SYMBOLIC COMPUTATION

Bruce D'Ambrosio

Qualitative Process Theory Using Linguistic Variables

Springer-Verlag

Bruce D'Ambrosio

Qualitative Process Theory Using Linguistic Variables

With 22 Figures

Springer-Verlag
New York Berlin Heidelberg
London Paris Tokyo Hong Kong

Bruce D'Ambrosio
Department of Computer Science
Oregon State University
Corvallis, OR 97331-4602

Library of Congress Cataloging-in-Publication Data
D'Ambrosio, Bruce.
Qualitative process theory using linguistic variables.
(Symbolic computation. Artificial intelligence)
Includes bibliographical references.
1. Artificial intelligence. 2. Reasoning.
3. Linguistics. I. Title. II. Series.
Q335.D36 1989 006.3 89-21698
ISBN 0-387-97135-1 (alk. paper)

Printed on acid-free paper

Camera-ready text prepared by the author using TEX.
Printed and bound by R. R. Donnelley & Sons, Harrisonburg, Virginia
Printed in the United States of America.

9 8 7 6 5 4 3 2 1

ISBN 0-387-97135-1 Springer-Verlag New York Berlin Heidelberg
ISBN 3-540-97135-1 Springer-Verlag Berlin Heidelberg New York

SYMBOLIC COMPUTATION

Artificial Intelligence

R.S. Michalski, J.G. Carbonell, T.M. Mitchell (Eds.): Machine Learning. An Artificial Intelligence Approach. XI, 572 pages, 1984

A. Bundy (Ed.): Catalogue of Artificial Intelligence Tools. Second, revised edition. IV, 168 pages, 1986

C. Blume, W. Jakob: Programming Languages for Industrial Robots. XIII, 376 pages, 145 figs., 1986

J.W. Lloyd: Foundations of Logic Programming. Second, extended edition. XII, 212 pages, 1987

L. Bolc (Ed.): Computational Models of Learning. IX, 208 pages, 34 figs., 1987

L. Bolc (Ed.): Natural Language Parsing Systems. XVIII, 367 pages, 151 figs., 1987

N. Cercone, G. McCalla (Eds.): The Knowledge Frontier. Essays in the Representation of Knowledge. XXXV, 512 pages, 93 figs., 1987

G. Rayna: REDUCE. Software for Algebraic Computation. IX, 329 pages, 1987

D.D. McDonald, L. Bolc (Eds.): Natural Language Generation Systems. XI, 389 pages, 84 figs., 1988

L. Bolc, M.J. Coombs (Eds.): Expert System Applications. IX, 471 pages, 84 figs., 1988

C.-H. Tzeng: A Theory of Heuristic Information in Game-Tree Search. X, 107 pages, 22 figs., 1988

L. Kanal, V. Kumar (Eds.): Search in Artificial Intelligence. X, 482 pages, 67 figs., 1988

H. Coelho, J. Cotta: Prolog by Example. 304 pages, 62 figs., 1988

H. Abramson, V. Dahl: Logic Grammars. XV, 258 pages, 40 figs., 1989

R. Hausser: Computation of Language. 425 pages, 1989

B. D'Ambrosio: Qualitative Process Theory Using Linguistic Variables. 156 pages, 22 figs., 1989

Contents

List of Figures

1

Overview

In this book we explore the use of linguistic variables as a semiquantitative extension to the qualitative value and relationship representations in qualitative process theory, for application in fuzzy logic control. Qualitative process (QP) theory, developed by Kenneth Forbus, describes the form of qualitative theories about the dynamics of physical systems. Its central thesis is that all change in such systems is the result of active processes, and that these processes should be explicitly represented and reasoned about. Much of QP theory's power derives from the qualitative representations used for the values of individual continuous-state parameters and the relations between these parameters. Qualitative descriptions are important because they provide the ability to reason with incomplete information and can guide the application of more detailed quantitative theories when additional information is available. Forbus has demonstrated that QP theory can be used to derive many significant deductions given only weak qualitative descriptions of variable values and relationships. For example, QP theory can be used to determine that the water in Fig. 1.1 will heat up and eventually boil, and that the container may eventually explode. However, there are at least three limitations to the current ability to analyze this situation using QP theory:

1. QP theory cannot be used to estimate how likely it is that there will be an explosion.

2. QP theory is unable to analyze situations only slightly more complicated than the one shown. For example, if we include a model for heat loss from the container to the surrounding environment, QP theory can no longer predict whether or not the water will boil.

3. QP theory provides little basis for reasoning about continuous control actions; for example, how much or when should the heat be turned down to avoid explosion?

