation.* He has been a Lilly Fellow, Senior Fulbright Fellow to Cambridge University, ASI Fellow, and a Killam Fellow.

CONTENTS

1 INTELLIGENT CONTROL TECHNIQUES

A. Filippidis and L.C. Jain
Knowledge-Based Intelligent Engineering Systems Centre
University of South Australia
Adelaide, The Levels, SA 5095, Australia

C.W. de Silva
Industrial Automation Laboratory
Department of Mechanical Engineering
The University of British Columbia
Vancouver, Canada

Fuzzy logic is useful in representing human knowledge in a specific domain of application, and in reasoning with that knowledge to make useful inferences or actions. Artificial neural networks are massively connected networks that can be trained to represent complex nonlinear functions at a high level of accuracy. They are analogous to the neuron structure in a human brain. Genetic algorithms are optimization techniques that can evolve through procedures analogous to human evolution, where natural selection, crossover, and mutation are central. Computational procedures associated with these techniques fall into the area of soft computing. This chapter will introduce some fundamental techniques of fuzzy logic, neural networks, and genetic algorithms. The main focus here will be their use in intelligent control. The basic treatment given in this chapter will lay the groundwork for a more rigorous analysis and application presented in the subsequent chapters of the book.

1 Introduction

This introductory chapter will outline the core techniques that form the basis of the developments and applications presented in the remaining chapters of the book. The presentation is not intended to be exhaustive, in view of the purpose of the chapter. A more rigorous treatment of some of the topics introduced here may be found in the subsequent chapters.

The term "Intelligent Control" may be loosely used to denote a control technique that can be carried out using the "intelligence" of a human who is knowledgeable in

0-8493-9805-3/99/$0.00+$.50

the particular domain of control [1]. In this definition, constraints pertaining to limitations of sensory and actuation capabilities and information processing speeds of humans are not considered. It follows that if a human in the control loop can properly control a plant, then that system would be a good candidate for intelligent control. Information abstraction and knowledge-based decision making that incorporates abstracted information, are considered important in intelligent control. Unlike conventional control, intelligent control techniques possess capabilities of effectively dealing with incomplete information concerning the plant and its environment, and unexpected or unfamiliar conditions. The term "Adaptive Control" is used to denote a class of control techniques where the parameters of the controller are changed (adapted) during control, utilizing observations on the plant (i.e., with sensory feedback), to compensate for parameter changes, other disturbances, and unknown factors of the plant. Combining these two terms, one may view "Intelligent Adaptive Control" as those techniques that rely on intelligent control for proper operation of a plant, particularly in the presence of parameter changes and unknown disturbances.

Computational procedures of Fuzzy Logic (FL), Neural Networks (NNs), and Genetic Algorithms (GAs) fall into the class of "soft computing" techniques, which can be directly utilized in intelligent control, either separately or synergistically. In particular, fuzzy logic may be employed to represent, as a set of "fuzzy rules," the knowledge of a human controlling a plant. This is the process of knowledge representation. Then, a rule of inference in fuzzy logic may be used according to this "fuzzy" knowledge base, to make control decisions for a given set of plant observations. This task concerns "knowledge processing" [1]. In this sense, fuzzy logic in intelligent control serves to represent and process the control knowledge of a human in a given plant.

Artificial neural networks represent a massively connected network of computational "neurons." By adjusting a set of weighting parameters of an NN, it may be "trained" to approximate an arbitrary nonlinear function to a required degree of accuracy. Biological analogy here is the neuronal architecture of a human brain. Since intelligent control is a special class of highly nonlinear control, neural networks may be appropriately employed there, either separately or in conjunction with other techniques such as fuzzy control. Fuzzy-neural techniques are applicable in intelligent adaptive control, in particular, when parameter changes and unknown disturbances have to be compensated.

Genetic algorithms belong to the area of evolutionary computing. They represent an optimization approach where a search is made to "evolve" a solution algorithm that will retain the "most fit" components, in a procedure that is analogous to biological evolution through natural selection, crossover, and mutation. It follows that GAs are applicable in intelligent control, particularly when optimization is an objective.
Summarizing, the biological analogies of fuzzy, neural, and genetic approaches are: fuzzy techniques attempt to approximate human knowledge and the associated

reasoning process; neural networks are a simplified representation of the neuron structure of a human brain; and genetic algorithms follow procedures that are crudely similar to the process of evolution in biological species.

