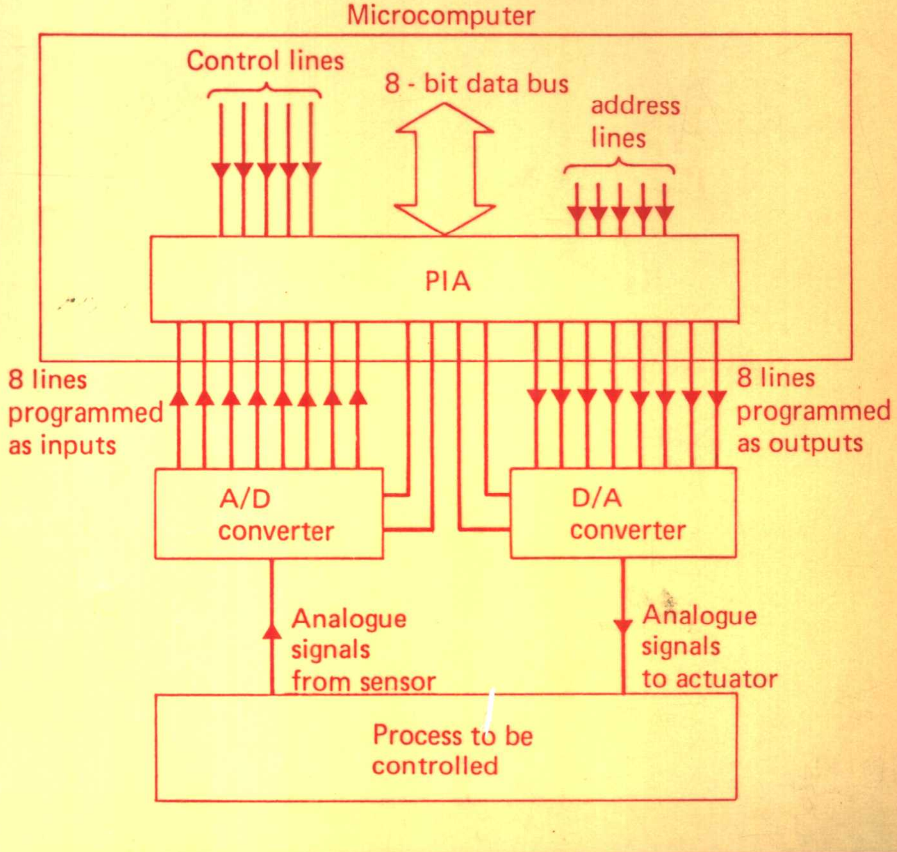
J. R. Leigh

APPLIED DIGITAL CONTROL



Theory, Design and Implementation

APPLIED DIGITAL CONTROL

Theory, Design and Implementation

J. R. LEIGH

Professor of Control Engineering
The Polytechnic of Central London

Prentice/Hall International

ENGLEWOOD CLIFFS, NEW JERSEY LONDON MEXICO NEW DELHI
RIO DE JANEIRO SINGAPORE SYDNEY TOKYO TORONTO WELLINGTON

Library of Congress Cataloging in Publication Data

Leigh, James R.
Applied digital control

Bibliography: p.
Includes index.
1. Digital control systems. I. Title.
TJ216.L45 1984 621.8'95 84-8253
ISBN 0-13-040189-7
ISBN 0-13-040171-4 (pbk.)

British Library Cataloguing in Publication Data

Leigh, J. R.
Applied digital control.
1. Digital control systems
2. Microprocessors
I. Title
629.8'95 TJ216

ISBN 0-13-040189-7
ISBN 0-13-040171-4 Pbk

ISBN 0-13-040189-7
ISBN 0-13-040171-4 PBK

PRENTICE-HALL, INC., *Englewood Cliffs New Jersey*
PRENTICE-HALL INTERNATIONAL, UK., LTD., *London*
PRENTICE-HALL OF AUSTRALIA PTY., LTD., *Sydney*
PRENTICE-HALL CANADA, INC., *Toronto*
PRENTICE-HALL OF INDIA PRIVATE LIMITED, *New Delhi*
PRENTICE-HALL OF JAPAN, INC., *Tokyo*
PRENTICE-HALL OF SOUTHEAST ASIA PTE., LTD., *Singapore*
EDITORA PRENTICE-HALL DO BRASIL LTDA., *Rio de Janeiro*
PRENTICE-HALL HISPANOAMERICANA, S.A., *Mexico*
WHITEHALL BOOKS LIMITED, *Wellington, New Zealand*

Printed in Great Britain by A. Wheaton & Co., Ltd., Exeter

10 9 8 7 6 5 4 3 2

PREFACE

Control engineering becomes ever more inextricably linked with developments in microprocessor technology. The educational implications are that control and computational aspects should be closely integrated to ensure that a student can move fluently from control design to computer implementation.

