# **Process Control**

## A First Course with MATLAB

Pao C. Chau

University of California, San Diego



PUBLISHED BY THE PRESS SYNDICATE OF THE UNIVERSITY OF CAMBRIDGE The Pitt Building, Trumpington Street, Cambridge, United Kingdom

CAMBRIDGE UNIVERSITY PRESS The Edinburgh Building, Cambridge CB2 2RU, UK 40 West 20th Street, New York, NY 10011-4211, USA 477 Williamstown Road, Port Melbourne, VIC 3207, Australia Ruiz de Alarcón 13, 28014 Madrid, Spain Dock House, The Waterfront, Cape Town 8001, South Africa

http://www.cambridge.org

© Cambridge University Press 2002

This book is in copyright. Subject to statutory exception and to the provisions of relevant collective licensing agreements, no reproduction of any part may take place without the written permission of Cambridge University Press.

First published 2002

Printed in the United States of America

Typeface Times Roman 10.25/12.5 pt. System  $LAT_EX 2_{\mathcal{E}}$  [TB]

A catalog record for this book is available from the British Library.

Library of Congress Cataloging in Publication Data Chau. Pao C.

Process control : a first course with MATLAB / Pao C. Chau.
p. cm. – (Cambridge series in chemical engineering)
Includes bibliographical references and index.
ISBN 0-521-80760-3 – ISBN 0-521-00255-9 (pb.)
1. Chemical process control – Data processing. 2. MATLAB. I. Title. II. Series.
TP155.75 .C42 2002
660'.2815 – dc21 2001052567
ISBN 0 521 80760 3 hardback
ISBN 0 521 00255 9 paperback

MATLAB is a registered trademark of The MathWorks, Inc.

## Contents

	Preface	<i>page</i> xi
I	Introduction	1
2	Mathematical Preliminaries	6
	2.1. A Simple Differential Equation Model	7
	2.2. Laplace Transform	8
	2.3. Laplace Transforms Common to Control Problems	12
	2.4. Initial- and Final-Value Theorems	15
	2.5. Partial-Fraction Expansion	16
	2.6. Transfer Function, Pole, and Zero	21
	2.7. Summary of Pole Characteristics	24
	2.8. Two Transient Model Examples	26
	2.9. Linearization of Nonlinear Equations	33
	2.10. Block-Diagram Reduction	37
	Review Problems	40
3	Dynamic Response	44
	3.1. First-Order Differential Equation Models	45
	3.2. Second-Order Differential Equation Models	48
	3.3. Processes with Dead Time	52
	3.4. Higher-Order Processes and Approximations	53
	3.5. Effect of Zeros in Time Response	58
	Review Problems	61
4	State-Space Representation	64
	4.1. State-Space Models	64
	4.2. Relation of State-Space Models to Transfer Function Mod	dels 71
	4.3. Properties of State-Space Models	78
	Review Problems	81
5	Analysis of Single-Loop Control Systems	83
	5.1. PID controllers	83
	5.2. Closed-Loop Transfer Functions	90

