

Robust Control of Time-delay Systems

Qing-Chang Zhong

Robust Control of Time-delay Systems

With 79 Figures

 Springer

Qing-Chang Zhong, PhD
Department of Electrical Engineering and Electronics
The University of Liverpool
Brownlow Hill
Liverpool
L69 3GJ
UK

British Library Cataloguing in Publication Data

Zhong, Qing-Chang
Robust control of time-delay systems
1. Robust control 2. Time delay systems
I. Title
629.8'312

ISBN-10: 1846282640

Library of Congress Control Number: 2006921167

ISBN-10: 1-84628-264-0

e-ISBN 1-84628-265-9

Printed on acid-free paper

ISBN-13: 978-1-84628-264-5

© Springer-Verlag London Limited 2006

MATLAB® and Simulink® are the registered trademarks of The MathWorks, Inc., 3 Apple Hill Drive Natick, MA 01760-2098, U.S.A. <http://www.mathworks.com>

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms of licences issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publishers.

The use of registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant laws and regulations and therefore free for general use.

The publisher makes no representation, express or implied, with regard to the accuracy of the information contained in this book and cannot accept any legal responsibility or liability for any errors or omissions that may be made.

Printed in Germany

9 8 7 6 5 4 3 2 1

Springer Science+Business Media
springer.com

This book is dedicated to
SHUHONG, LILLY and LISA.

Preface

Systems with delays frequently appear in engineering. Typical examples of time-delay systems are communication networks, chemical processes, teleoperation systems, biosystems, underwater vehicles and so on. The presence of delays makes system analysis and control design much more complicated. During the last decade, we have witnessed significant development in the robust control of time-delay systems. The aim of this book is to present a systematic and comprehensive treatment of robust (H^∞) control of such systems in the frequency domain. The emphasis is on systems with a single input/output delay, although the delay-free part of the plant can be multi-input–multi-output (MIMO), when the delays in different channels are the same.

This book collects work carried out recently by the author in this field. It covers the whole range of robust (H^∞) control of time-delay systems: from controller parameterisation, controller design to controller implementation; from the Nehari problem, the one-block problem to the four-block problem; from theoretical developments to practical issues. The major tools used in this book are similarity transformations, chain-scattering approach and J -spectral factorisations. The main idea is to “*make everything as simple as possible, but not simpler (Albert Einstein).*” This book is self-contained and should be of interest to final-year undergraduates, graduates, engineers, researchers, and mathematicians who work in the area of control and time-delay systems.

The book is divided into two parts: Controller Design (Chapters 2–10) and Controller Implementation (Chapters 11–13). The classical control of time-delay systems is summarised in Chapter 2 and then some mathematical preliminaries are collected in Chapter 3. The J -spectral factorisation of regular para-Hermitian transfer functions is developed in Chapter 4 to prepare for the solution of the Nehari problem discussed in Chapter 5. An extended Nehari problem is solved in Chapter 6 to prepare for the solutions of the one-block problem and the standard H^∞ control problem discussed in Chapter 7, where the chain-scattering approach is applied to reduce the standard H^∞ control problem to a delay-free problem and a one-block problem. The latter is then further reduced to an extended Nehari problem. With the solution to the ex-

tended Nehari problem obtained in Chapter 6, the controllers for the one-block problem and the standard H^∞ problem are recovered. A transformed standard H^∞ problem is discussed in Chapter 8 to obtain a simpler but more conservative solution. The parameterisation of all stabilising controllers for time-delay systems are discussed in Chapter 9. All the controllers for the above problems have the same structure: incorporating a modified Smith predictor (MSP). A practical issue, a numerical problem with the MSP, is discussed in Chapter 10 and a unified Smith predictor is proposed to overcome this, followed by revisiting some well-studied problems. Another practical issue, the implementation of MSP, is tackled in Part II. The implementation of MSP, *i.e.*, a distributed delay, is not trivial because of the inherent hidden unstable poles. In Chapter 11, this is done by using discrete delays in the z -domain and in the s -domain. In Chapter 12, this is done by using rational transfer functions based on the δ -operator and then in Chapter 13 a faster converging rational implementation is discussed using bilinear transformations.

It is a pleasure to express my gratitude to G. Weiss, K. Gu, G. Meinsma and L. Mirkin. Special thanks go to O. Jackson (the Editor), S. Moosdorf (the Production Editor) and M. Saunders (the Copy Editor) for their professional and efficient editorial work on this book. There are no words that suffice to thank my wife Shuhong Yu for her endurance, love, support and sacrifice for my research over years. I am also grateful for financial support for my research from the Engineering and Physical Sciences Research Council (EPSRC), UK under Grant No. EP/C005953/1.

