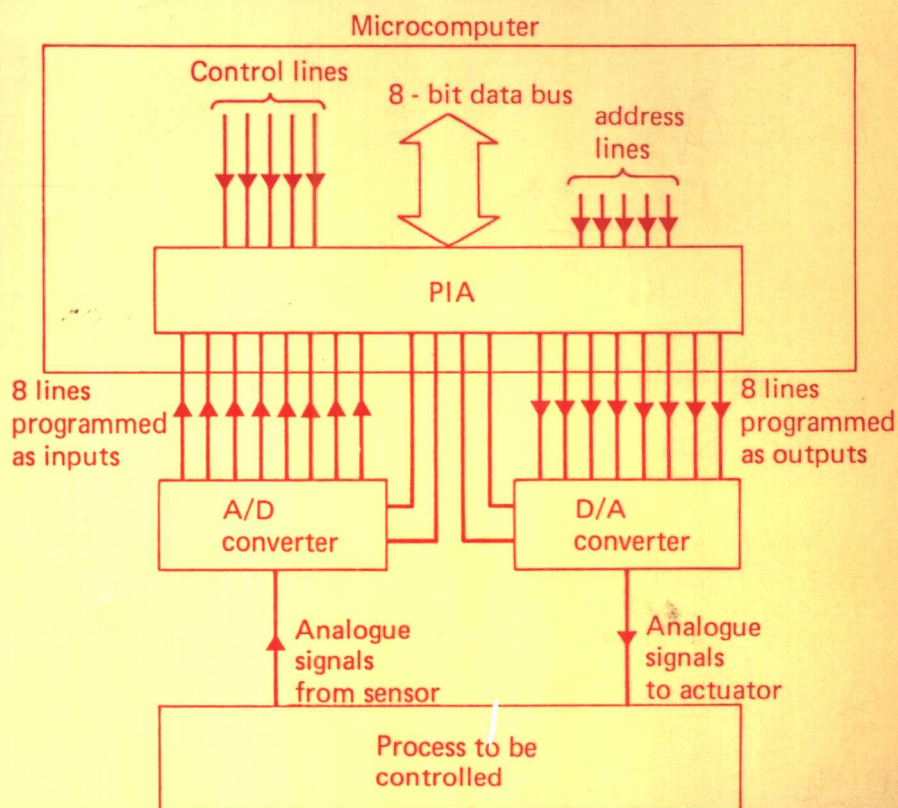


J. R. Leigh

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*



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

© 1985 by PRENTICE-HALL INTERNATIONAL, UK., LTD

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of Prentice-Hall International, UK., Ltd. For permission within the United States contact Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632.

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.

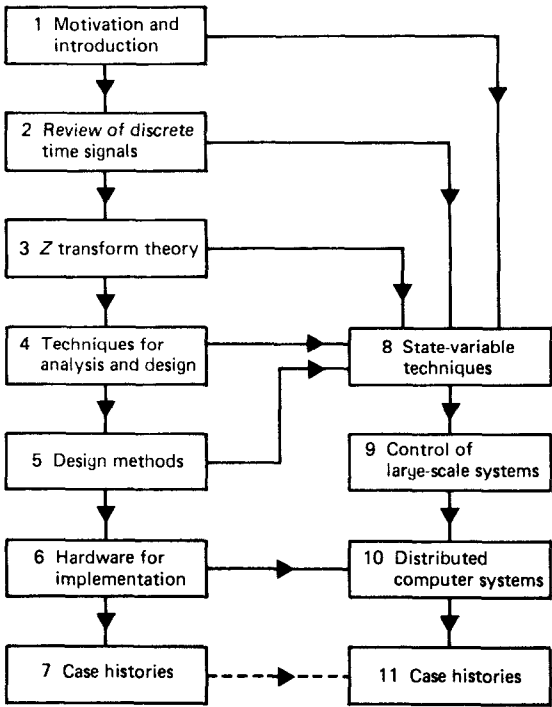
Appendix B lists transform pairs relating time functions with Z transforms and Laplace transforms with Z transforms.