### Contents

<ul><li>5.3. Closed-Loop System Response</li><li>5.4. Selection and Action of Controllers Review Problems</li></ul>	95 102 105
<ul> <li>Design and Tuning of Single-Loop Control Systems</li> <li>6.1. Tuning Controllers with Empirical Relations</li> <li>6.2. Direct Synthesis and Internal Model Control Review Problems</li> </ul>	108 108 115 127
<ul> <li>Stability of Closed-Loop Systems</li> <li>7.1. Definition of Stability</li> <li>7.2. The Routh–Hurwitz Criterion</li> <li>7.3. Direct-Substitution Analysis</li> <li>7.4. Root-Locus Analysis</li> <li>7.5. Root-Locus Design</li> <li>7.6. Final Remark on Root-Locus Plots Review Problems</li> </ul>	129 129 130 135 137 143 145 146
<ul> <li>Frequency-Response Analysis</li> <li>8.1. Magnitude and Phase Lag</li> <li>8.2. Graphical Analysis Tools</li> <li>8.3. Stability Analysis</li> <li>8.4. Controller Design Review Problems</li> </ul>	147 147 152 161 168 176
<ul> <li>Design of State-Space Systems</li> <li>9.1. Controllability and Observability</li> <li>9.2. Pole-Placement Design</li> <li>9.3. State Estimation Design Review Problems</li> </ul>	178 178 182 188 194
Multiloop Systems 10.1. Cascade Control 10.2. Feedforward Control 10.3. Feedforward–Feedback Control 10.4. Ratio Control 10.5. Time-Delay Compensation – The Smith Predictor 10.6. Multiple-Input Multiple-Output Control 10.7. Decoupling of Interacting Systems Review Problems	198 198 203 206 207 209 210 217 221
MATLAB Tutorial Sessions Session 1. Important Basic Functions Session 2. Partial-Fraction and Transfer Functions Session 3. Time-Response Simulation Session 4. State-Space Functions Session 5. Feedback Simulation Functions Session 6. Root-Locus Functions	226 226 234 238 243 249 254 254
	<ul> <li>5.3. Closed-Loop System Response</li> <li>5.4. Selection and Action of Controllers Review Problems</li> <li>Design and Tuning of Single-Loop Control Systems</li> <li>5.1. Tuning Controllers with Empirical Relations</li> <li>5.2. Direct Synthesis and Internal Model Control Review Problems</li> <li>Stability of Closed-Loop Systems</li> <li>7.1. Definition of Stability</li> <li>7.2. The Routh-Hurwitz Criterion</li> <li>7.3. Direct-Substitution Analysis</li> <li>7.4. Root-Locus Analysis</li> <li>7.5. Root-Locus Design</li> <li>7.6. Final Remark on Root-Locus Plots Review Problems</li> <li>5.7. Root-Locus Design</li> <li>7.6. Final Remark on Root-Locus Plots Review Problems</li> <li>5.8. Controller Design</li> <li>7.9. Stability Analysis</li> <li>7.9. Controller Design</li> <li>7.9. State Estimation Design Review Problems</li> <li>Design of State -Space Systems</li> <li>9.1. Controllability and Observability</li> <li>9.2. Pole-Placement Design Review Problems</li> <li>9.3. State Estimation Design Review Problems</li> <li>9.4. State Control</li> <li>10.3. Feedforward-Feedback Control</li> <li>10.3. Feedforward-Feedback Control</li> <li>10.4. Ratic Control</li> <li>10.5. Time-Delay Compensation – The Smith Predictor</li> <li>10.6. Multiple-Input Multiple-Output Control</li> <li>10.7. Decoupling of Interacting Systems Review Problems</li> <li>MATLAB Tutorial Sessions</li> <li>Session 1. Important Basic Functions</li> <li>Session 2. Partial-Fraction and Transfer Functions</li> <li>Session 3. Time-Response Simulation</li> <li>Session 4. State-Space Functions</li> <li>Session 5. Feedback Simulation Functions</li> <li>Session 5. Feedback Simulation Functions</li> <li>Session 6. Root-Locus Functions</li> <li>Session 7. Frequency-Response Functions</li> </ul>

Contents

Homew	267	
Part I.	Basic Problems	267
Part II.	Intermediate Problems	275
Part III.	Extensive Integrated Problems	290
Referen	aces	305
Index	307	

ontrol systems are tightly intertwined in our daily lives so much so that we take them for granted. They may be as low tech and unglamorous as our flush toilet. Or they may be as high tech as electronic fuel injection in our cars. In fact, there is more than a handful of computer control systems in a typical car that we now drive. In everything from the engine to transmission, shock absorber, brakes, pollutant emission, temperature, and so forth, there is an embedded microprocessor controller keeping an eye out for us. The more gadgetry, the more tiny controllers pulling the trick behind our backs.<sup>1</sup> At the lower end of consumer electronic devices, we can bet on finding at least one embedded microcontroller.

In the processing industry, controllers play a crucial role in keeping our plants running – virtually everything from simply filling up a storage tank to complex separation processes and chemical reactors.