Liverpool, United Kingdom
January 2006

Qing-Chang Zhong
<http://come.to/zhongqc>

Contents

Notation and Abbreviations	xv
List of Figures	xix
List of Tables	xxi
1 Introduction	1
1.1 What Is a Delay?	1
1.2 Examples of Time-delay Systems	2
1.2.1 Shower	2
1.2.2 Chemical Processes	3
1.2.3 Communication Networks	3
1.2.4 Underwater Vehicles	4
1.2.5 Combustion Systems	6
1.2.6 Exhaust Gas Recirculation (EGR) Systems	7
1.2.7 Biosystems	8
1.3 A Brief Review of the Control of Time-delay Systems	9
1.4 Overview of This Book	10

Part I Controller Design

2 Classical Control of Time-delay Systems	17
2.1 PID Control	17
2.1.1 Structure of PID Controllers	17
2.1.2 Tuning Methods for PID Controllers	18
2.1.3 Simulation Examples	21
2.2 Smith Predictor (SP)-based Control	22
2.2.1 Control Difficulties Due to Delay	22
2.2.2 Smith Predictor	24
2.2.3 Robustness	25

2.2.4	Disturbance Rejection	27
2.2.5	Simulation Examples	28
2.3	Modified Smith Predictor (MSP)-based Control	30
2.3.1	Modified Smith Predictor	30
2.3.2	Zero Static Error	31
2.3.3	Simulation Examples	32
2.4	Finite-spectrum Assignment (FSA)	39
2.5	Connection Between MSP and FSA	39
2.5.1	All Stabilising Controllers for Delay Systems	39
2.5.2	Predictor–Observer Representation: MSP	40
2.5.3	Observer–Predictor Representation: FSA	41
2.5.4	Some Remarks	43
3	Preliminaries	45
3.1	FIR Operators	45
3.2	Chain-scattering Approach	46
3.2.1	Representations of a System: IOR and CSR	46
3.2.2	Linear Fractional Transformations: The Standard LFT and the HMT	48
3.2.3	Some Important Properties	49
3.3	State-space Operations on Systems	50
3.3.1	Operations on Systems	51
3.3.2	Similarity Transformations	57
3.4	Algebraic Riccati Equations	58
3.4.1	Definitions	58
3.4.2	Stabilising Solution	58
3.4.3	Block-diagram Representation	60
3.4.4	Similarity Transformations and Stabilising Solutions	61
3.4.5	Rank Defect of Stabilising Solutions	67
3.4.6	Stabilising or Grouping?	68
3.5	The Σ Matrix	68
3.5.1	Definition of the Σ Matrix	69
3.5.2	Important Properties of Σ	70
3.6	The $L_2[0, h]$ -induced Norm	72
4	J-spectral Factorisation of Regular Para-Hermitian Transfer Matrices	73
4.1	Introduction	73
4.2	Properties of Projections	74
4.3	Regular Para-Hermitian Transfer Matrices	75
4.4	J -spectral Factorisation of the Full Set	77
4.4.1	Via Similarity Transformations with Two Matrices	77
4.4.2	Via Similarity Transformations with One Matrix	78
4.5	J -spectral Factorisation of a Smaller Subset	79
4.6	J -spectral Factorisation of $A = G \sim JG$ with Stable G	82

4.7	Numerical Examples	84
4.7.1	$\Lambda(s) = \begin{bmatrix} 0 & \frac{s-1}{s+1} \\ \frac{s+1}{s-1} & 0 \end{bmatrix}$	84
4.7.2	$\Lambda(s) = \begin{bmatrix} -\frac{s^2-4}{s^2-1} & 0 \\ 0 & \frac{s^2-1}{s^2-4} \end{bmatrix}$	84
4.8	Summary	86
5	The Delay-type Nehari Problem	87
5.1	Introduction	87
5.2	Problem Statement (NP_h)	88
5.3	Solution to the NP_h	89
5.4	Proof	90
5.5	Special Cases	93
5.5.1	The Stable Case	93
5.5.2	The Conventional Nehari Problem	93
5.5.3	The Conventional Nehari Problem with Stable A	94
5.6	Realizations of Θ^{-1} and Θ	94
5.7	J -spectral Co-factor of Θ^{-1}	96
5.8	A Numerical Example	99
5.8.1	The Stable Case ($a < 0$)	100
5.8.2	The Unstable Case ($a > 0$)	103
5.9	Summary and Notes	108
6	An Extended Nehari Problem	109
6.1	Problem Statement	109
6.2	The Solvability Condition	110
6.3	Solution	110
6.4	Proof	111
6.4.1	Rationalisation by Z_1	111
6.4.2	Completing the J -losslessness	112
6.5	Realization of M	113
6.6	Summary	116
7	The Standard H^∞ Problem	117
7.1	Introduction	117
7.2	Problem Statements	119
7.2.1	The Standard H^∞ Problem (SP_h)	119
7.2.2	The One-block Problem (OP_h)	119
7.3	Reduction of the Standard Problem (SP_h)	120
7.3.1	The Standard Delay-free H^∞ Problem (SP_0)	120
7.3.2	Reducing SP_h to OP_h	121
7.3.3	Reducing OP_h to ENP_h	122
7.4	Solutions	124
7.4.1	Solution to OP_h	124
7.4.2	Solution to SP_h	124