This book aims to establish a strong theoretical background to support applications material relevant to the design and implementation of digital control systems.

The material presented here has been developed from lectures given to the third year of a B.Sc. Degree Course in Control and Computer Engineering and an M.Sc. Degree in Instrumentation and Digital Systems at the Polytechnic of Central London. The strong applications theme running throughout the book rests on the author's close involvement in design of practical control systems for industry.

It is hoped that the book will be found useful and stimulating by industrial engineers and scientists as well as by students on formal courses.

Prerequisites for the book are a knowledge of mathematics, computation and classical control theory such as is possessed by most engineering undergraduates at the start of their final year. The material can be covered in a lecture time of 30 hours with, preferably, an approximately equal time allocation for supporting laboratory sessions, computer workshops and seminars.

I am pleased to acknowledge the assistance of computer manufacturers in supplying and agreeing to the publication of the material on which I have based Chapters 10 and 11. I am particularly indebted to Alain Zucho and Christian Gillet, both of Honeywell S.A. (Brussels).

I thank my colleagues, Messrs Martinho and Winter for allowing me to use some of their problems as exercises for the book. I also thank staff of the Polytechnic's Library and Computer Centre for assistance in compiling the references and bibliography. I am grateful to Alan Whittle for his help in preparing the manuscript for press.

Finally, it is my pleasure to thank Miss Sharon O'Keeffe for typing part of the first version of the manuscript, and the staff of Prentice-Hall for their help in conversion of the manuscript into the finished product.

THE STRUCTURE OF THE BOOK

Chapter 1 briefly provides motivation and outlines some of the problems with which the book is concerned.

Chapters 2–7 constitute a complete course in the theory and design of individual digital control loops. Chapter 2 surveys concisely the theory of discrete time signals as required by later chapters. Chapter 3 establishes fluency in the use of the Z transform. Chapter 4 establishes a toolkit of design-oriented techniques. Chapter 5 is the nucleus of the first part of the book. It puts forward alternative design methods and follows these with practical design considerations such as the choice of sampling interval and word length. Chapter 6 considers the (analogue and digital) hardware of the control loop with emphasis on dynamic response and cost-effective choice of elements. Chapter 7 illustrates by case histories some of the methods advocated in earlier chapters.

Chapters 8–11 constitute a concise and self-contained course in the theory and design of multi-loop digital systems. Chapter 8 is the nucleus of the second half of the book. It establishes theory and design techniques for multivariable systems with coverage of continuous as well as discrete systems. Chapter 9 introduces recent theoretical work on the control of large complex systems. It explains various approaches to the subdivision and overall control of systems composed of many individual elements and with long- and short-term control objectives. Chapter 10 examines and rationalizes the control of large systems from a practical industrial viewpoint. The chapter contains original suggestions that may be used for characterizing distributed computer networks and matching them to a particular given industrial control requirement. Chapter 11 follows on the material of Chapter 10 by reviewing typical commercially available distributed control systems to demonstrate how manufacturers are meeting the objectives described in Chapters 9 and 10. Chapter 11 is completed by four case histories.

Appendix A gives a detailed description of a peripheral chip as used for implementation in a control loop.

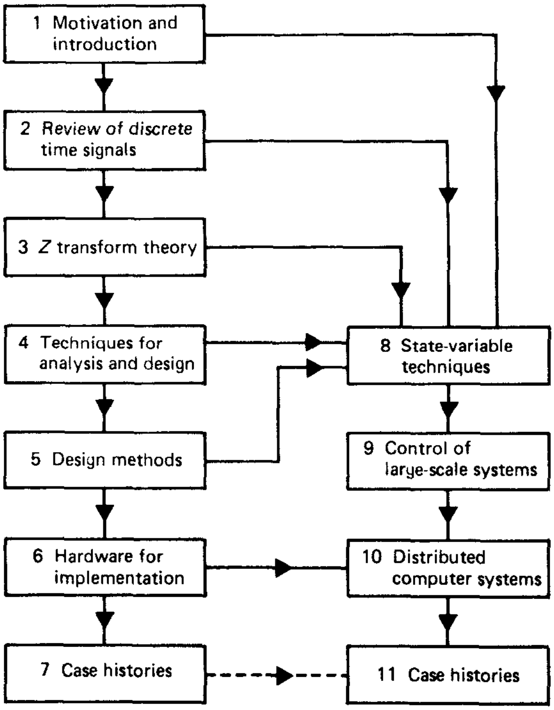
Appendix B lists transform pairs relating time functions with Z transforms and Laplace transforms with Z transforms.

The References/Bibliography list is preceded by an introductory section to assist the reader in selecting further reading.

The block diagram shows the main features of the book and their interrelation.

Teacher's Manual

A Teacher's Manual containing detailed solutions to the exercises is available from the publisher to those teachers who adopt the book as their classroom text.