We show that QP theory can be extended through the use of linguistic variables [Zad75a] to characterize both quantity magnitudes and necessary aspects of functional relationships. These extensions can reduce the ambiguity of QP analyses in terms of both the number of possible situations that may be occurring and the magnitude, time scale, etc., over which situations occur. These extensions reason at the appropriate level of detail for the kinds of questions typically asked in reasoning about the control of engineered systems; they are computationally tractable and can reason with

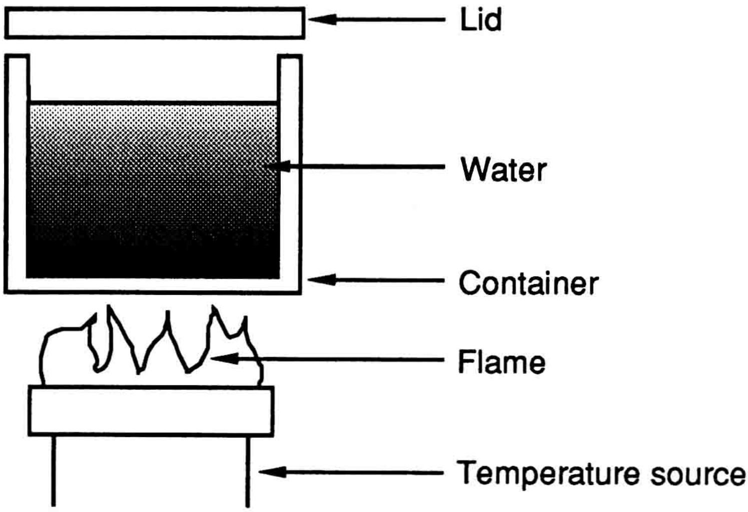

FIGURE 1.1. Boiling Example, after Forbus [For84].

the imprecise or uncertain data typically available to real-world control systems. These extensions are motivated by an examination of the potential use of QP theory in reasoning about the control of an engineered physical system, a chemical reaction furnace. Examination of the detailed model reveals a number of limitations similar to those listed. The source of these limitations is then traced to the restricted model of naive mathematics contained in QP theory. The proposed extensions to this mathematics are then described and demonstrated to be capable of eliminating many of the limitations at the price of requiring additional system-specific information about the system being modeled.

The three extensions presented in this book are based on the notion of a linguistic variable. First, an extension to the truth values and quantity ordering relations in QP theory enables representation of uncertain measurement data and estimates of state likelihoods. Second, an examination of qualitative characterizations of functional relationships reveals a number of possible extensions to the qualitative proportionality and influence relations of QP theory. One in particular, relationship strength, is shown to be capable of resolving some of the ambiguity that arises when attempting to apply QP theory to complex situations. Finally, an extension theory for the quantity representations used by QP theory, when combined with the other described extensions, provides estimates of the quantitative effects of adjustments to continuous control parameters. Much of the information required to improve the specificity of the results of applying QP theory is physical system, state, or even query specific. The work presented here is based on the hypothesis that problem solving does not proceed by choosing a single representation and manipulating it until a solution is found, but rather by choosing an initial representation, performing some initial problemsolving, "patching" the representation in response to problems encountered, and again resuming problem solving. This cycle may iterate several times before a satisfactory solution is reached. "Patches" are applied on the basis of the problems encountered and the query being asked, and are drawn from sources of information outside the theory being applied. We do not address the source of this external information here, but some initial explorations of this question are described in [D'A85].

Chapter 2 introduces the basic notions of fuzzy logic control. Chapter 3 provides a review of QP theory, including background and related research. Chapter 4 then presents an example, in which QP theory is used to analyze a simple continuous-flow industrial chemical system. Chapter 5 examines the results of this analysis, identifies certain limitations of QP theory in reasoning for process control, and analyzes the sources of those limitations. Chapter 5 ends with a description of an approach to addressing these limitations using linguistic variables. Chapter 6 provides some basic review material on linguistic variables, and presents some of the basic machinery needed to support the deductions used in later work. Chapter 7 introduces the concept of a linguistic quantity space, and show it integrates multiple representations of quantity information. Chapter 8 examines the problem of characterization of functional relationships. A number of different kinds of characterizations are identified, and one in particular, the notion of relationship strength, is shown to be capable of resolving one of the undecidable questions exposed in the example in chapters 4 and 5. Chapter 9 discusses the problem of estimating the effects of adjustments to continuous control parameters and presents linguistic perturbation analysis, a technique for producing these estimates for a wide class of situations. Finally, Chapter 10 summarizes the work presented here, compares it with related work, and outlines possibilities for further research.

2

Fuzzy Logic Control

2.1 Classical Control Theory

In this chapter we review previous results in the effort to incorporate approximate reasoning into the control of physical processes. In particular, we wish to review the research in fuzzy logic control and show how the research presented here is a logical extension of that work. Fuzzy control is an outgrowth of classical control theory, so we start with a brief review of classical control.