7.5	Proof	125
7.5.1	Recovering the Controller	125
7.5.2	Realization of V^{-1}	126
7.6	Summary and Notes	128
8	A Transformed Standard H^∞ Problem	129
8.1	Introduction	129
8.2	The Transformation	130
8.3	Solution	132
8.4	A Numerical Example	135
8.5	Summary	138
9	2DOF Controller Parameterisation	139
9.1	Parameterisation of the Controller	139
9.2	Two-degree-of-freedom Realization of the Controller	142
9.2.1	Control Structure	142
9.2.2	Set-point Response	143
9.2.3	Disturbance Response	144
9.2.4	Robustness Analysis	145
9.2.5	Ideal Disturbance Response	145
9.2.6	Realization of $P_1 - NY P$	147
9.3	Application to Integral Processes with Dead Time	148
9.4	Summary	151
10	Unified Smith Predictor	153
10.1	Introduction	153
10.2	Predictor-based Control Structure	154
10.3	Problem Identification and the Solution	156
10.3.1	A Numerical Problem with the MSP	156
10.3.2	The Unified Smith Predictor (USP)	156
10.4	Control Systems with a USP: Equivalent Diagrams	160
10.5	Applications	164
10.5.1	Parameterisation of All Stabilising Controllers	164
10.5.2	The H^2 Problem	165
10.5.3	A Transformed H^∞ Problem	169
10.6	Summary	170

Part II Controller Implementation

11	Discrete-delay Implementation of Distributed Delay in Control Laws	173
11.1	Introduction	173
11.2	A Bad Approximation of Distributed Delay in the Literature ..	175
11.3	Approximation of Distributed Delay	176

11.3.1	Integration $\int_0^{\frac{h}{N}} y(t - \tau)d\tau$	176
11.3.2	Approximation in the s -domain via the Laplace Transform.....	177
11.3.3	Direct Approximation in the s -domain	180
11.3.4	Equivalents for the Backward Rectangular Rule	182
11.4	Implementation of Distributed Delay Z	183
11.4.1	Implementation of Z in the z -domain	183
11.4.2	Implementation of ZOH in the s -domain	184
11.4.3	Implementation of Z in the s -domain	185
11.5	Stability Issues Related to the Implementation	187
11.6	Numerical Examples	188
11.6.1	Approximations and Implementations of Distributed Delay	188
11.6.2	System Responses using Different Implementations	191
11.6.3	Numerical Integration Using the Improved Rectangular Rules	193
11.7	Summary.....	194
12	Rational Implementation Inspired by the δ-operator	195
12.1	Introduction	195
12.2	The δ -operator	196
12.3	An Initial Approximation.....	196
12.4	Implementation with Zero Static Error	197
12.5	Convergence of the Implementation	201
12.6	Structure of the Implementation.....	203
12.7	Numerical Examples	204
12.8	Summary.....	205
13	Rational Implementation Based on the Bilinear Transformation	207
13.1	Preliminary: Bilinear Transformation	207
13.2	Implementation of Distributed Delay.....	208
13.3	Convergence of the Implementation	211
13.4	Numerical Examples	214
13.5	Summary.....	216
	References	217
	Index	229

Notation and Abbreviations

\mathbb{Z}, \mathbb{R} and \mathbb{C}	fields of integral, real and complex numbers
$j\mathbb{R}$	imaginary axis
$\operatorname{Re} s$ and $\operatorname{Im} s$	real and imaginary parts of $s \in \mathbb{C}$
\in	belong to
\cap	intersection
\subset	subset
I_n	$n \times n$ identity matrix (n is often omitted when not confusing)
$J_{p,q}, J_\gamma$ and J	shorthand for $\begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix}$, $\begin{bmatrix} I & 0 \\ 0 & -\gamma^2 I \end{bmatrix}$ and $\begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}$
A^T and A^*	transpose and complex conjugate transpose of A
A^{-1} and A^{-*}	inverse of A and shorthand for $(A^{-1})^*$
$\det(A)$ and $\rho(A)$	determinant and spectral radius of A
$\operatorname{Im} A$	image of A
$\begin{bmatrix} A & B \\ C & D \end{bmatrix}$	explicitly partitioned matrix of $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$
$G(s) = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$	shorthand for $G(s) = D + C(sI - A)^{-1}B$
$G^\sim(s)$	shorthand for $G^T(-s) = [G(-s^*)]^* = \begin{bmatrix} -A^* & -C^* \\ B^* & D^* \end{bmatrix}$
$\mathcal{F}_l(M, Q)$ and $\mathcal{F}_u(M, Q)$	lower and upper linear fractional transformations (LFT)
$\mathcal{H}_r(M, Q)$ and $\mathcal{H}_l(M, Q)$	right and left homographic transformations (HMT)
$\mathcal{C}_r(M)$ and $\mathcal{C}_l(M)$	right and left chain-scattering transformations (CST)
$\pi_h(G)$	completion operator
$\tau_h(G)$	truncation operator