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

Preface

xv

1	Introduction	1
1.1	Microprocessor basics	1
1.1.1	Main features	1
1.1.2	Interrupt mechanism	2
1.1.3	Monitor program	2
1.1.4	Programming in assembly code	2
1.1.5	Programming in high-level languages	2
1.1.6	Communication between keyboard/display and computer	3
1.1.7	Programmable interface chips	3
1.1.8	Discussion	3
1.2	A digital computer in the control loop	3
1.3	Control of interlinked processes	6
1.4	Summary	7
2	Discrete-time signals; idealized approximation; frequency spectrum; reconstruction	8
2.1	Introduction	8
2.2	The nature of sampled signals	8
2.3	Loss of information in a sampled signal	10
2.4	Creation of spurious information in a sampled signal	10
2.5	The sampling process	10
2.6	Ideal sampling	14
2.7	Digital control of a single-input/single-output process	16
2.8	Reconstruction of the signal $f(t)$ from the discrete-time signal $f^*(t)$	18
2.9	Approximate reconstruction of $f(t)$ from $f^*(t)$ using hold devices	19
2.10	An approach to the analysis of a control loop containing a digital controller	21
2.11	Other methods of sampling	22
	Exercises	22

3	Z transform techniques	24
3.1	Introduction	24
3.2	The Z transform	24
3.2.1	The Z transform of a time function	25
3.2.2	The Z transform corresponding with a particular Laplace transform	26
3.2.3	Considering z as an operator	28
3.2.4	Properties of the Z transform	29
3.3	The pulse transfer function	32
3.3.1	Manipulation of block diagrams	33
3.4	A brief description of three methods for obtaining information on system behavior between sampling instants	36
3.5	Inverse Z transformation to yield time solutions	40
3.5.1	Inversion by means of the method of residues	40
3.5.2	Transform inversion by partial fraction expansion followed by recognition of inverse transforms in tables	42
3.5.3	Transform inversion by long division	43
3.6	Difference equations	43
3.6.1	Introduction	43
3.6.2	Solution by means of the Z transform	44
3.6.3	Numerical solution of difference equations	46
3.6.4	Computer realization of difference equations by means of delay blocks	46
3.7	The z plane and its relation with the s plane	47
3.7.1	Representation of poles and zeros of $G(z)$ in the z plane	47
3.7.2	Relation between the z plane and the s plane	48
3.7.3	Establishment of some specific relations between s and z planes	49
3.8	Correspondence between pole locations in the z plane and system time response	57
3.9	Analysis of a simple loop containing a discrete-time controller	57
3.9.1	Example	57
3.9.2	Intuitive analysis of the equivalent discrete-time system	60
3.9.3	Z transform analysis of the discrete-time system	61
	Exercises	63
4	Methods of analysis and design	71
4.1	Introduction	71
4.2	Discretization	71
4.2.1	Introduction: discretization requirements for digital control	71

4.2.2	Principles of discretization	73
4.2.3	Approximation of differential equations by difference equations	76
4.2.4	Discretization by matching poles and zeros	78
4.2.5	Discretization by substitution of the first few terms in the series for $(1/T) \ln z$	81
4.2.6	Discretization by \mathcal{Z} transformation of $G(s)$	82
4.2.7	Discretization by matching the step response	83
4.2.8	Discretization by solution of the continuous equation over each time step	84
4.2.9	Discussion	87
4.3	The root-locus diagram	87
4.3.1	Some general rules for sketching the root locus	88
4.3.2	Example	89
4.3.3	Use of the root locus as a design tool	94
4.3.4	Two difficulties in the interpretation of the root locus in the z plane	95
4.4	Frequency-response methods	95
4.4.1	Direct method of frequency-response determination	95
4.4.2	The effect of z plane zeros on frequency response	97
4.4.3	Approximate frequency response through transformation of $G(z)$ to the w plane	98
4.4.4	Discussion	100
4.5	Stability tests	101
4.5.1	Jury's test for equations of order 2 and 3	102
4.5.2	Stability investigation by transforming to the w plane	103
	Exercises	105
5	Digital control algorithms	112
5.1	Introduction	112
5.2	Design methods	112
5.2.1	Approaches to digital controller design	112
5.2.2	Obtaining a simple continuous-time model of the process	113
5.2.3	Design approach 1: transformation of $G(s)$ into the z domain followed by digital design of a controller	115
5.2.4	Design approach 2: digital modelling followed by digital design	127
5.2.5	Design approach 3: continuous modelling, continuous control design, discrete approximation	130
5.2.6	Design approach 4: transformation of the problem into the w' domain followed by control design using the Bode diagram	133
5.3	Design descriptions	143

5.3.1	Discrete-time versions of feedforward and cascade-control algorithms	143
5.3.2	The simplest algorithm: on-off control	145
5.4	Design realizations: realization of algorithms in terms of unit delays and constant multipliers	146
5.4.1	Direct method	146
5.4.2	Parallel method	148
5.4.3	Factorization method	148
5.5	Design considerations	149
5.5.1	Choice of sampling interval	149
5.5.2	Quantization effects	160
5.5.3	Interaction between quantization effects and sampling rate effects	181
5.5.4	Effects of noise	182
5.5.5	Comparison of algorithms	185
5.6	Oscillations in control loops	186
	Exercises	187
6	Elements in the control loop	197
6.1	Introduction	197
6.2	Design of input circuits	198
6.2.1	Some typical measurement inputs	198
6.3	Actuators	202
6.3.1	Choice of control configuration	202
6.3.2	Velocity limiting in actuators	202
6.4	Analogue interfacing	203
6.4.1	Control aspects of A/D and D/A conversion	204
6.4.2	Digital-to-analogue conversion	204
6.4.3	Analogue output arrangements	206
6.4.4	Analogue-to-digital conversion	206
6.4.5	Sample-and-hold devices	209
6.4.6	Preceding an A/D converter by a sample-and-hold device	210
6.4.7	Input multiplexing	211
6.4.8	Choice of a multiplexer according to application	214
6.4.9	Multiplexer timing	216
6.4.10	Typical specifications of data-conversion devices	218
6.5	Digital interfacing	219
6.5.1	Introduction	219
6.5.2	Serial interfacing	220
6.5.3	Parallel interfacing	220
6.6	Microprocessors for control applications	222
	Exercises	223
7	Case histories	226
7.1	Introduction	226

A	Temperature control achieved by a low-cost ready-made configuration, programmed in the BASIC language	226
7.2	Outline of Problem A	226
7.3	Computation and interfacing	227
7.4	Control of data transfer	228
7.5	Identification of the oven	228
7.6	Control design	230
7.6.1	Design of a continuous controller followed by discretization	230
7.6.2	Addition of a term to compensate for oven heat losses	232
7.7	Choice of sampling interval	232
7.8	Input quantization	234
7.9	Alternative approach—Design of a temperature controller to achieve a particular closed-loop behavior	235
7.10	Comments on the temperature-control application	237
B	Design of a thickness and flatness control system for metal rolling	237
7.11	Outline of Problem B	237
7.12	Thickness measurement and control	237
7.13	Flatness measurement	238
7.14	Flatness control	239
7.15	Control-system specification	240
7.16	Choice of sampling interval T for the control loops	240
7.17	Choice of input-signal scanning arrangement	240
7.18	Choice of word lengths	241
7.19	Mathematical models of the process	242
7.20	Control-algorithm design	242
7.21	Fixing of the numerator $N(z)$	243
7.22	Calculation of the control algorithms	244
7.23	Realization of the control algorithms	246
7.24	Implementation	247
7.25	Program development	247
7.26	The complete scheme	248
	Exercises	248
8	State-variable techniques	254
8.1	Introduction	254
8.2	The concept of state	254
8.3	Alternative system descriptions	255
8.4	The mapping representation of Σ	256
8.5	The modelling of continuous-time systems through state-space equations	258
8.5.1	A general nonlinear model	258
8.5.2	Linearization of the model	259