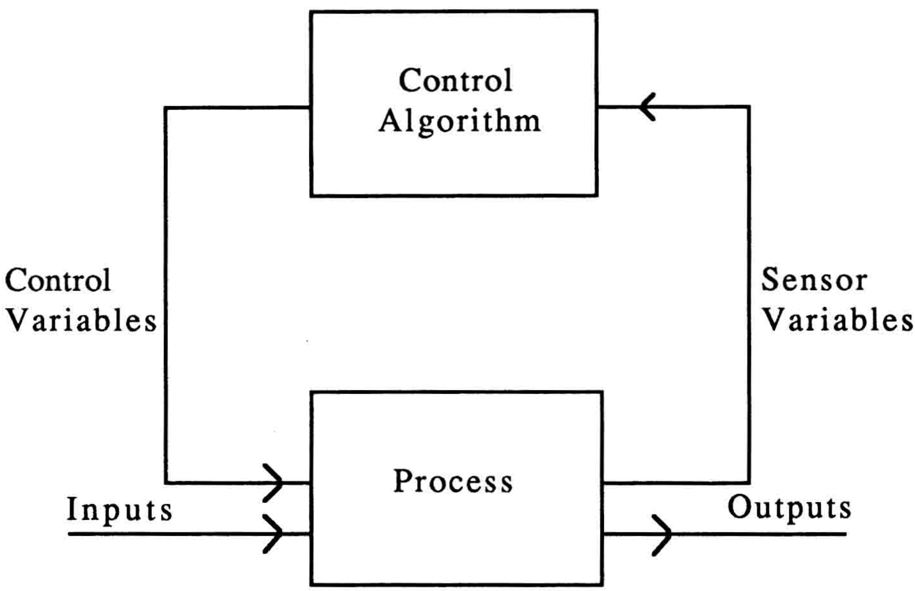

FIGURE 2.1. Classical Model of Process Control

As shown in Fig. 2.1, classical control theory is concerned with four basic components: the process to be controlled, a set of process parameters that can be observed (the sensor or control system input variables), a second set of process parameters that can be directly controlled (called the control or control system output variables), and a control algorithm that transforms sets of sensor variable observations into sets of control variable settings. The control algorithm is derived from a model of the process to be controlled. When the model is accurate and the control algorithm derivation performe(d correctly, the performance and stability of the control algorithm can be guaranteed within known limits. Control theory is well known and understood, and is the control mechanism of choice when applicable. However, in many situations, classical control theory may not be applicable for several reasons:

1. There may be no complete model of the process, or available models may be too complex or make unacceptable assumptions.
2. Control algorithms derived by classical techniques do not respond well to noise in sensor variable measurements [Mur85].
3. The stability and performance available from classical control algorithms may not be adequate for the requirements of the task at hand.

2.2 A New Approach to Control of Complex Systems

Fuzzy logic control was developed to overcome some of the problems cited above. Research on fuzzy control is an outgrowth of a landmark paper by Zadeh [Zad73] outlining a new approach to decision making in complex domains. This approach is based on the *principle of incompatibility*:

> ... as the complexity of a system increases, our ability to make precise yet significant statements about its behavior diminishes until a threshold is reached beyond which precision and significance (or relevance) become almost mutually exclusive characteristics.[1]

In this paper Zadeh introduced the *linguistic variable* to provide approximate descriptions of significant parameter values, the *fuzzy conditional statement* to provide for descriptions of simple relationships between linguistic variables, and the *fuzzy algorithm* to describe more complex relationships. Briefly, a linguisitic variable is a variable whose value is represented as a possibility distribution over a value space (for more details see Chapter 7). Fuzzy algorithms are constructed from fuzzy assignment and fuzzy conditional statements. Particularly relevant in the context of fuzzy control is the *fuzzy relational algorithm* introduced in [Zad73]. A fuzzy relational algorithm describes the relationship between fuzzy variables and is defined

[1] See [Zad73], p. 28.

as a set of conditional statements. Fuzzy relational algorithms can express the relationship between two or more variables, as well as the relationships between different aspects of the variables involved. Following is an example from [Zad73] describing the relationship between the value and first derivative of the variable x and the first derivative of the variable y:

Algorithm F(x, y):

1. **If** x is *small* and x is increased *slightly*, **then** y will increase *slightly*.
2. **If** x is *small* and x is increased *substantially*, **then** y will increase *substantially*.
3. **If** x is *large* and x is increased *slightly*, **then** y will increase *moderately*.
4. **If** x is *large* and x is increased *substantially*, **then** y will increase *very substantially*.

In the case in which multiple rules apply, each serves to restrict the possible value of the consequent, and therefore Zadeh recommends that the proper combining rule is to intersect the results.

2.3 Fuzzy Control

Mamdani [MA75], [Mam76] was the first to apply the ideas presented in Zadeh's paper to the problem of process control. Mamdani encountered the problem that classical control theory was often inapplicable, usually because of lack of a suitable formal model of the process. However, there was available a large body of vague, nonnumeric information about how to control the process of interest, a simple steam engine in his first study. Mamdani found that fuzzy relational algorithms provided a direct representation for this knowledge, and that the defined semantics of fuzzy relational algorithms, when applied to these representations, determined control actions that corresponded with the control actions performed by skilled operators. The new model of control proposed by Mamdani is shown in Fig. 2.2 and is considerably different from the standard model of control.[2]

The process, sensor parameters, and control parameters are as before. Sensor readings, which are assumed to be nonfuzzy, must be converted to fuzzy set form, and this occurs in a process called *fuzzification*. The definitions of the fuzzy predicates are stored in a *database* associated with the controller. The linguistic control knowledge is stored as a set of fuzzy conditional statements in a *rule-base* and can be seen as specifying a fuzzy

[2] Figure adapted from [Ton84].