ARE	algebraic Riccati equation
w.r.t.	with respect to
iff	if and only if
IOR	input-output representation
CSR	chain-scattering representation
LFT	linear fractional transformation
HMT	homographic transformation
SP	(classical) Smith predictor
MSP	modified Smith predictor
USP	unified Smith predictor
FSA	finite-spectrum assignment
FIR	finite impulse response
SP_h	standard H^∞ problem with a single delay h
SP_0	the conventional standard H^∞ problem without a delay ($h = 0$)
OP_h	one-block problem with a delay h
ENP_h	extended Nehari problem with a delay h
NP_h	delay type Nehari problem
SISO	single-input single-output
MIMO	multiple-input multiple-output
ZOH	zero-order hold

List of Figures

1.1	Sketch of a shower system	3
1.2	Communication networks: A single connection	4
1.3	The MIT underwater vehicle: Odyssey II, Xanthos	5
1.4	Physical underwater vehicle system	6
1.5	Dynamics in a combustion system	7
2.1	Unity negative feedback control system	20
2.2	PID control: Example 1	22
2.3	PID control: Example 2	23
2.4	SP-based control system	24
2.5	SP-based control system: Internal model control	25
2.6	SP-based control system: Nominal case and $d = 0$	25
2.7	SP Example 1: Nominal responses	28
2.8	SP Example 1: Robustness	29
2.9	SP Example 2: Unstable plant	30
2.10	MSP Example 1: Stability region of (K_p, T_i)	33
2.11	Root-locus design of PI controller for MSP-based stable system	33
2.12	Performance of the MSP-based stable system: $K_p = 7, T_i = 2$	34
2.13	Performance of the MSP-based stable system: $K_p = 7, T_i = 0.2$	35
2.14	MSP Example 2: Stability region of (K_p, T_i)	36
2.15	Root-locus design for MSP-based unstable system	37
2.16	Performance of the MSP-based unstable system	38
2.17	Stabilising controllers for processes with dead time	40
2.18	Predictor–Observer Representation: MSP scheme	42
2.19	Observer–Predictor Representation: FSA scheme	42
3.1	Completion operator π_h and truncation operator τ_h	46
3.2	Input–output representation of a system	47
3.3	Chain-scattering representations of the system M in Figure 3.2	48
3.4	State feedback $u = Fx + v$	53
3.5	Output injection	53

3.6	Star product of interconnected systems	54
3.7	Similarity transformation on a system G with T	57
3.8	Block-diagram representation of algebraic Riccati equation	61
3.9	Solution generator when $T = \begin{bmatrix} I & 0 \\ L & I \end{bmatrix}$	63
3.10	Solution generator when $T = \begin{bmatrix} L & 0 \\ 0 & L^{-T} \end{bmatrix}$	64
3.11	Solution generator when $T = \begin{bmatrix} I & L \\ 0 & I \end{bmatrix}$	65
3.12	Solution generator when $T = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}$	66
3.13	Solution generator when $T = \begin{bmatrix} -I & 0 \\ 0 & I \end{bmatrix}$	67
5.1	Representation of the NP_h as a block diagram	90
5.2	Surface of $\hat{\Sigma}_{22}$ with respect to ah and $a\gamma$ ($a < 0$)	102
5.3	Contour $\hat{\Sigma}_{22} = 0$ on the ah - $a\gamma$ plane ($a < 0$)	102
5.4	Locus of the hidden poles of Z	103
5.5	Nyquist plots of Π_{22} ($a < 0$ and $ah = -1$)	104
5.6	Surface of $\hat{\Sigma}_{22}$ with respect to ah and $a\gamma$ ($a > 0$)	106
5.7	Contour $\hat{\Sigma}_{22} = 0$ on the ah - $a\gamma$ plane ($a > 0$)	106
5.8	Nyquist plots of Π_{22} ($a > 0$ and $ah = 1$)	107
6.1	Solving the ENP_h	111
7.1	General setup of control systems with a single I/O delay	119
7.2	Reduction of SP_h to SP_0 and a one-block delay problem	122
7.3	Reducing the one-block problem to ENP_h	123
7.4	Recovering the controller K	127
8.1	General control setup for dead-time systems	131
8.2	An equivalent structure	131
8.3	Graphic interpretation of the transformation	132
8.4	Setup for mixed sensitivity minimisation	135
8.5	Comparison of $\ T_{zw}(j\omega)\ $	136
8.6	Singular value plots of $Z_1(s)$	137
9.1	Parameterised control structure with 2DOF	143
9.2	The dual locus to judge controller stability	151
10.1	Dead-time plant with a predictor-based controller	155
10.2	Unified Smith predictor $Z = Z_s + Z_u$	158