8.6	Calculation of time solutions using the transition matrix . .	260
8.6.1	The time-invariant case	260
8.6.2	The time-varying case	265
8.6.3	The periodically time-varying case	267
8.7	Properties of the transition matrix	268
8.8	Relation between the transfer-matrix description and the vector-matrix description	269
8.9	Equivalent systems	269
8.10	System realization	270
8.10.1	Existence	270
8.10.2	Uniqueness	270
8.11	Stability	271
8.11.1	Stability tests for continuous-time systems	271
8.12	Reachability, controllability, observability and reconstruct- ibility for continuous-time systems	271
8.12.1	Controllability and observability tests for continuous- time, time-invariant systems by the matrices Q_c and Q_o	272
8.12.2	Controllability and observability tests for continuous- time, time-invariant systems by the diagonalization method	272
8.13	The unforced state equation in discrete time	273
8.13.1	Existence and uniqueness of solution	273
8.13.2	The transition matrix for the discrete-time equation	274
8.14	Solution of the discrete-time state equation with forcing	274
8.15	Obtaining the Z transform equivalent of the state equation	275
8.16	Stability tests	276
8.17	Reachability, controllability, observability and reconstruct- ibility for discrete-time systems	276
8.18	Canonical state-space representations	278
8.18.1	Introduction	278
8.18.2	The reachability canonical form	280
8.18.3	The controllability canonical form	282
8.18.4	The observability canonical form	285
8.18.5	The reconstructibility canonical form	286
8.18.6	State equations for multi-input-multi-output pro- cesses	287
8.18.7	The Jordan canonical form	288
8.19	The state-variable approach to control-system design	290
8.20	Design of non-interacting controllers	290
8.21	Design to achieve a specified closed-loop characteristic equa- tion	295
8.22	A brief introduction to optimal control	298
8.23	State estimation	303
8.23.1	The reasons for using state estimation	303

8.23.2	State observer	304
8.23.3	The principle of state estimation	304
8.23.4	The Kalman filter	305
8.23.5	Variants of the Kalman filter	310
8.24	The separation theorem	311
	Exercises	313
9	Control of large-scale systems	323
9.1	Introduction	323
9.2	Initial approaches to system decomposition	324
9.3	Hierarchical control	326
9.4	Multilayer control	326
9.5	Multilayer optimization	328
9.6	Digraphs as a tool in the modelling of complex systems	332
9.7	Multilevel control	339
9.8	The coordination problem in multilevel control	340
9.9	The steady-state coordination problem in multilevel control	341
9.10	The direct method of coordination	342
9.11	Implementation of the direct method using penalty functions	343
9.12	The interaction balance method	344
9.13	Coordination with feedback for on-line control	345
	Exercises	346
10	Implementation of control designs using distributed computer networks	348
10.1	Introduction	348
10.2	Distributed computer networks	348
10.2.1	Component parts of the network	348
10.2.2	The local controller	349
10.2.3	The data links	350
10.2.4	The coordinating controller	351
10.2.5	The central information display unit	352
10.3	Outline of the approach to applications	352
10.4	Provisions for enhancing reliability	353
10.4.1	Introduction	353
10.4.2	Redundancy within a local controller	353
10.4.3	One spare local controller as standby for several operational local controllers	354
10.4.4	Dynamic redundancy	355
10.4.5	Enhancing the reliability of the data links	355
10.5	Some general aspects	355
10.5.1	Further discussion of configurations and data-transfer protocols	355
10.5.2	Standard interfaces	357
10.5.3	Real-time languages	357

10.6	Criteria for selection of distributed control systems	358
10.7	Implementations by distributed computer networks as a natural follow-on from modern control design techniques	362
	Exercises	362
11	Commercially available distributed control systems and their industrial application	364
11.1	Introduction	364
11.2	A typical commercially available distributed control system—The Honeywell TDC 3000 system	364
11.2.1	Individual elements in the control loop	364
11.2.2	A typical system	368
11.2.3	Overall energy management system	368
11.3	Examples of other commercially available distributed com- puter systems	370
11.4	Case history 1: application of a batch controller to a thermoset resin plant	370
11.5	Case history 2: 'Damatic' system for control of paper mills	376
11.6	Case history 3: Modcomp minicomputer network for control of a pumped storage scheme	377
11.7	Case history 4: enhanced boiler drum level control	378
11.8	Concluding remarks	383
11.9	Mini-project	383
Appendix A	The structure and operation of a peripheral interface adaptor (PIA chip)	385
A1	Addressing the PIA chip	386
A2	PIA control lines	387
Appendix B	Tables of transform pairs	392
References and bibliography		396
Glossary of symbols		402
Index		404

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).