Structure of the book

CONTENTS

1 INTRODUCTION

1.1 MICROPROCESSOR BASICS

1.1.1 Main Features

A microprocessor is a logic circuit that moves through a sequence of states as dictated by stored instructions (the computer program) and at a rate dictated by pulses from a clock. It contains a number of storage locations that it uses for holding current data, current instructions, the address of any peripheral device with which it is due to communicate and the address of the next line of the program.

Any particular task is accomplished by completing a number of cycles of data transfer between the microprocessor, its memories and its peripheral devices. Each cycle can be timed with reasonable accuracy and so the time required for any particular task can be estimated.

Random Access Memory (RAM) is used as general-purpose working space during computation and data transfer. It is volatile, in the sense that the information that it contains is lost when the electrical supply is removed.

Read Only Memory (ROM), Programmable Read Only Memory (PROM), and Erasable Programmable Read Only Memory (EPROM) all serve rather similar purposes. They are rapid sources of information of the type that never (ROM, PROM) or infrequently (EPROM) need to be modified. Examples are: Pascal computer software, paper tape conversion look-up tables, standard control algorithms.

Communication between the microprocessor, the RAM and ROM memories and the interface leading on to the peripheral devices is by means of a time-shared bidirectional data bus. For instance, to transfer a particular piece of data from ROM to the microprocessor, first the relevant address is sent to the ROM and then the contents of the addressed byte are transferred on to the data bus. The actual data transfers are achieved through tristates—they act as closed switches when enabled (i.e. energized) and as open switches otherwise.

1.1.2 Interrupt Mechanism

The interrupt mechanism makes it possible for the system to jump from the program it is currently executing to undertake a task that has a higher priority. The mechanism works as follows. When an event occurs that necessitates a jump from the current program, a signal is sent to the interrupt interface of the computer. This interface raises a flag. The microprocessor, on detecting the flag, then checks to see which peripheral device is requesting service. The current program is interrupted, the request satisfied and then the interrupted program is restored and restarted.

(A flag is a signal that indicates that one device wishes to communicate with another.)

1.1.3 Monitor Program

Programs are developed, edited, tested and started with the aid of a monitor program. The user's access to the microprocessor is essentially through this program.

1.1.4 Programming in Assembly Code

It is necessary to learn the mnemonic code peculiar to the machine that is to be used. Program listings are rather long, since objectives have to be achieved by transferring data between registers, decrementing the contents of registers and making conditional jumps amongst the stored instructions. Programming efficiency can be improved by using either subroutines or macros for recurring tasks. A subroutine is written once only and is accessed wherever it is required. The organizational problem is to ensure that the program returns to the correct point on completion of the subroutine. This is achieved using a first-in last-out memory called a stack.

Macros are simply blocks of program that are inserted wherever they are required. They do not need to be written out in full every time—this is done by the cross-assembler on recognizing an appropriate key word in the assembly-code listing.

1.1.5 Programming in High-Level Languages

The attractions of programming in a high-level language are obvious. Concise programs at a level of detail (ideally) corresponding with the natural thought processes of a design engineer can be written, tested and edited with ease.

The two drawbacks of high-level languages are inefficiency and inflexibility. In general a high-level language program is not so efficient as a reasonably well-written machine code counterpart. However, the high cost of a programmer's time and the decreasing cost of computer power mean that high-level languages are increasingly attractive. Not every task of which a computer is capable can be achieved using a high-level language, and therefore it remains necessary to write machine-code programs (or blocks of machine code for insertion into high-level programs) for special tasks.

For real-time working, the usual high-level instructions are typically augmented by PEEK, which can read the register of an input port and POKE, which can write to the register of an output port. The other facility that is needed is a timer.

1.1.6 Communication Between Keyboard/Display and Computer

Communication between keyboard and computer is almost always by serial transmission using standardized electrical protocols. At the computer, the data are converted into parallel form in a serial–parallel converter. Seven bits are sufficient to represent any character on a standard keyboard. The computer therefore represents characters received or characters to be sent to display by a seven-bit word.

1.1.7 Programmable Interface Chips

There is an increasing tendency to use intelligent interface chips that can be programmed for a particular data transfer task, which they then carry out with little dependence on the system CPU.

1.1.8 Discussion

In the first part of the book, microcomputers are considered as elements in the control loop. Later, computer networks are considered as supervisory tools for the control and coordination of industrial complexes.

1.2 A DIGITAL COMPUTER IN THE CONTROL LOOP

Figure 1.1 shows a fairly standard analogue (continuous time) speed-control system in which an electric motor is controlled by a feedback loop to follow the reference signal $v(t)$.

Figure 1.2 shows the same speed-control system with a digital computer included in the loop. Notice how the computer is necessarily preceded by an analogue-to-digital converter (A/D) and followed by a digital-to-analogue converter (D/A).