10.3	Implementation of the USP (10.9) for multiple delays, using two resetting LTI systems to implement Z_u . The logical controller which controls the switches and generates the resetting signals R_a and R_b (in an open-loop manner) is not shown.	160
10.4	Control system comprising a dead-time plant P_h (with a rational part P) and a stabilising controller K	161
10.5	Control system from Figure 10.4, in which the controller K has been decomposed into a USP denoted Z and a stabilising compensator C , so that $K = C(I - ZC)^{-1}$	162
10.6	Equivalent representation of the control system in Figure 10.5, using the decomposition of P^{aug} given in Proposition 10.1.	162
11.1	Illustration of Lemma 11.1.	176
11.2	Magnitude coefficients (for a scalar A)	181
11.3	Implementations of Z in the z -domain	184
11.4	Impulse responses of different implementations of ZOH.	186
11.5	Errors of different approximations ($N = 20$)	189
11.6	E_{b0} and E_{f0} for different N (zero error at frequencies 0 and $+\infty$)	190
11.7	Implementation error of $Z_{f\epsilon}$ for different ϵ ($N = 1$)	191
11.8	Unit step response ($N = 8$)	192
11.9	Unit-step response: Z implemented as $Z_{f\epsilon}$ ($N = 1$)	193
12.1	Surface of $f(\sigma, \omega)$: $f(\sigma, \omega) = 0$ on the white lines	200
12.2	Contour of $f(\sigma, \omega)$ at level 0	200
12.3	Rational implementation: $Z_r = \sum_{k=1}^N \Pi^k \cdot \Phi^{-1} B$	203
12.4	Implementation error of Z_r for different N	204
12.5	Unit-step response.	205
13.1	Rational implementation of distributed delay: $Z_r = \sum_{k=0}^{N-1} \Pi^k \Xi B$	209
13.2	Circle into which all eigenvalues of $A \frac{h}{N}$ fall when $N > \bar{N}$	212
13.3	Area mapped from the right-half circle in Figure 13.2 via $\phi = c \frac{1+e^{-c}}{1-e^{-c}}$	212
13.4	Implementation error of Z_r for different N	215
13.5	Comparison of different implementations ($N = 5$)	215
13.6	System responses when $r(t) = 1(t)$	216

List of Tables

2.1	Ziegler–Nichols tuning formulas	19
2.2	PI control parameters for Example 1	21
2.3	PI control parameters for Example 2	22
3.1	CSR notation used in [58] and in this book	47
3.2	Similarities between CSR and IOR	50
3.3	Basic similarity transformation operations on a system	57
8.1	Performance comparison	136
10.1	USP needed for different types of plants	158

Introduction

Systems with delays abound in the world. One reason is that nature is full of transparent delays. Another reason is that time-delay systems are often used to model a large class of engineering systems, where propagation and transmission of information or material are involved. The presence of delays (especially, long delays) makes system analysis and control design much more complex. In this chapter, some examples of time-delay systems are discussed and then a brief review on the control of time-delay systems, followed by an overview of this book, is given.

1.1 What Is a Delay?

Time delay is the property of a physical system by which the response to an applied force (action) is delayed in its effect [124]. Whenever material, information or energy is physically transmitted from one place to another, there is a delay associated with the transmission. The value of the delay is determined by the distance and the transmission speed. Some delays are short, some are very long. The presence of long delays makes system analysis and control design much more complex. What is worse is that some delays are too long to perceive and the system is misperceived as one without delays.

Time delays abound in the world. They appear in various systems such as biological, ecological, economic, social, engineering systems etc. For example, over-exposure to radiation increases the risk of cancer, but the onset of cancer typically follows exposure to radiation by many years. In economics, the central bank in a country often attempts to influence the economy by adjusting interest rates; the effect of a change in interest rates takes months to be translated into an impact on the economy. In politics, politicians need some time to make decisions and they will have to wait for some time before they find out if the decisions are correct or not. When reversing a car around a corner, the driver has to wait for the steering to take effect. In engineering,

on which this book focuses, there are a lot of systems with delays; see the next section for examples.