As an illustration, let's take a look at a bioreactor (Fig. 1.1). To find out if the bioreactor is operating properly, we monitor variables such as temperature, pH, dissolved oxygen, liquid level, feed flow rate, and the rotation speed of the impeller. In some operations, we may also measure the biomass and the concentration of a specific chemical component in the liquid or the composition of the gas effluent. In addition, we may need to monitor the foam head and make sure it does not become too high. We most likely need to monitor the steam flow and pressure during the sterilization cycles. We should note that the schematic diagram is far from complete. By the time we have added enough details to implement all the controls, we may not recognize the bioreactor. These features are not pointed out to scare anyone; on the other hand, this is what makes control such a stimulating and challenging field.

For each quantity that we want to maintain at some value, we need to ensure that the bioreactor is operating at the desired conditions. Let's use the pH as an example. In control calculations, we commonly use a *block diagram* to represent the problem (Fig. 1.2). We will learn how to use mathematics to describe each of the blocks. For now, the focus is on some common terminology.

To consider pH as a *controlled variable*, we use a pH electrode to measure its value and, with a transmitter, send the signal to a controller, which can be a little black box or a computer. The controller takes in the pH value and compares it with the desired pH, what

<sup>&</sup>lt;sup>1</sup> In the 1999 Mercedes-Benz S-class sedan, there are approximately 40 "electronic control units" that control up to 170 different variables.



**Figure 1.1.** Schematic diagram of instrumentation associated with a fermentor. The steam sterilization system and all sensors and transmitters are omitted for clarity. The thick solid lines represent process streams. The thin solid lines represent information flow.

is called the *set point* or the *reference*. If the values are not the same, there is an *error*, and the controller makes proper adjustments by manipulating the acid or the base pump – the *actuator*.<sup>2</sup> The adjustment is based on calculations made with a *control algorithm*, also called the control law. The error is calculated at the summing point, where we take the desired pH minus the measured pH. Because of how we calculate the error, this is a *negative-feedback* mechanism.

This simple pH control example is what we call a *single-input single-output* (SISO) system; the single input is the set point and the output is the pH value.<sup>3</sup> This simple feedback



**Figure 1.2.** Block-diagram representation of a single-input singleoutput negative-feedback system. Labels within the boxes are general. Labels outside the boxes apply to the simplified pH control discussion.

- <sup>2</sup> In real life, bioreactors actually use on-off control for pH.
- <sup>3</sup> We will learn how to identify input and output variables, how to distinguish among manipulated variables, disturbances, measured variables, and so forth. Do not worry about remembering all the terms here; they will be introduced properly in subsequent chapters.

mechanism is also what we call a *closed loop*. This single-loop system ignores the fact that the dynamics of the bioreactor involves complex interactions among different variables. If we want to take a more comprehensive view, we need to design a *multiple-input multiple-output* (MIMO), or *multivariable*, system. When we invoke the term *system*, we are referring to the *process*<sup>4</sup> (the bioreactor here), the *controller*, and all other instrumentation, such as *sensors*, *transmitters*, and *actuators* (like valves and pumps) that enable us to control the pH.

When we change a specific operating condition, meaning the set point, we would like, for example, the pH of the bioreactor to follow our command. This is what we call *servocontrol*. The pH value of the bioreactor is subjected to external *disturbances* (also called *load changes*), and the task of suppressing or rejecting the effects of disturbances is called *regulatory control*. Implementation of a controller may lead to instability, and the issue of system *stability* is a major concern. The control system also has to be *robust* such that it is not overly sensitive to changes in process parameters.

What are some of the issues when we design a control system? In the first place, we need to identify the role of various variables. We need to determine what we need to control, what we need to manipulate, what the sources of disturbances are, and so forth. We then need to state our design objective and specifications. It may make a difference whether we focus on the servo or on the regulator problem, and we certainly want to make clear, quantitatively, the desired response of the system. To achieve these goals, we have to select the proper control strategy and controller. To implement the strategy, we also need to select the proper sensors, transmitters, and actuators. After all is done, we have to know how to tune the controller. Sounds like we are working with a musical instrument, but that's the jargon.

The design procedures depend heavily on the dynamic model of the process to be controlled. In more advanced model-based control systems, the action taken by the controller actually depends on the model. Under circumstances for which we do not have a precise model, we perform our analysis with approximate models. This is the basis of a field called *system identification and parameter estimation*. Physical insight that we may acquire in the act of model building is invaluable in problem solving.

Although we laud the virtue of dynamic modeling, we will not duplicate the introduction of basic conservation equations. It is important to recognize that all of the processes that we want to control, e.g., bioreactor, distillation column, flow rate in a pipe, drug delivery system, etc., are what we have learned in other engineering classes. The so-called model equations are conservation equations in heat, mass, and momentum. We need force balance in mechanical devices, and, in electrical engineering, we consider circuit analysis. The difference between what we now use in control and what we are more accustomed to is that control problems are *transient* in nature. Accordingly, we include the time-derivative (also called accumulation) term in our balance (model) equations.

What are some of the mathematical tools that we use? In *classical* control, our analysis is based on linear ordinary differential equations with constant coefficients – what is called *linear time invariant* (LTI). Our models are also called *lumped-parameter* models, meaning that variations in space or location are not considered. Time is the only independent variable. Otherwise, we would need partial differential equations in what is called *distributed-parameter* models. To handle our linear differential equations, we rely heavily

<sup>&</sup>lt;sup>4</sup> In most of the control world, a process is referred to as a *plant*. Here "process" is used because, in the process industry, a plant carries the connotation of the entire manufacturing or processing facility.

Example	Page no.
Example 4.7	72
Example 4.7A	73
Example 4.7B	186
Example 4.7C	187
Example 4.8	75
Example 4.8A	181
Example 5.7	101
Example 5.7A	112
Example 5.7B	123
Example 5.7C	123
Example 5.7D	172
Example 7.2	133
Example 7.2A	135
Example 7.2B	139
Example 7.2C	170
Example 7.2D	171
Example 7.3	133
Example 7.3A	135
Example 7.3B	140
Example 7.4	136
Example 7.4A	172
Example 7.5	137
Example 7.5A	143
Example 7.5B	188

Table 1.1. Examples used in different chapters

on *Laplace transform*, and we invariably rearrange the resulting algebraic equation into the so-called *transfer functions*. These algebraic relations are presented graphically as block diagrams (as in Fig. 1.2). However, we rarely go as far as solving for the time-domain solutions. Much of our analysis is based on our understanding of the roots of the characteristic polynomial of the differential equation – what we call the *poles*.

At this point, a little secret should be disclosed. Just from the terminology, it may be inferred that control analysis involves quite a bit of mathematics, especially when we go over stability and frequency-response methods. That is one reason why these topics are not immediately introduced. Nonetheless, we have to accept the prospect of working with mathematics. It would be a lie to say that one can be good in process control without sound mathematical skills.

Starting in Chap. 6, a select set of examples is repeated in some subsections and chapters. To reinforce the thinking that different techniques can be used to solve the same problem, these examples retain the same numeric labeling. These examples, which do not follow conventional numbering, are listed in Table 1.1 to help you find them.

It may be useful to point out a few topics that go beyond a first course in control. With certain processes, we cannot take data continuously, but rather in certain selected slow intervals (e.g., titration in freshmen chemistry). These are called *sampled-data* systems. With

computers, the analysis evolves into a new area of its own – *discrete-time* or *digital* control systems. Here, differential equations and Laplace transform do not work anymore. The mathematical techniques to handle discrete-time systems are difference equations and *z* transforms. Furthermore, there are multivariable and state-space controls, which we will encounter in a brief introduction. Beyond the introductory level are optimal control, non-linear control, adaptive control, stochastic control, and fuzzy-logic control. Do not lose the perspective that control is an immense field. Classical control appears insignificant, but we have to start somewhere, and onward we crawl.