Modern industrial plants and technological products are often required to perform complex tasks with high accuracy, under ill-defined conditions. Conventional control techniques may not be quite effective in these systems whereas intelligent control has a tremendous potential. The emphasis of the present book is on practical applications of intelligent control, primarily using FL, NN, and GA techniques. The remainder of this chapter will give an introduction to some fundamental techniques of fuzzy logic, neural networks, and genetic algorithms, within the context of intelligent control [1].

2 Knowledge-Based Systems

The expert systems are offshoots of artificial intelligence (AI) which is concerned with using computers to simulate human intelligence in a limited way. Artificial intelligence can be defined as the science of making machines do things that would require intelligence if done by humans. An expert system is defined as a software system (Figure 1) with a high symbolic and descriptive information content, which can simulate the performance of a human expert in a specific field or domain [2,3]. Expert systems acquire knowledge mostly from human experts. These systems deal with complex knowledge for which real expertise is required. Knowledge-Based Systems (KBS) are a general form of expert systems in view of the fact that they are quite general in application and not limited to mimicking the role of a human expert. Furthermore, very often they acquire knowledge from non-human information sources.

The major logical components of an expert system are a Knowledge Base, an Inference Engine, an Interface between the system and the external environment, an Explanation Facility, and a Knowledge Acquisition Facility.

A "knowledge engineer" gathers the expertise about a particular domain from one or more experts, and organizes that knowledge into the form required by the particular expert system tool that is to be used. Forms of knowledge representation may include logic, production system (rules), semantic scripts, semantic primitives, and frames and scripts. A typical expert system uses a data structure as the basis of the particular representation technique it implements. Consequently, the knowledge engineer needs to know the general form in which the knowledge is to be represented and the particular form of that representation required by the expert system itself. The knowledge that is "engineered" in this manner is the Knowledge Base [4].

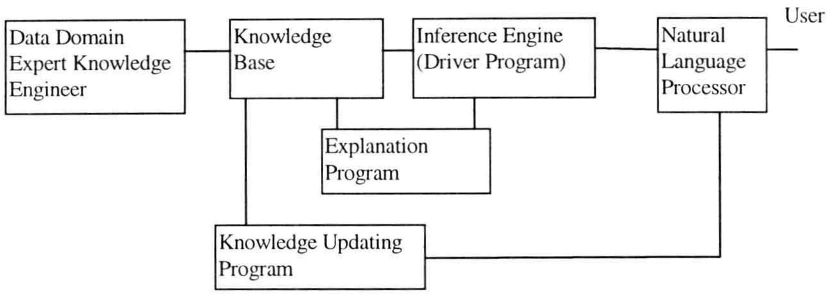


Figure 1: Simplified block diagram of a typical expert system.

The Inference Engine is the "driver" program. It traverses the Knowledge Base, in response to observations and other inputs provided to it from the external world, and possibly previous inferences and results from the KBS itself, and will identify one or more possible outcomes or conclusions [5]. This task of making inferences for arriving at solutions will involve "processing" of knowledge. It follows that representation and processing of knowledge are central to the functioning of a KBS [1]. The data structure selected for the specific form of knowledge representation determines the nature of the program created as an Inference Engine.

Keyboards, screen displays, sensors, transducers, and even output from other computer programs including expert systems, usually provide the Interface between an expert system and the external world.

In the case of production (rule-based) systems, the Knowledge Base consists of a set of rules written as an ASCII file. Therefore, the Knowledge Acquisition Facility that is required for these systems is often merely an editor [3]. In systems based on other forms of representation, the Knowledge Acquisition Facility will generally be an integral part of the expert system and can be used only with that system.

Typically, what is commercially available for developing expert systems is an expert system "shell." It will consist of the software programs required, but will not contain a knowledge base. It is the responsibility of the user, then, to organize the creation of the required knowledge base, which should satisfy the system's requirements with respect to the form of knowledge representation that is used and the structure of the knowledge base [2].

An expert system that is used to supervise the control system of a plant is called Control Expert System. Such expert systems are directly applicable at a high level, for process monitoring, supervision, and diagnosis, in intelligent control.

3 Neural Networks

3.1 Introduction

The successful operation of an autonomous machine depends on its ability to cope with a variety of unexpected and possibly unfamiliar events in its operating environment, perhaps relying on incomplete information [1]. Such an autonomous machine would only need to be presented a goal; it would achieve its objective through continuous interaction with its environment and automatic feedback about its response. In fact, this is an essential part of "learning." By enabling machines to possess such a level of autonomy, they would be able to learn higher-level cognitive tasks that are not easily handled by existing machines [6]. In designing a machine to emulate the capabilities found in biological controls, which depend on "intelligence." some experience on the structural, functional, and behavioral biological neural systems would be valuable. Neural networks have a great potential in the realm of nonlinear control problems [7]. A significant characteristic of neural networks is their ability to approximate arbitrary nonlinear functions. This ability of neural networks has made them useful in modeling nonlinear systems. A neuro-controller (neural network-based control system), in general, performs a specific task for adaptive control, with the controller taking the form of a multilayer network, and adaptable parameters being defined as the adjustable weights. In general, neural networks represent parallel-distributed processing structures, which make them prime candidates for use in multi-variable control systems. The neural-network approach defines the problem of control as the mapping of measured signals of system "change" into calculated "control actions," as shown in Figure 2.

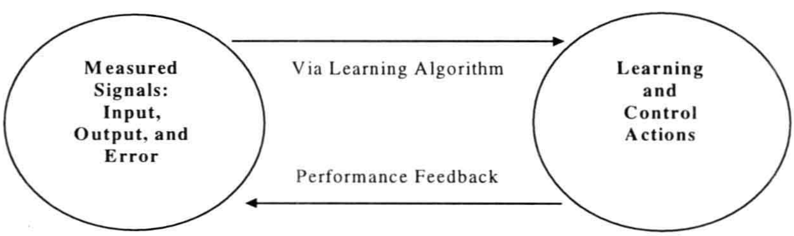


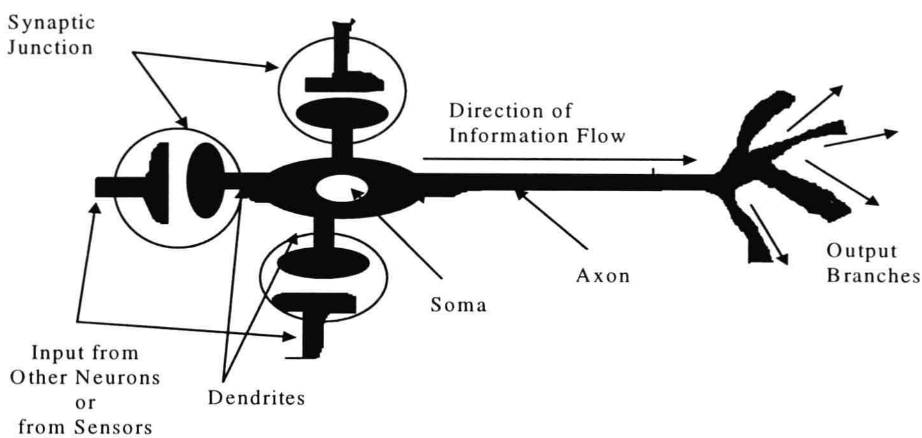
Figure 2: Representation of learning and control actions in a neural network approach. Mapping of the measured signals onto the learning and control space [8].

3.2 Biological Neuronal Morphology

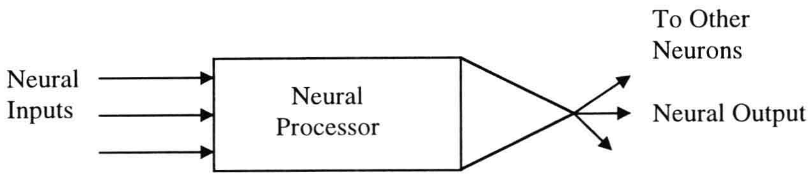
The human brain consists of approximately 10 billion individual nerve cells called neurons. Each neuron is interconnected to many other neurons, forming a densely connected network called neural network. These massive interconnections provide an exceptionally large computing power and memory. The basic building block of the central nervous system is the neuron, the cell that processes and communicates information to and from various parts of the body [8]. From an information-processing point of view, an individual neuron consists of the following three parts, each associated with a particular mathematical function:

(1) The dendrites are a receiving area for information from other neurons.
(2) The cell body, called soma, collects and combines incoming information from other neurons.
(3) A neuron transmits information to other neurons through a single fiber called an axon. The axon is a tubular structure bounded by a typical cell membrane.

The junction of an axon with a dendrite of another neuron is called a synapse. Synapses provide memory for accumulating experience or knowledge. A single axon may be involved with hundreds of other synaptic connections. A schematic diagram of the biological neuron is shown in Figure 3(a). From a system-theoretic point of view, the neuron can be considered a multiple-input—single-output (MISO) system, as depicted in Figure 3(b).



(a)



(b)

Figure 3: (a) A schematic view of the biological neuron.
(The soma of each neuron receives parallel inputs through its synapses and dendrites, and transmits a common output via the axon to other neurons.)
(b) Model representation of a biological neuron, with multiple inputs and a single output [8, 9].

Each neuron acts as a parallel processor because it receives pulses in parallel from neighboring neurons and then transmits pulses in parallel to all neighboring synapses [10]. The processing of information within the biological neuron involves two distinct operations [11,12]:

(1) Synaptic operation: This provides a weight to the neural inputs. Thus, the synaptic operation assigns a relative weight (significance) to each incoming signal according to the past experience (knowledge or memory) stored in the synapse.
(2) Somatic operation: This provides aggregation, thresholding, and nonlinear activation to the dendritic inputs. If the weighted aggregation of the neural inputs exceeds a certain threshold, the soma will produce an output signal.

3.3 Static Neural Networks

A neuron receives inputs from a number of other neurons or from the external world. A weighted sum of these inputs constitutes the argument of a nonlinear activation function as shown in Figure 4 [8]. The neuron is said to have been fired if the weighted sum of its inputs exceeds a certain threshold, w_0. Mathematically, the function of a neuron can be modeled as

$$y(t) = \Psi\left[\sum_{i=1}^{n} w_i x_i - w_0\right] \qquad (1)$$

where [$x_1, \ldots, x_n$] represent neuron inputs, [$w_1, \ldots, w_n$] are the synaptic weights, *y(t)* is the neural output, and $\Psi[.]$ is some nonlinear activation function with threshold w_0. Using this model, many neural morphologies, usually referred to as feedforward neural networks, have been reported in literature [8,11]. These feedforward networks respond instantaneously to inputs because they possess no dynamic elements in their structure. Therefore, these are called static neural networks. A schematic representation of a static neural network is shown in Figure 4(b).

The neural network attributes, such as learning from examples, generalization, redundancy, and fault tolerance, provide strong incentives for choosing neural networks as an appropriate approach for modeling biological systems. The potential benefits of such a network can be summarized as follows [8,11]:

(1) The neural network models have many neurons (the computational units) linked via adaptive (synaptic) weights, arranged in a massive parallel structure. Because of its high parallelism, failure of a few neurons does not significantly affect the overall performance (known as fault tolerance).
(2) The ability to adapt and learn from the environment means that the neural network models can deal with imprecise data and ill-defined situations. A suitably trained network can generalize; i.e., it can accurately deal with inputs that do not appear in the trained data.
(3) The ability to approximate any nonlinear continuous function to a desired degree of accuracy.
(4) Neural networks have many inputs and many outputs; hence, they are easily applicable to multivariable systems.
(5) With advances in hardware technology, many vendors have introduced dedicated VLSI hardware implementations of neural networks. This brings additional speed to neural computing.

3.4 Common Types of Artificial Neural Networks

The commonly used neural network structures [9,11] include feedback and feedforward networks.

The feedback networks are neural networks that have connections between network output and some or all other neuron units (see Figure 4(c)). Certain unit outputs in the figure are used as activated inputs to the network, and other unit outputs are used as network outputs. Due to the feedback, there is no guarantee that the networks become stable. To guarantee stability, constraints on synaptic weights are introduced so that the dynamics of the feedback network is expressed by a Lypunov function. Concretely, a constraint of equivalent mutual connection weights of two units is implemented. The Hopfield network is one such neural network.