A general tendency in responding to some errors in a system is to react immediately to the errors and to react more if the errors are not lessened or eliminated in time as expected. However, for a system with time delays, only after the inherent delays will the errors start to change. Hence, it is very important to properly understand the existence of delays and not to over-react. Otherwise, the system is very likely to overshoot or even become unstable. When dealing with time-delay systems, “patience is a virtue.”

For a given delay element with a delay $h \geq 0$, the output $y(t)$ corresponding to the input $u(t)$ is

$$y(t) = u(t - h).$$

Hence, the transfer function of a delay element is given by e^{-sh} .

1.2 Examples of Time-delay Systems

Some typical examples of time-delay systems in engineering are discussed here.

1.2.1 Shower

A simple example of a time-delay system from everyday life is the shower,¹ as depicted in Figure 1.1. Most people have experienced the difficulty in adjusting the water temperature: it gets too cold or too warm. The actual temperature often overshoots the desired and, sometimes, it takes a while to get the temperature right. This is because it takes time for the increased (or decreased) hot/cold water to flow from the tap to the shower head (or the human body). This time is a delay, which depends on the water pressure and the length of the pipe. The change of the faucet position is almost immediate, however, the change of the water temperature has to wait until the delay has elapsed. If the faucet position is constantly adjusted according to the currently perceived temperature, then it is very likely that the temperature will fluctuate.

Assume that the water is an incompressible fluid and stationary flow. According to the Poiseuille law, the flow rate of water is

$$F = \frac{\pi R^4}{8\mu l} \Delta p,$$

where $\mu = 0.01$ is the kinematic viscosity of water, R is the radius of the pipe, l is the length of the pipe and Δp is the pressure difference between the two ends of the pipe. The time delay h can then be found as

$$h = \frac{\pi R^2 l}{F} = \frac{8\mu}{\Delta p} \left(\frac{l}{R} \right)^2.$$

¹ A shower model written in SIMULINK® is available at <http://www.aut.bme.hu/education/contheor/contheor/theory/model/shower/>.

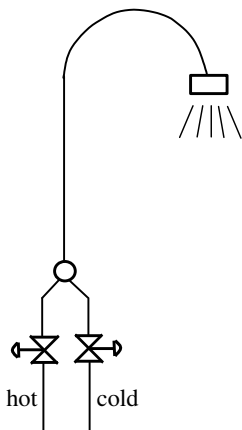


Figure 1.1. Sketch of a shower system

1.2.2 Chemical Processes

It is well known that many chemical processes contain delays. This has been well documented in the literature and there is no need to give any detailed example here. The following first-order plus dead time (FOPDT) is widely used to model chemical processes:

$$G(s) = \frac{K}{Ts + 1} e^{-sh},$$

where K is the static gain of the plant, $T > 0$ is the time constant and h is the transparent delay or dead time.

1.2.3 Communication Networks

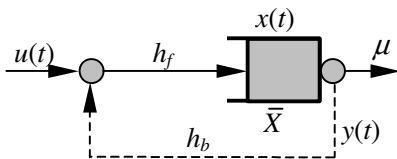
In recent years, communication networks [143] have been among the fastest-growing areas in engineering and there has been increasing interest in controlling systems over communication networks. Thanks to high-speed networks, control-over-Internet is now available [69, 122]. These systems are frequently modelled from the control point of view as time-delay systems because of the inherent propagation delays [52, 71]. These delays are crucial to the system stability and the quality-of-service (QoS).

A single connection between a source controlled by an access regulator and a distant destination node served with a constant transmission capacity μ is given as an example here. This can be described by the fluid model [52, 71] shown in Figure 1.2(a). At the source node, the access regulator controls the input rate $u(t)$, according to the congestion status of the destination node.

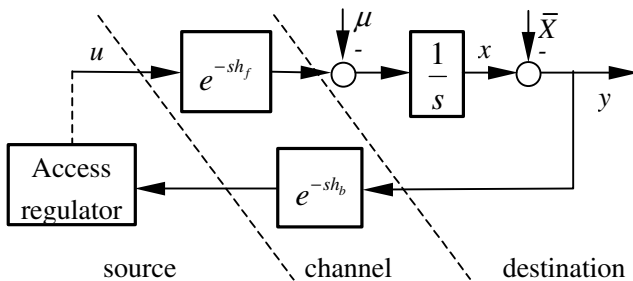
The congestion status $y(t)$ is defined as the difference between the current buffer contents $x(t)$ and the target value \bar{X} , *i.e.*,

$$y(t) = x(t) - \bar{X}.$$

Due to the propagation delay from the destination node to the source node, called the backward delay h_b , this status arrives at the source node (the access regulator) only after this delay period. There is also a forward delay h_f for the package to arrive at the destination node from the source node. The arrived packages are stored/accumulated in the buffer and then sent with a constant transmission capacity μ . The control objective is to adapt $u(t)$ to μ dynamically while maintaining the buffer $x(t)$ at an acceptable level. The block diagram is shown in Figure 1.2(b).



(a) fluid model



(b) block diagram

Figure 1.2. Communication networks: A single connection

The communication networks in reality, which are built from single connections, are much more complicated. The delays are often time-varying and stochastic. The information transmitted via communication networks is quantised and there exist package losses as well.

1.2.4 Underwater Vehicles

Recently, there have been more and more applications of underwater vehicles. They can be used for exploring ocean bottoms, installation/inspection/repair

tasks and, of course, military missions. They have advantages over human divers in that they can descend to greater depths, can stay there for greater lengths of time and require less support equipment. Thus they can reach places divers cannot, and they can be less expensive to operate [64].

There are different types of underwater vehicles. One is the *Remotely Operated Vehicle* (ROV). An ROV is connected to a surface support ship via an umbilical cable, which provides power supply and a communication link, and hence the range of operation is somewhat limited. Another one is the *autonomous underwater vehicle* (AUV), which carries an on-board power unit and is equipped with advanced control capabilities to undertake tasks with the minimum of human intervention. The communication is carried out through an acoustic link. The MIT underwater vehicle Odyssey II Xanthos, shown in Figure 1.3, is taken as an example. Odyssey II Xanthos is a video survey AUV, equipped with various sensors including scanning and homing sonars, depth sensor, temperature salinity and related sensors, video, inertial sensors, acoustic modem and acoustic navigation tracking pingers. The rated operating depth is 3,000 m. More details can be found at <http://auvlab.mit.edu>.



Figure 1.3. The MIT underwater vehicle: Odyssey II, Xanthos

(Courtesy of C. Chryssostomidis and R. Damus, MIT Sea Grant AUV Lab, <http://auvlab.mit.edu>)

The control problems involved in these vehicles include navigation, task planning and the low-level autopilot. Due to the long cable or distance, there exists a long delay in the system. For AUVs, the delay is caused by the finite sound speed in water, nominally, 1,500 m/s.

A physical system [140] is shown in Figure 1.4, where a surface ship is shown positioning an underwater vehicle through a long cable. The vehicle may be searching the ocean floor or mapping the topography of the bottom, or it may be the platform for a smaller vehicle equipped with thrusters. An

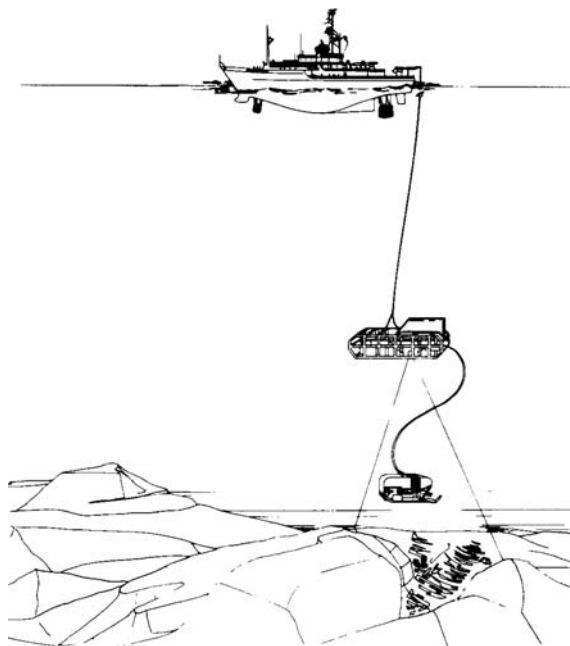


Figure 1.4. Physical underwater vehicle system

(Reprinted, with permission, from [140]. ©IEEE)

approximate model, which was validated for a cable of 2,500 m and a vehicle weighing 17,000 N in air, was given in [140] as

$$G(s) = \frac{ce^{-sh}}{as^2 + bs + c},$$

with $a = 1$, $b = 1.1 \times 10^{-4}$, $c = 2.58 \times 10^{-2}$ and $h = 40$ s. More details about this system can be found in [140] and the references therein.

1.2.5 Combustion Systems

Continuous combustion systems are widely used in power generation, heating and propulsion. Examples include domestic and industrial burners, steam and gas turbines, waste incinerators, and jet and ramjet engines. These systems are intricate and include a wide variety of dynamic behaviour. Pressure oscillations are considered the most significant in terms of the impact on system performance; much effort has been devoted to this [3, 24, 103].

There are two major dynamics in a combustion system: flame dynamics and acoustic wave dynamics. They are coupled to form a loop as shown in

Figure 1.5. Due to wave propagation, there is a delay in the wave dynamics. This often causes combustion instability [3, 24, 103]. Delays also appear in the measurement and the actuator units of the system. Detailed modelling of combustion systems can be found in [3, 24] and are omitted here.

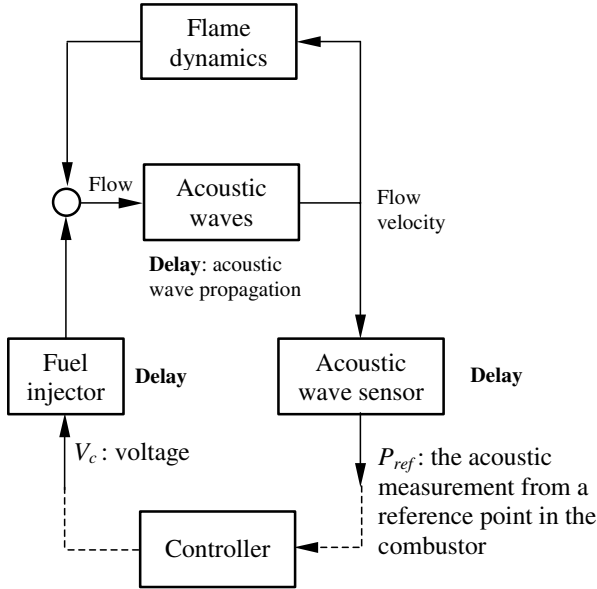


Figure 1.5. Dynamics in a combustion system

1.2.6 Exhaust Gas Recirculation (EGR) Systems

Oxides of nitrogen (NO_x) are a group of highly reactive gases that contain varying amounts of NO and NO_2 . These are key elements of greenhouse gases and harmful to human health and the environment. Motor vehicles are one of the key sources of NO_x . Exhaust gas recirculation (EGR) systems were introduced in the early 1970s to reduce NO_x emissions.

The EGR valve recirculates exhaust into the intake stream. Exhaust gases have already combusted, so they do not burn again when they are recirculated. These gases displace some of the normal intake charge. This chemically slows and cools the combustion process by several hundred degrees, thus reducing NO_x formation.²

It is a challenge to precisely control the flow of recirculated exhaust so that the system provides good performance and economy. Too much flow will retard

² <http://www.asashop.org/autoinc/nov97/gas.htm>.

engine performance and cause a hesitation on acceleration, and too little flow will increase NO_x and cause engine ping. Some EGR systems simply operate in open loop due to nonlinearity, engine vibrations, pressure fluctuations and high-order unmodelled dynamics [109]. The measurement, *e.g.*, of oxygen in the exhaust [1] for feedback, often introduces delay, which complicates the control design. On the other hand, it is very difficult to derive a mathematical model for the system. The model is often obtained via system identification. The following model was given in [109]:

$$G_i(z) = k(z - p)z^{-d} \quad (i = 1, 2, 3, 4)$$

at four operating points dependent on load, speed and the desired EGR rate for an EGR system. With a sampling period of 10 ms, the vectors of the parameters are:

$$\begin{aligned} \text{gain } k &= (88, 110, 180, 220), \\ \text{delay } d &= (5, 15, 13, 6), \\ \text{pole } p &= (0.9, 0.9, 0.9, 0.9). \end{aligned}$$

The model changes dramatically at different operating points. The delay varies from 50 ms to 150 ms and the gain varies from 88 to 220.

1.2.7 Biosystems

This subsection is based on [10, 16], in which more examples and references can be found.

Time delay has been introduced to model biosystems to produce better consistency with nature and predictive results. The ultimate objective is to further understand the systems and then to control them. The biosystems studied in this field cover population dynamics, epidemiology, physiology, immunology, neural networks and cell kinetics.

The following delay model for population dynamics was introduced in [48], modifying the classical Verhulst model to account for hatching and maturation periods:

$$y'(t) = ry(t) \left(1 - \frac{y(t - \tau)}{K} \right).$$

Here, the nonnegative parameters r and K are, respectively, the intrinsic growth rate and the environmental carrying capacity. This simple model can explain the observed oscillatory behaviour in a single species population, without any predatory interaction of other species.

Another simple example is the model of growth in cell populations, which is given by

$$y'(t) = \alpha y(t) + \beta y(t - \tau).$$

The equilibrium solution $y(t) = 0$ becomes unstable when the value of the delay exceeds the following bound: