

©
All rights reserved

This work, in whole or in part, may not be copied (in any media), printed, translated, stored in a retrieval system, transmitted via the internet or other electronic means, except for "fair use" of brief quotations for academic instruction, criticism, or research purposes only.
Commercial use of this material is completely prohibited.

©
כל הזכויות שמורות

אין להעתיק (במדיה כלשהי), להדפיס, לתרגם, לאחסן במאגר מידע, להפיץ באינטרנט, חיבור זה או כל חלק ממנו, למעט שימוש הוגן בקטעים קצרים מן החיבור למטרות לימוד, הוראה, ביקורת או מחקר.
שימוש מסחרי בחומר הכלול בחיבור זה אסור בהחלט.

ROBUST CONTROL USING DEAD-TIME COMPENSATORS

Research Thesis

Submitted in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Mechanical Engineering

ROMAN GUDIN

Submitted to the Senate of the Technion — Israel Institute of Technology

Kislev 5768 Haifa November 2007

The Research Thesis Was Done Under The Supervision of Prof. Leonid Mirkin
in the Faculty of Mechanical Engineering

Acknowledgment

I would like to say sincere words of gratitude to Prof. L. Mirkin for complete support throughout this research and my studies.

Thanks to the team of the “Control of Flexible Structures” Laboratory, Ilya Shamis and Victor Royzen, for the technical support of the experimental part of the research.

Generous financial help of the Technion is gratefully acknowledged.

Contents

Abstract **1**

Lists of acronyms and notation **3**

1 Introduction **5**
1.1 Dead-time systems . . . 5
1.1.1 Effects of loop delay . . . 8
1.1.2 Approaches to control of DT systems . . . 11
1.2 Dead-time compensation . . . 14
1.2.1 The Smith predictor . . . 14
1.2.2 Modifications and alternatives . . . 16
1.2.3 Some DTC shortcomings . . . 17
1.3 The goals of this research . . . 19
1.4 Organization of the thesis . . . 19

2 Delay margin of DTCs **21**
2.1 Preliminary: delay margin (dead-time tolerance) . . . 21
2.2 Control of an integrator and dead-time . . . 23
2.2.1 Two-stage design with a static primary controller . . . 23
2.2.2 H^∞ loop shaping . . . 26
2.3 Generalizations . . . 28
2.3.1 High-gain design of Smith predictor . . . 28
2.3.2 M and N circles . . . 28
2.3.3 Bode's gain-phase relation . . . 30
2.4 Some design guidelines . . . 31
2.5 Technical derivations . . . 32
2.5.1 Modified Smith predictor . . . 32
2.5.2 H^∞ loop shaping . . . 36

3 Implementation of controllers including FIR blocks **41**
3.1 Servo system for a delayed DC motor . . . 41
3.1.1 Classical loop shaping . . . 42
3.1.2 DTC design via the H^∞ loop shaping . . . 44
3.2 Analog implementation of DTC controllers . . . 46
3.2.1 Lumped-delay approximations of distributed-delay elements . . . 47
3.2.2 Implementation of distributed-delay elements using resetting mechanism . . . 47
3.2.3 Some comparisons . . . 49
3.3 Analysis of the LDA method . . . 50

3.3.1 Out of the box implementation . . . 50
3.3.2 Inaccuracy mechanisms . . . 51
3.3.3 Balancing controller loop: loop shifting . . . 53
3.3.4 Fast stable dynamics of the H^∞ DTC . . . 57
3.3.5 The use of non-proper weights . . . 59
3.4 Control of the laboratory pendulum . . . 63
3.4.1 Pendulum . . . 64
3.4.2 Inverted pendulum . . . 64

4 Conclusions and future research **67**

A Laboratory experiment **69**
A.1 Experiment description . . . 69
A.2 Systems modeling . . . 71
A.2.1 Equations of motion of the mechanical part . . . 71
A.2.2 Equations of motion of the electro-mechanical part . . . 73
A.2.3 State equations and linearization . . . 74
A.2.4 Identification of I_r and b . . . 75
A.2.5 Identification of ζ . . . 77
A.2.6 Transfer functions of the experimental setup . . . 77

B Matlab implementation details **79**
B.1 Simulink implementation of the resetting mechanism . . . 79
B.2 Auxiliary Matlab functions . . . 80
B.2.1 FIR block implementation by the RM method for SISO systems . . . 80
B.2.2 FIR block implementation by the RM method for SIMO systems . . . 82
B.2.3 FIR block implementation by the LDA method . . . 83

References **84**

Hebrew Abstract **xi**

List of Figures

1.1 Delay operator . . . 5
1.2 Systems with loop delay for Examples 1.1 and 1.2 . . . 6
1.3 The transfer function of a heated can and its approximation (Example 1.3) . . . 7
1.4 Free-free uniform rod (Example 1.4) . . . 8
1.5 Feedback control system setup with loop delay . . . 8
1.6 Frequency response of $L_{\mathrm{r}}(s) = \frac{5}{s+1}\mathrm{e}^{-sh}$ for $h = 0, 0.1, 1$ (Example 1.5) . . . 9
1.7 Delayed plant and its Padé approximation (Example 1.7) . . . 13
1.8 Observer-predictor control scheme . . . 14
1.9 Control system setup with Smith controller . . . 15
1.10 Equivalent representation of Smith controller . . . 15
1.11 General DTC setup . . . 16

2.1 Plant with several crossover frequencies: $L(s)$ (solid line) and $L(s)\mathrm{e}^{-0.5s}$ (dashed line) . . . 22
2.2 General DTC setup . . . 24
2.3 Stability margins vs. normalized controller gain $hk_pC(0)$ (two-stage design) . . . 24
2.4 Nichols charts of $L(s)$ at the first crossover proliferation . . . 25
2.5 Nichols charts of $L(s)$ for different crossover proliferation (two-stage design) . . . 26
2.6 Control system setup for H^∞ loop shaping . . . 27
2.7 Stability margins vs. normalized controller gain $hk_pC(0)$ (H^∞ loop shaping) . . . 27
2.8 Nichols charts of $L(s)$ for different crossover proliferation (H^∞ loop-shaping design) . . . 28
2.9 M and N circles as Nichols charts grid . . . 29
2.10 $C(s)$ for the designs in Section 2.2 . . . 30
2.11 $\varkappa$ vs. $\tilde{\omega}_{\mathrm{c}}$ . . . 34
2.12 Optimal performance as a function of the loop delay h . . . 39

3.1 Nichols charts of $P_{\mathrm{m}}(s)\mathrm{e}^{-sh}$ with different delays . . . 42
3.2 Classical loop-shaping designs for $P_{\mathrm{m}}(s)\mathrm{e}^{-sh}$ . . . 43
3.3 H^∞ loop-shaping designs for $P_{\mathrm{m}}(s)\mathrm{e}^{-sh}$ . . . 45
3.4 Control system setup for H^∞ loop shaping . . . 45
3.5 Loop transfer function for DTC controller and its rational approximation . . . 46
3.6 Lumped-delay approximation (LDA) of $\Pi(s)$ . . . 48
3.7 Resetting mechanism (RM) setup for implementing $\Pi(s)$ . . . 48
3.8 Realization of block Π using observer form . . . 49
3.9 Realization of block Π using state-space form . . . 49
3.10 Nichols plot of L for Π and its approximations for different ν . . . 51
3.11 Bode plots of $\tilde{C}$ and Π for different delays . . . 52
3.12 Frequency response of $\tilde{C}\Pi$ for resulting regulators . . . 52
3.13 Bode plots of the components of $\Pi = \tilde{P} - \hat{P}\,\mathrm{e}^{-sh}$ in the low-frequency range . . . 53
3.14 Loop shifting stages . . . 54

3.15 Lumped-delay approximation (LDA) of $\Pi_{\text{irr}}(s)$ . . . 55
3.16 Internal loop components after loop shifting . . . 55
3.17 Frequency response of $\Pi_{\text{irr}}\tilde{C}_{\text{s}}$ . . . 56
3.18 Step responses for $h = 0.1$ . . . 56
3.19 Step responses for $h = 0.15$ . . . 57
3.20 Step responses for $h = 0.2$ . . . 57
3.21 Root loci for the poles of $\hat{P}$ . . . 59
3.22 Designs with strictly proper (W_2) and non-proper ($W_{\bar{\text{p}}2}$) weights ($h = 0.15$) . . . 60
3.23 Designs with strictly proper (W_3) and non-proper ($W_{\bar{\text{p}}3}$) weights ($h = 0.2$) . . . 61
3.24 Root loci for the poles of $\hat{P}$ with non-proper weighting functions . . . 61
3.25 Designs with proper ($W_{\text{p},3}$) and non-proper ($W_{\bar{\text{p}}3}$) weights ($h = 0.2$) . . . 62
3.26 Root loci for the poles of $\hat{P}$ with proper weighting function $W_{\text{p},3}$ ($h = 0.2$) . . . 63
3.27 Step responses for $h = 0.15$ (design with $W_{\bar{\text{p}}2}$) . . . 63
3.28 Step responses for $h = 0.2$ (design with $W_{\text{p},3}$) . . . 64
3.29 Step responses for the pendulum experiment ($h = 0.1$) . . . 65
3.30 Step responses for the inverted pendulum experiment ($h = 0.2$) . . . 66

A.1 The experiment system . . . 69
A.2 The experiment system (zoomed in) . . . 70
A.3 Disassembled upper system part . . . 70
A.4 Sketch of the experiment system . . . 71
A.5 v_{c} sketch . . . 72
A.6 Experiment for identifying I_{r} and b . . . 75
A.7 Block-diagram of the DC motor with axial load . . . 76
A.8 Time response in the experiment for identifying I_{r} and b . . . 76
A.9 Time responses in the experiment for identifying ζ . . . 77
A.10 Frequency response of the pendulum $P_{\text{down}}(s) = \left[\begin{array}{cc} P_y(s) & P_\theta(s) \end{array}\right]'$ . . . 78
A.11 Frequency response of the inverted pendulum $P_{\text{up}}(s) = \left[\begin{array}{cc} P_y(s) & P_\theta(s) \end{array}\right]'$ . . . 78

B.1 Block scheme for implementing resetting FIR blocks in Simulink . . . 79
B.2 Signal from the pulse generator . . . 80

List of Tables

3.1 ν for LDA of Π for different h 50
3.2 Eigenvalues of resulting Hamiltonian matrices 53
3.3 ν for LDA of Π for different h (loop shifting) 56
3.4 ν for LDA of Π for different h (non-proper weights) 62

A.1 Measurable parameters of the experimental setup 75
A.2 Identified parameters of the experimental setup 77

Abstract

Dead-time systems are control systems containing time delays in the feedback loop. One widely used scheme for the control of dead-time systems is the so-called dead-time compensation (DTC) controller configuration. The idea here is to pull the loop delay out of the feedback loop by introducing an irrational internal loop into the controller. The DTC scheme was proposed by Otto J. M. Smith half a century ago (the Smith predictor). Since then, the Smith predictor and its numerous modifications become a part of the standard toolkit for control of dead-time systems. Moreover, it was recently shown that DTCs are an intrinsic part of optimal controllers under H^2, H^∞, and L^1 performance measures, which might be thought of as an analytic justification of the DTC configuration.

It is known that DTCs may be very sensitive to uncertainty in the loop delay (weak dead-time tolerance) and troublesome in implementation. Yet the underlying mechanisms of these problems are still not well understood. This research examines the aforementioned two problems associated with DTCs. Specifically, the following two aspects of the robust DTC design are investigated.

1. It is shown that the reasons for the high sensitivity to delay uncertainty can be revealed using standard Nyquist criterion arguments. It is argued that this phenomenon is caused by *proliferation of crossover frequencies*, which, in turn, is facilitated by the use of DTCs. The dead-time tolerance of DTCs is first studied for the plant containing an integrator and a dead time, which is a standard benchmark problem in the control of dead-time systems. Both the classical two-stage design procedure and the analytical H^∞ loop-shaping approach are applied to show that the increase of the closed-loop bandwidth inevitably leads to the increase of the number of crossover frequencies (crossover proliferation). This, in turn, gives rise to a dramatic decrease of the achievable dead-time tolerance. As a byproduct of the proposed analysis, the discontinuity of the delay margin as a function of the controller parameters is shown, which might question the numerical robustness of some known methods for calculating the delay margin of DTCs. The proposed analysis is then extended to more general plants using such tools of the classical loop shaping as the M and N circles and the Bode's gain-phase relation. Finally, somewhat more conscious guidelines for the design of DTC-based controllers are proposed.

2. Existing approaches to the numerical implementation of DTC-based controllers are studied. The key issue here is the approximation of a distributed-delay (DD) part of the controller. The main emphasis of this research is placed on the so-called *lumped-delay approximation* (LDA) approach, in which the DD element is approximated by a system with commensurate lumped delays, similarly to the Riemann sum approximation of integrals. It turns out that the out-of-the-box use of this approach might not be feasible for the implementation even on rather powerful real-time control hardware. The reasons for this are shown to be the ill-posedness of the internal loop of the controller and a significant growth of the gain of the DTC block, which is caused by its fast stable modes. It is shown that these problems can be substantially alleviated by loop shifting of the internal loop. The idea is to extract a stable rational part from the DTC and to augment it to the primary controller. This reduces the gain of the DTC part and improves the robustness of the internal loop of the controller.

In addition, the source of the fast stable modes of the DTC block under the H^{∞} loop shaping design is shown to be the fast modes of the weighting function. As a possible remedy the use of non-proper weighted functions is proposed, which further improves the robustness of the controller internal loop. The proposed solutions are validated by laboratory experiments.

Lists of acronyms and notation

Lists of acronyms

BIBO	Bounded Input / Bounded Output
DD	Distributed Delay
DT	Dead-Time
DTC	Dead-Time Compensation
FSA	Finite Spectrum Assignment
LQG	Linear Quadratic Gaussian
LDA	Lumped Delay Approximation
LTI	Linear Time-Invariant
MIMO	Multiple-Inputs / Multiple-Outputs
MSP	Modified Smith Predictor
RM	Resetting Mechanism
RHP	Right Half Plane
LHP	Left Half Plane
SISO	Single-Input / Single-Output
SIMO	Single-Input / Multiple-Outputs

Notation

h	time delay
μ_{g}	gain margin
μ_{ph}	phase margin
μ_{d}	delay margin/dead-time tolerance (defined on p. 23)
μ_{cp}	crossover proliferation margin (defined on p. 32)
ω_{c}	crossover frequency
$C(s)$	controller
$\tilde{C}(s)$	primary controller
$P_{\mathrm{r}}(s)$	rational part of the plant $P(s)$
$\Pi(s)$	dead-time compensator
$L(s)$	loop transfer function
$\pi_h\{G(s)\mathrm{e}^{-sh}\}$	FIR completion of the time-delay system $G(s)\mathrm{e}^{-sh}$ (defined on p. 37)

Chapter 1

Introduction

1.1 Dead-time systems

In this thesis dead-time systems, i.e., systems with loop delays are studied. We thus start with a brief description of the delay phenomenon in control systems. The continuous-time delay element D_h is pre-

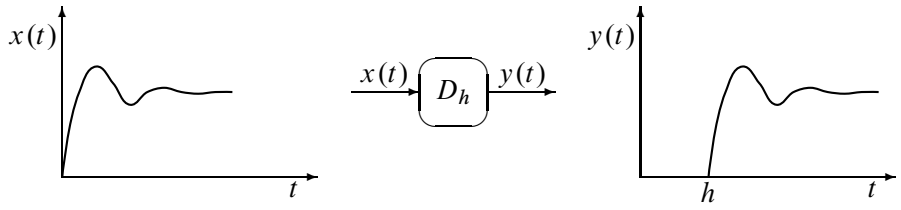


Figure 1.1: Delay operator

sented in Fig. 1.1, where the output signal $y(t)$ is the delayed, by h units of time, copy of the input signal $x(t)$. Formally, the delay block D_h can be defined as

$$y(t) = D_h x(t) \quad \Longleftrightarrow \quad y(t) = x(t-h).$$

In this work only constant delays are considered. It can be shown that in this case D_h is linear time invariant (LTI) and BIBO stable and has the *irrational* transfer function $D_h(s) = \mathrm{e}^{-hs}$.

Loop delays subsist in numerous systems. There are transport and communication delays in chemical, mechanical, and biological systems (Marshall *et al.*, 1992; Gu *et al.*, 2003). Delays can be caused by digital processing in control applications, measurement delays. Moreover, delays can arise as a part of plant dynamics: sometimes a plant given by high order rational transfer function can be approximated by rather simple transfer function and delay part (Zwart and Bontsema, 1997). Delays are also a natural part of systems controlled via communication network due to buffering and propagation delays (Hespanha *et al.*, 2007). The four examples below present some applications in which time delays arise.

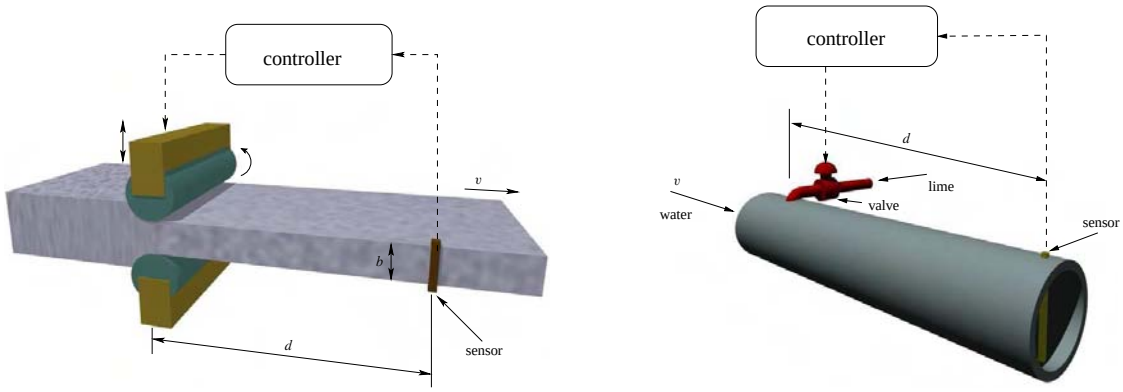
Example 1.1 *A hot rolling mill is depicted in Fig. 1.2(a). Here a pair of opposing rollers is used to flatten hot steel billet into uniform sheets of the thickness b. A thickness sensor, downstream from the rollers, gauges the sheet and causes the controller to apply (more or less) pressure to compensate for any out-of-spec thickness.*

Ideally, the thickness gauge should be positioned immediately adjacent to the rollers to minimize the dead time (DT) between the change in roller pressure and the resulting change in the thickness measurement. Otherwise, the controller will be unable to detect mistakes it may have been making soon enough

to prevent even more of the sheet from turning out too thick or too thin. If the optical thickness gauge is too far from the rollers in this steel rolling example, the controller will take too long to correct thickness errors.

Unfortunately, the thickness gauge cannot be placed over the exact spot where the steel's thickness is being manipulated by the rollers. Each newly flattened chunk of the sheet must travel at least some distance downstream before it can be measured. Therefore, some DT is unavoidable. If the steel is traveling at a velocity v, then the DT in this case between any steel thickness change and its recording by the sensor is $h = \frac{d}{v}$.

This phenomenon, known as transport delay, plagues many processes that involve a material traveling from the actuator to the sensor: liquid flowing through a pipe, conditioned air traveling down a duct, material moving along a conveyor belt, etc. In each case, moving the sensor upstream as far as possible can reduce the DT, but it cannot be eliminated. ▽



(a) Steel rolling mill control system (b) Water acidity control system

Figure 1.2: Systems with loop delay for Examples 1.1 and 1.2

Example 1.2 *Another example of the transport delay is the acidity control system depicted in Fig. 1.2(b). Such system is usually used to control the acidity of water draining from coal mine by adding lime to the water. Here a sensor located downstream gauges the water acidity and the controller adjusts the valve to add the required amount of lime to the water.*

As in the previous examples, DT is unavoidable in this system. Any attempt to locate the acidity sensor adjacent to the point of the lime supply will bring the system to fault because of unreliable data collecting by the sensor. Some minimal distance d is needed to allow full mixing before sensing. Assuming that the amount of lime is negligible with respect to the amount of water, the delay between the lime supply point and the acidity water sensing can be calculated as $h = \frac{d}{v}$, where v is the water velocity, which can be measured. ▽

Example 1.3 *This example, taken from (Zwart and Bontsema, 1997), demonstrates that an infinite dimensional transfer function can be successfully approximated by a rather simple transfer function and a delay part. The transfer function of a heated can (from the heating steam temperature to the temperature of a coordinate in the can) is*

$$G(s) = \frac{1}{J_0\left(\sqrt{\frac{-s}{\alpha}}R\right)} + \sum_{m=1}^{\infty}\left[\frac{2}{\lambda_m R J_1(\lambda_m R)} \cdot \frac{s}{s+\alpha\lambda_m^2} \cdot \frac{1}{\cosh\left(\frac{L}{2}\sqrt{\frac{s}{\alpha}+\lambda_m^2}\right)}\right],$$

where R is the radius of the can, L is the height of the can, α is a positive constant denoting the thermal diffusivity, J_0 is the Bessel function of zero order defined as

$$J_0(\rho) := \sum_{k=0}^{\infty} \frac{(-1)^k}{(k!)^2} \left(\frac{\rho}{2}\right)^{2k},$$

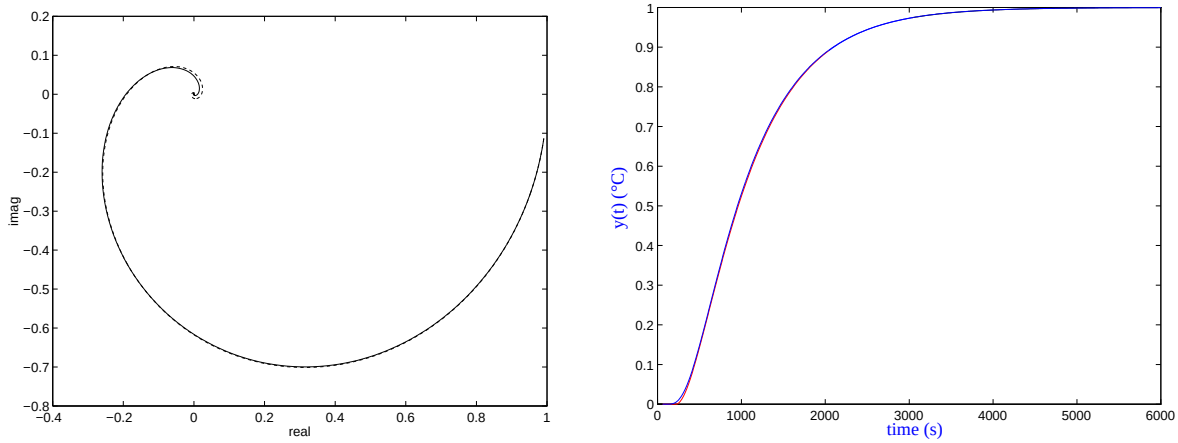
and J_1 is the Bessel function of order one defined as

$$J_1(\rho) := \sum_{k=0}^{\infty} \frac{(-1)^k}{(k!)^2 \Gamma(k+\rho+1)} \left(\frac{\rho}{2}\right)^{2k+\rho},$$

where $\Gamma(z) := \int_0^\infty t^{z-1} \mathrm{e}^{-t} \mathrm{d}t$ and λ_m is the mth positive zero of $J_0(R)$. The transfer function $G(s)$ above is too complicated to use in the controller design. One may consider a truncation of the infinite sum to obtain a finite dimensional approximation. Yet the convergence of this series is very slow, so that the order of truncated approximation should be very high to guarantee a reasonable accuracy. An attractive alternative is to approximate $G(s)$ by the following irrational transfer function:

$$G_a(s) = \frac{1}{(\tau_1 s + 1)(\tau_2 s + 1)} \mathrm{e}^{-sh},$$

where the time constants τ_1 and τ_2 and the dead time h can be found via a numerical optimization procedure (Zwart and Bontsema, 1997). Comparing the Nyquist plot and the step response of analytically



(a) Nyquist plot (b) step response

Figure 1.3: The transfer function of a heated can and its approximation (Example 1.3)

calculated transfer function and its approximation (Fig. 1.3), one can see that this approximation is reasonably good. Thus, the model with dead time not only gives a good approximation, but also simplifies the analysis of the resulting plant. ▽

Example 1.4 *Some systems can be described as models with DT. Consider, for example, a free-free uniform rod excited by a torque moment $M(t)$ at one of its ends shown in Fig. 1.4. Here I_p denotes the polar moment of inertia, G is the shear elasticity modulus, ρ is the material density, $c = G\rho$ is the wave propagation velocity, and $\beta = \frac{x}{L}$ denotes a non-dimensional measurement coordinate. The transfer function from the applied moment to the torsion angle as a function of the spatial coordinate x (which can be interpreted as the sensor location) is (Raskin, 2000)*

$$\frac{\theta(x,s)}{M(s)} = \frac{c}{GI_p} \frac{1}{s} \frac{\mathrm{e}^{-\frac{2L}{c}(1-\beta)s} + 1}{1 - \mathrm{e}^{-\frac{2L}{c}s}} \mathrm{e}^{-\frac{L\beta}{c}s},$$

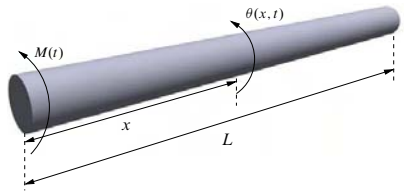


Figure 1.4: Free-free uniform rod (Example 1.4)

which includes several delay elements. This example is different from those we considered before as there are delays in both the numerator and the denominator of the resulting transfer function. This example can show that system models including delay can be very complicated. ▽

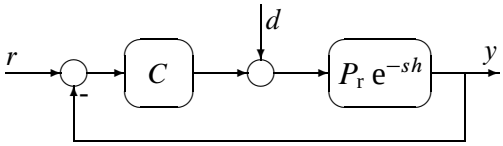


Figure 1.5: Feedback control system setup with loop delay

Loop delays can in general appear in every part of controlled plant, i.e., between the actuator and the plant, between the plant and the sensor, etc. Yet because in this thesis the discussion is mostly limited to SISO (single-input/single-output) continuous-time LTI systems, the location of the loop delay is irrelevant (as the delay always commutes with all other blocks). Hence, without loss of generality we address the setup depicted in Fig. 1.5. Here the delay element e^{-sh} is considered a part of the plant $P(s) = P_r(s)e^{-sh}$, where $P_r(s) = C(sI - A)^{-1}B$ is the rational part of the plant and $C(s)$ is a controller. In other words, we do not distinguish between sensor, actuator, etcetera delays.

1.1.1 Effects of loop delay

In this subsection some implications of the presence of the loop delay on the analysis and design of the system in Fig. 1.5 are discussed. We consider effects of the delay on the frequency response of the loop gain, closed-loop poles, and state-space description and briefly examine closed-loop performance issues.

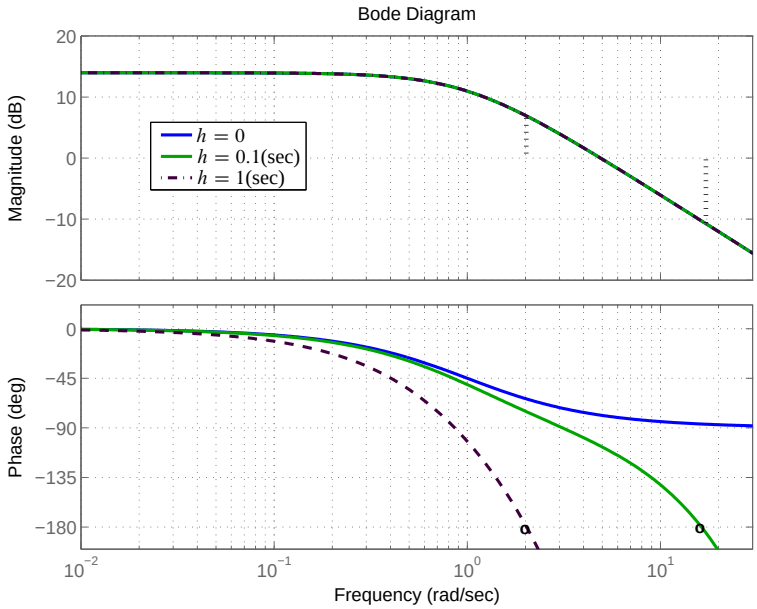
Frequency response

The loop transfer function of this system is $L(s) = P_r(s)C(s)e^{-hs}$. Denoting by $L_r := P_rC$ the rational part of the loop gain, the frequency response of $L(s)$ can be written as $L(j\omega) = L_r(j\omega)e^{-j\omega h}$. It is then readily seen that

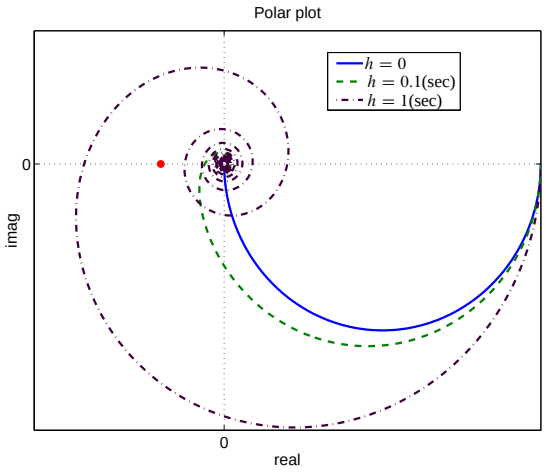
$$|L(j\omega)| = |L_r(j\omega)| \quad \text{and} \quad \arg L(j\omega) = \arg L_r(j\omega) - \omega h.$$

In other words, the DT does not affect the magnitude of $L(j\omega)$ and adds the phase lag of ωh radians to the delay-free loop frequency response.

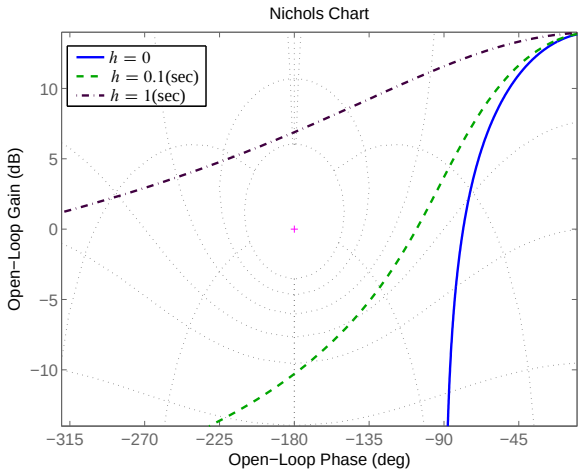
Example 1.5 *Let $L_r = \frac{5}{s+1}$. Fig. 1.6 shows the frequency responses of this plant in the delay-free case and when a delay of* 0.1 *sec* / 1 *sec is added. One can see that the phase lag provided by delay shifts the phase curve down by $\frac{180}{\pi}\omega h$ degrees and does not changes the magnitude of the curve in Bode diagram (Fig. 1.6(a)). The addition of delay rotates the frequency response curve around the origin by ωh rad on the Polar plot (Fig. 1.6(b)) and shifts the frequency response curve left by $\frac{180}{\pi}\omega h$ degrees on Nichols chart (Fig. 1.6(c)).* ▽



(a) Bode diagram



(b) Polar plot



(c) Nichols chart

Figure 1.6: Frequency response of $L_{\mathrm{r}}(s) = \frac{5}{s+1}\mathrm{e}^{-sh}$ for $h = 0, 0.1, 1$ (Example 1.5)

The simplicity of the dependence of the frequency response $L(\mathrm{j}\omega)$ on the delay makes classical frequency-domain analysis and design methods particularly well suitable for DT systems. More about this can be found in §1.1.2 below.

Pole location

The modal analysis of closed-loop systems, on the other hand, is significantly complicated by the presence of loop delay. The characteristic polynomial (or, more correctly, *quasi-polynomial*) of the closed-loop system is

$$\chi_h(s) = D(s) + N(s)\mathrm{e}^{-sh},$$

where $D(s)$ and $N(s)$ are the denominator and numerator polynomials of $L_{\mathrm{r}}(s)$, respectively. Generally, this quasi-polynomial has infinite number of roots for all $h > 0$. For example, if $L_{\mathrm{r}}(s) = K$, then

the characteristic quasi-polynomial of the closed-loop system is $\chi_h(s) = 1 + K\mathrm{e}^{-sh}$. Its roots are the solutions of

$$\mathrm{e}^{sh} = -K = K\mathrm{e}^{\mathrm{j}\pi(1+2q)}, \quad \text{for any } q \in \mathbb{Z},$$

which are

$$s = \frac{\ln K}{h} + \frac{\mathrm{j}\pi(1+2q)}{h}, \quad q \in \mathbb{Z}.$$

Thus, the closed-loop poles are equidistant points located on the vertical line with the real part at $\frac{\ln K}{h}$. In general, it might not be possible to calculate all roots of χ_h analytically. There are methods enabling one to calculate, how many roots of χ_h are located in the right-half plane of $\mathbb{C}$ (Cooke and Grossman, 1982; Marshall *et al.*, 1992; Gu *et al.*, 2003; Mirkin and Palmor, 2005). These methods, however, can only be used in the analysis to verify the stability of the closed-loop system for given P_r and C and are not readily applicable to the controller design. An approach aiming at the placement of only a subset of the closed-loop poles is proposed by Michiels *et al.* (2002).

State space representation

The state vector of a dynamical system is often defined as a "history accumulator". Indeed, when the state vector of a system at a given time is known, no prior history is required to calculate future system behavior for future input. If the state space realization of the delay-free plant P_r is given by

$$\dot{x}(t) = Ax(t) + Bu(t),$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$. The future value of $x(t+\tau)$ is calculated as

$$x(t+\tau) = \mathrm{e}^{A\tau}x(t) + \int_t^{t+\tau} \mathrm{e}^{A(\tau-\sigma)}Bu(\sigma)\mathrm{d}\sigma.$$

As we see, the value of state vector $x(t+\tau)$ at any given time $t+\tau$ can be computed using the knowledge of the initial condition at the time instance t (the first term in the right-hand side) and the "future" input in the interval $[t, t+\tau]$ (the second term). When the input delay h is present in the loop, the state space realization becomes

$$\dot{x}(t) = Ax(t) + Bu(t-h) \tag{1.1}$$

and then the value of $x(t+\tau)$ is

$$\begin{aligned} x(t+\tau) &= \mathrm{e}^{A\tau}x(t) + \int_t^{t+\tau} \mathrm{e}^{A(\tau-\sigma)}Bu(\sigma-h)\mathrm{d}\sigma \\ &= \mathrm{e}^{A\tau}x(t) + \int_{t-h}^{t} \mathrm{e}^{A(\tau-\sigma-h)}Bu(\sigma)\mathrm{d}\sigma + \int_t^{t+\tau-h} \mathrm{e}^{A(\tau-\sigma-h)}Bu(\sigma)\mathrm{d}\sigma \end{aligned} \tag{1.2}$$

The first and the third terms of (1.2) are similar to the terms in the delay-free case and depend on the initial condition and the future input, respectively. The second term of (1.2) depends on the *past* input, over the "time window" $[t, t-h)$. Thus, the knowledge of $x(t)$ and future inputs is no longer sufficient to calculate the future values of x. We also need to know past inputs in $[t-h, t]$, i.e., the initial condition $x(t)$ must be completed with $u_h(t)$, where $u_h(t) = u(t+\theta) \in L^2_{\mathbb{R}^m}[-h, 0]$, $\theta \in [-h, 0]$. This suggests that $x(t)$ is not the state vector of delayed system, since $x(t)$ does not include sufficient information to calculate the future $x(t+\tau)$ to the future input $u(t+\tau)$. The *complete state vector* of the system (1.1) at time t is $(x(t), u_h(t)) \in \mathbb{R}^n \times L^2_{\mathbb{R}^m}[-h, 0]$, this vector includes all information required to calculate the future state vector for the future input.

One of the important consequences from the state vector definition is that the state feedback $Fx(t)$ is not able to move *all poles* of the system to preassigned places in the complex plane. When $Fx(t)$ is applied to (1.1), it becomes an *output* feedback. According to the complete state vector, the "true" state feedback is an operators mapping $\mathbb{R}^n \times L^2_{\mathbb{R}^m}[-h, 0] \to \mathbb{R}^m$. One option for such an operator is the following control law

$$u(t) = F_x x(t) + \int_{t-h}^{t} F_u(t-\sigma)u(\sigma)\mathrm{d}\sigma \tag{1.3}$$

for some $F_x \in \mathbb{R}^{m\times n}$ and $F_u \in L^2_{\mathbb{R}^m}[-h, 0]$. The control law of the form (1.3) is infinite dimensional and is called a *distributed delay control law* due to the form of the second term on the right-hand side.

Achievable performance of closed-loop systems

The existence of a delay in the measurement channel implies that the controller receives "outdated" information about process behavior. Similarly, if there is a delay in the actuation channel, control action cannot be applied "on time," thus reducing the efficiency of compensation of the effect of disturbances and modeling uncertainty. These reasonings suggest that the presence of loop delays not only complicates the analysis of DT systems, but also imposes strict limitations on achievable feedback performance. In general, this is indeed true. Loop delays are shown to impose limitations on the closed-loop bandwidth (Sidi, 1997; Yaniv and Gutman, 2002), tracking performance (Su *et al.*, 2003), optimally achievable cost functions (Mirkin and Raskin, 2003; Mirkin and Tadmor, 2002), etc.

A simple example presented below shows how additional loop delay can worsen stability properties and closed-loop performance.

Example 1.6 *Consider plant $P_r = \frac{5}{s+1}$ from the previous example when the loop is closed with proportional regulator $C = k$ as shown in Fig. 1.5. Let us find the maximum value of k in three cases: $h = 0, 0.1, 1$ (sec). In the first case there is no upper bound for the maximum value of k, since gain margin is infinity in this case. It can be seen from frequency response figures that addition of delay decreases stability margins and puts restriction to maximum value of proportional regulator. When $h = 0.1$ sec, the maximum value of k is 3.3 and 0.45 in the last case, see Fig. 1.6(a). The maximum value of the proportional regulator in this case is connected to maximum achievable bandwidth of the resulting closed loop system. When in the first case any desired bandwidth can be achieved by choice of appropriate k, in the two latter cases it is bounded to 26 rad/sec in the second case and 3 rad/sec in the last one and than achievable closed loop performance (like rise time, disturbance attenuation, etc) are much worse in two last cases than in the first one.* ▽

More generally, there is a rule of thumb (Smith, 1959) saying that the gain of a proportional control law cannot exceed the ratio between the largest time constant of the system and the dead time.

Note that loop delays can in some situations improve stability and performance of closed-loop systems. Stabilizing effects of delay have been known for decades, see for example (Cooke and Grossman, 1982, p. 606) and (Marshall *et al.*, 1992, p. 27), yet more as a curious mathematical example rather than an opportunity for advanced control design. The situation started to change recently, with a trend in the control literature advocating the introduction of loop delays as an alternative to conventional methods, see, e.g., (Michiels *et al.*, 2003; Kharitonov *et al.*, 2005). These attempts, however, have not substantiated any advantage over conventional design methods yet. Hence, loop delay are treated in this thesis as an obstacle.

1.1.2 Approaches to control of DT systems

The infinite dimensionality of the delay element might also complicate design procedures. Not all design approaches are readily applicable to delay systems. In this section the application of some methods to DT

systems is discussed.

Classical methods not based on analytical model of the plant

Some design methods are not based upon an analytical model of the plant. The classical example are algorithms of tuning PID controllers, which effectively need the knowledge of only one or several points of the plant frequency response (Åström and Hagglund, 1995). For these methods the presence of loop delay does not complicate the design technique at all (although obviously imposes limitations on the achievable performance).

Another example is the family of design methods based on the loop-shaping idea (achieving a desired closed-loop performance via shaping the open-loop frequency response), including its advanced modifications like QFT (Yaniv, 1999). These methods are based on a graphical representation of the plant frequency response. As mentioned in the previous section, the effect of the delay element on the open-loop frequency response is rather simple (only additional phase lag), so these methods also can be used mutatis mutandis. The design for DT systems, however, is typically more complicated because of the additional phase lag due to the delay (see §3.1.1 for an example). In fact, one might need a high-order controller to counteract the phase lag of the delay, which could turn loop shaping highly non-trivial.

Rational approximations

One approach to circumvent the infinite dimensionality of the delay element is to approximate it by a rational transfer function. This makes it possible to use conventional methods to analyze and design controllers for DT systems. A common approach to approximate the delay element is via the following rational transfer function:

$$\mathrm{e}^{-sh} \approx P_{\mathrm{d}}(s) = \frac{p(-hs)}{p(hs)}$$

where $p(z)$ is an appropriate polynomial without zeros in the right half plane $\operatorname{Re} z \geq 0$. There are several well known choices of $p(z)$ (Richard, 2003): Laguerre-Fourier series, Kautz series, Bessel series and, perhaps the most commonly used, Padé series given by:

$$p(z) = \sum_{i=0}^{n} \frac{(2n-i)!n!}{(2n)!(n-i)!i!}(z)^i.$$

One would definitely prefer to have the order of $p(z)$ as low as possible because this would simplify analysis and design (e.g., in the root locus approach) and decrease the order of resulting regulator (e.g., in the H^∞ and H^2 approaches). Yet low-order approximations might not be sufficiently accurate.

Example 1.7 *Consider the Nichols plot in Fig. 1.7, where the frequency responses of* $P_r(s) = \frac{5}{s+1}$, $P(s) = P_r(s)\mathrm{e}^{-0.1s}$, *and the first-, second- and third-order Padé approximations of* $P(s)$ *are presented. It is seen that already at* $\omega = 1$ *rad/sec the delay cannot be neglected. Up to* $\omega \approx 1$ *rad/sec the first-order approximation is quite accurate though. Thus, if the required crossover frequency does not exceed* 1 *rad/sec, the first-order approximation may be sufficient. Yet at* $\omega = 5$ *rad/sec the first-order approximation does not reflect the phase lag of* $\mathrm{e}^{-0.1s}$. *Hence, any design that aims at reaching this crossover frequency should use at least the second-order approximation. Proceeding further, at* $\omega = 45$ *rad/sec the second-order approximation is also not accurate, so the the third-order approximation might be required to reach this level of the crossover frequency.*

Example 1.7 demonstrates that the order of the rational approximation of e^{-sh} should depend on properties of the resulting control system, like the achieved crossover frequency. Yet such properties might not be known before the controller is designed (this is true in most analytical design methods).

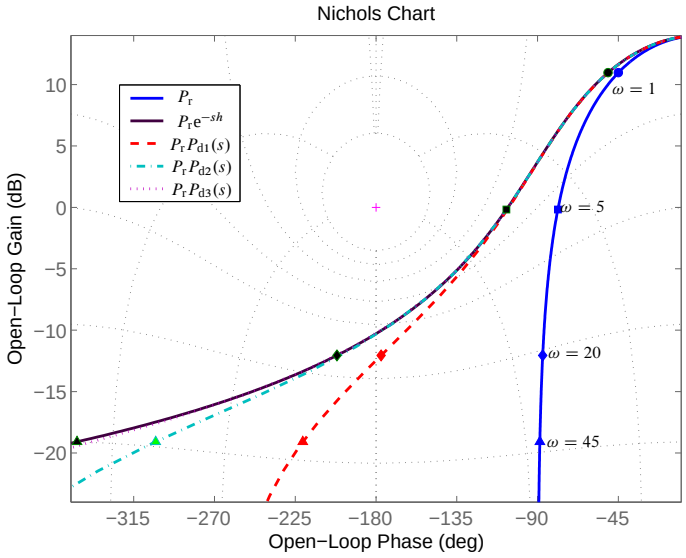


Figure 1.7: Delayed plant and its Padé approximation (Example 1.7)

Thus, the required approximation order might not be easy to pick before the design is performed. This might either require to run several iterations or lead to unnecessarily high order of the approximation.

Finite spectrum assignment

As shown in the previous section, the characteristic quasi-polynomial of DT systems has in general an infinite number of poles. Therefore, classical root locus approach cannot be applied. Manitius and Olbrot (1979) proposed a control law, based on the state vector of the delay-free plant and a *distributed-delay* processing of the control input according to the following algorithm:

$$z(t) = \int_{t-h}^{t} \mathrm{e}^{A(t-\tau)} B u(\tau) \mathrm{d}\tau. \tag{1.4}$$

As the result, they managed to assign a finite number of poles arbitrarily, while "eliminating" the others by moving their real parts to $-\infty$. This control method was called the *finite spectrum assignment* (FSA), since the resulting spectrum of closed loop is finite. The same idea was independently proposed by Lewis (1979). The distributed delay part (1.4) is actually an h time units ahead predictor of the state vector of P_r under zero initial conditions, hence the term *state predictor controller*.

The FSA method was extended to the output feedback setting by Furukawa and Shimemura (1983), who added an observer to estimate the state vector. The resulting control scheme, known as the *observer-predictor*, is depicted in Fig. 1.8. It consists of the standard (Luenberger) observer of the delayed state vector x with a gain L (such that the matrix $A_L := A + LC$ is Hurwitz), which produces the estimate $\hat{x}$, the predictor block, which includes the distributed-delay part Π acting as (1.4), which produces a prediction of $\hat{x}(t+h)$, the vector $\bar{x}(t)$, and the static state feedback gain F. As shown in (Furukawa and Shimemura, 1983), the resulting closed-loop system has a finite number of finite poles satisfying $\det(sI - A - BF)\det(sI - A - LC) = 0$, exactly as in delay-free case with the observer-based controller.

Optimal control

The application of optimal control methods—more precisely, the LQG approach— to DT systems can be traced back to '60s. The problem formulation is this case does not change. The solution, however, might

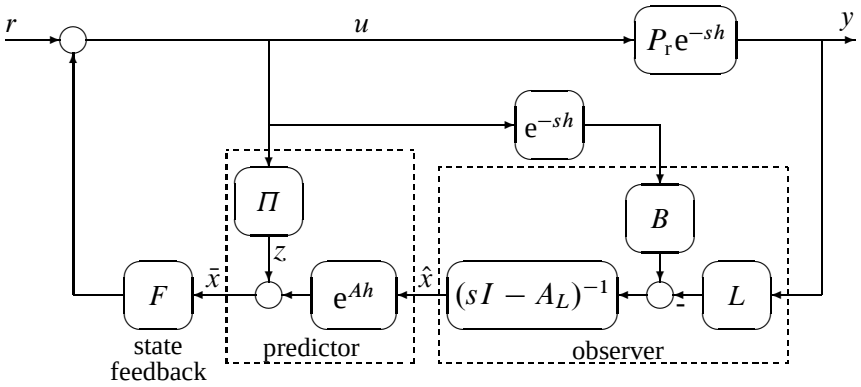


Figure 1.8: Observer-predictor control scheme

require additional efforts comparing with the delay-free case. The first solution of the LQG problem for systems with input delay is due to Kleinman (1969). The solution there is obtained in terms of two algebraic Riccati equations, exactly the same as in the delay-free LQG problem. The resulting control law in (Kleinman, 1969) is actually the observer-predictor scheme, which was then rediscovered in '80s, see Fig. 1.8. The difference from (Furukawa and Shimemura, 1983) is that the gains F and L are now the standard LQR and Kalman filtering gains, respectively, rather than chosen by pole-placement arguments.

The H^∞-optimal control of DT plants has been an active research area since mid '80s, see (Foias *et al.*, 1996; Mirkin and Tadmor, 2002) and the references therein. Most solutions obtained until 2000 involved several factorization-based stages and resulted in irrational controllers having rather complicated and non-transparent structure. Tadmor (1997a,b, 2000) then showed that the solution can be obtained in a form, closely resembling the observer-predictor configuration.

More about optimal control of DT systems can be found in §1.2.2.

1.2 Dead-time compensation

The *dead-time compensation* (DTC) approach discussed below can be regarded as custom built for DT systems. The DTC method makes use of the structure of the dead-time element to design an infinite-dimensional, yet implementable, controller, which can be designed by standard finite-dimensional methods.

1.2.1 The Smith predictor

The history of DTC began in 1957, when Otto J. M. Smith presented a new control scheme for DT systems (Smith, 1957). His control scheme became known as the Smith controller. This scheme can be presented as shown in Fig. 1.9, where the plant, consisting of a rational part $P_r(s)$ and a delay h, is controlled by a controller $C(s)$. The latter consists of a primary controller $\tilde{C}(s)$ and an internal feedback $P_r(s)(1 - e^{-sh})$ called the *Smith predictor*. The rationale behind this scheme becomes apparent when the closed-loop transfer function $T_{yr}(s)$ from r to y is considered. It can be shown (Smith, 1957, 1959) that

$$T_{yr}(s) = \frac{\tilde{C}(s)P_r(s)}{1 + \tilde{C}(s)P_r(s)} e^{-sh}, \tag{1.5}$$

i.e., the delay is pulled out of the feedback loop and does not appear in the denominator of T_{yr}. Thus, if $\tilde{C}$ is finite dimensional, so is the characteristic polynomial of T_{yr} and the stability of T_{yr} and a stabilizing $\tilde{C}$ can therefore be verified and designed by standard methods. One can also argue that the performance

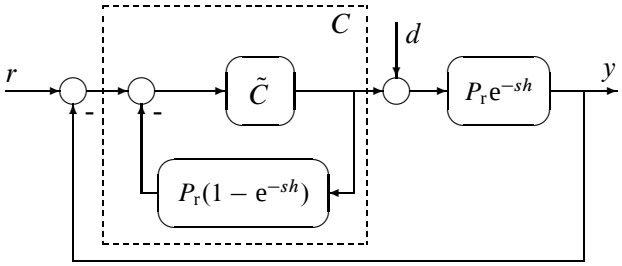


Figure 1.9: Control system setup with Smith controller

of T_{yd} can be expressed in terms of its delay-free part, so that the design of $\tilde{C}$ may be based on the rational part of the plant. This enables one to end up with the following two-stage design procedure, which constitute the core of the DTC philosophy:

(𝔄) the primary controller $\tilde{C}$ is designed for the delay free plant P_r.

(𝔅) the resulting controller is implemented by adding the Smith predictor as an internal feedback.

Thus, although the overall controller is infinite dimensional, the design procedure is completely finite dimensional, so that well understood control methods can be used in the choice of $\tilde{C}$. It is worth stressing that although $\tilde{C}$ in (𝔄) is designed for the delay-free system, it should not be designed "as if there were no delays" at all (Palmor, 1980). Rather, the loop delay imposes *implicit* constraints on the choice of $\tilde{C}$.

The operation principle of the Smith predictor is easy to see if its block diagram is redrawn as shown in Fig. 1.10. If the plant and its model are equivalent and $d = 0$, then $y \equiv \bar{y}$ and then $e = r$. Therefore

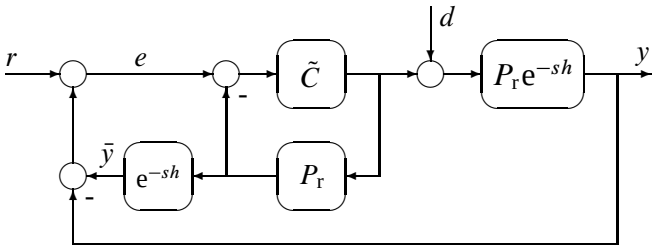


Figure 1.10: Equivalent representation of Smith controller

the closed loop transfer function T_{ry} is the cascade of the plant transfer function $P_r e^{-sh}$ and $\tilde{C}/(1 + \tilde{C} P_r)$, which is exactly Eq. (1.5).

From the frequency-response viewpoint, the Smith controller can be thought of a complicated lead network (Åström, 1977; Horowitz, 1983). Phase lead is provided by the inner regulator feedback including the plant model, see Fig. 1.9.

The idea of Smith might considerably mitigate the design procedure for delayed system. This scheme thus gave rise to profound interest of researchers, both in academia and in industry. This interest was substantially increased with the appearance of inexpensive microprocessor based digital equipment, which enables an easy implementation of such control law. The idea of Smith (1957, 1959) had a strong impact on both theory and practice of control of time-delay systems. Over the years, numerous studies of the properties of the Smith controller and its modifications and extensions have been carried out, hundreds articles and technical reports have been written, see (Palmor, 1996) and the references therein. Even now, the Smith predictor helps to solve challenging problems in various fields of applied control engineering,

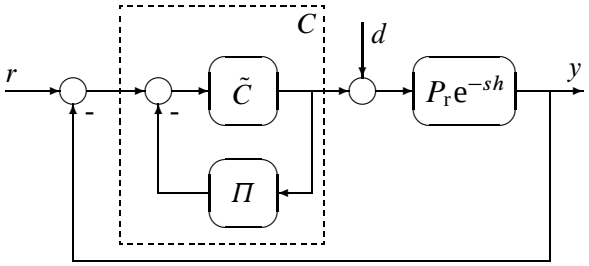


Figure 1.11: General DTC setup

e.g., in medicine (Reboldi *et al.*, 1991), wireless communication (Lee *et al.*, 2004), computer networks (Mascolo, 2000), etc.

1.2.2 Modifications and alternatives

Modified Smith predictors

One disadvantage of the Smith controller is that it can only be applied to stable plants, since the closed-loop characteristic equation contains all poles of $P_{\rm r}$, see (Palmor, 1980). This problem, however, can be overcome by modifying the predictor block. Let us denote the dead-time compensation (predictor) block in general form as

$$\Pi(s) = \tilde{P}(s) - \hat{P}(s)\mathrm{e}^{-sh}, \tag{1.6}$$

see Fig. 1.11. Modified (or generalized) Smith predictor (MSP) refers to the scheme when $\hat{P} = P_{\rm r}$ and $\tilde{P}$ is a rational transfer function such that the resulting Π is stable, i.e. belongs to H^∞. It is known (Curtain *et al.*, 1996; Mirkin and Raskin, 2003) that if $\Pi \in H^\infty$, then the internal stability of the system in Fig. 1.9 is equivalent to that of the delay-free system consisting of the plant $\tilde{P}$ and the controller $\tilde{C}$. Thus the stage (ℜ) above should be replaced with

(ℜ$_1$) the primary controller $\tilde{C}$ is designed for the delay-free auxiliary plant $\tilde{P}$

(rather than for $P_{\rm r}$ as in the Smith controller case). When $P_{\rm r}$ is stable, $\tilde{P}$ can be any stable transfer function. Yet one should pay attention to the fact that the central regulator is designed for closed-loop transfer function $T_{yr} = \frac{\tilde{C}P_{\rm r}}{1+\tilde{C}P_{\rm r}}$, whereas the resulting closed-loop transfer function is $T_{yr} = \frac{\tilde{C}P_{\rm r}}{1+\tilde{C}\tilde{P}}$, so that the choice of $\tilde{P}$ does affect the resulting closed-loop system. The best known options for $\tilde{P}$ are:

- $\tilde{P} = P_{\rm r}$ resulting in the Smith predictor case;
- $\tilde{P} = 0$ resulting in the internal model controller configuration (Morari and Zafiriou, 1989).

For unstable $P_{\rm r} = C(sI - A)^{-1}B$ the choice is less evident, yet the required $\tilde{P}$ can always be found:

- $\tilde{P} = C\,\mathrm{e}^{-Ah}(sI - A)^{-1}B$ is an admissible choice for any $P_{\rm r}$, because in this case

$$\Pi(s) = C(\mathrm{e}^{-Ah} - \mathrm{e}^{-sh})(sI - A)^{-1}B = C\,\mathrm{e}^{-Ah}\int_0^h \mathrm{e}^{-(sI-A)\sigma}\mathrm{d}\sigma B \in H^\infty$$

  is an entire function such that $\zeta(t) = \Pi u(t) = Cz(t)$, where $z(t)$ is defined by the distributed-delay law (1.4), see (Mirkin and Palmor, 2005);

- $\tilde{P} = -C\int_0^h \mathrm{e}^{-A\eta}\mathrm{d}\eta + C\,\mathrm{e}^{-Ah}(sI - A)^{-1}B$ insures that $\Pi(0) = 0$ and then $C(0) = \tilde{C}(0)$, which is useful for the design of servo-controllers, see (Watanabe and Ito, 1981; Mirkin and Zhong, 2003).

There is a number of other approaches to the choice of $\tilde{P}$, both general (Zwart and Bontsema, 1997; Meinsma and Zwart, 2000) and custom built for some specific processes (Åström *et al.*, 1994; Mataušek and Micić, 1999; Normey-Rico and Camacho, 2002).

Another modification of the Smith controller was proposed by Palmor and Powers (1985) for the situation when the disturbance signal is measurable. In this case d should be injected into the predictor.

Connection between the MSP and the observer-predictor schemes

The DTC (Fig. 1.11) and observer-predictor (Fig. 1.8) schemes had been developed independently for about two decades and regarded as different schemes. It turns out, however, that they are actually equivalent. The equivalence can be shown by a simple similarity transformation (Mirkin and Raskin, 2003). More precisely, the observer-predictor is equivalent to the MSP when $\tilde{P}$ is chosen as $\tilde{P} = C\,\mathrm{e}^{-Ah}(sI - A)^{-1}B$ and the primary controller is the standard observer based feedback law for this $\tilde{P}$. The principal difference between these two schemes is that the MSP predicts the output of the delayed plant, whereas the observer-predictor controller predicts the state vector.

DTCs in optimal control

An important observation about DTC configurations is that they arise naturally in several optimal control problems for dead-time systems. This turns out to be true in the H^2 (LQG) (Kleinman, 1969; Mirkin and Raskin, 2003; Mirkin, 2003), H^∞ (Başar and Bernhard, 1991; Tadmor, 2000; Meinsma and Zwart, 2000; Mirkin, 2003), and L^1 (Mirkin, 2006) cases. The DTC blocks in the H^2 and L^1 cases coincide actually with the modified Smith predictor $\Pi = \tilde{P} - P_{\mathrm{r}}\mathrm{e}^{-sh}$ in (Watanabe and Ito, 1981). This DTC attempts to compensate the loop delay assuming that there are no disturbances in the system. The H^∞ DTC is more complicated, with different (and necessarily unstable) $\tilde{P}$ and $\hat{P}$ systems in (1.6). As shown in (Mirkin, 2003), the H^∞ DTC "predicts" not only the plant output, but also the *worst-case* disturbance for the open loop H^∞ problem. The predicted worst-case disturbance is then used in the controller similarly the use of the measurable disturbance is (Palmor and Powers, 1985).

The fact that optimization problems produce DTC-based control laws is important. It suggests that dead-time compensation is not just one of many possible heuristic approaches simplifying controller design, but rather is a fundamental concept in control of dead-time systems.

1.2.3 Some DTC shortcomings

As discussed above, the DTC configuration simplifies the overall controller design, and is, in a sense, optimal from H^2, H^∞, and L^1 points if view. Moreover, DTCs are argued to be an integral part of some biological systems, see (Miall *et al.*, 1993). Nevertheless, the DTC controller structure is sometimes believed to have intrinsic shortcomings, such as weak disturbance attenuation properties, poor robustness, and implementation difficulties. It should be emphasized that the sources of some of these shortcomings are empirical and their existence is a part of the control folklore rather than the outcome of a rigorous analysis, see the discussion in (Morari and Zafiriou, 1989) for example.

Regarding disturbance attenuation and robustness to (unstructured and complex structured) plant uncertainty, it is probably safe to say that the use of the DTC structure actually improves these characteristics of feedback systems, rather than harms them. This follows from the aforementioned facts, that optimal disturbance attenuation and robust stability problems are solved by DTC-based controllers. It appears that the reported problems are mostly caused by an improper design of the primary controller in two-stage methods, where one may be tempted to impose design requirements without accounting for the presence of the loop delay. This might result in non-robust overall system and fall short of the disturbance attenuation performance in the first stage.

The situation with the potentially poor robustness of DTCs to uncertainties in the loop delay and problems with the implementation of DTCs for unstable systems is yet to be clarified.

Delay robustness of DTCs

Arguably, DTC-based controllers are most frequently criticized on the ground of their potentially *poor robustness* with respect to *uncertainty in the delay*. A classical example is the result of (Palmor, 1980), where it is shown that under a seemingly reasonable Smith predictor design the loop might be destabilized by an arbitrarily small delay mismatch (termed *practically unstable* system).

In principle, robustness of DTC with respect to delay uncertainty can be analyzed along the same line by covering delay uncertainty with a frequency weighted (either unstructured or structured) multiplicative uncertainty (Wang *et al.*, 1994; Mirkin and Palmor, 2005). Such an approach, however, might lead to a rather conservative design. It is therefore important to have more problem-oriented tools to address the sensitivity of DTC to delay variations explicitly. For the *calculation* of the delay margin of DTCs some problem-oriented method are available in the literature, e.g., (Palmor, 1980; Furutani and Araki, 1998; Michiels and Niculescu, 2003). These methods, however, do not *explain* why DTC-based controllers may become sensitive to delay mismatch and whether this sensitivity is intrinsic to the method.

It was pointed out in (Horowitz, 1983), by means of numerical examples, that design of high gain primary controller in the Smith predictor results in non-robust closed-loop systems, including high sensitivity to delay uncertainty. Horowitz (1983) also noticed that the Smith predictor produces a loop with many crossover frequencies, yet did not develop this point. Georgiou and Smith (1992) faced with a similar situation analyzing optimal controllers in the gap metric yet did not connect this to the dead-time tolerance deterioration. The link between large number of crossover frequencies and considerable deterioration of delay margin was pointed out by Adam *et al.* (2000). Also it was shown that existence of many crossover frequencies can bring to a *discontinuity* of the delay margin as a function of the system parameters. The paper (Adam *et al.*, 2000) addresses the delay robustness of the classical Smith predictor and mostly emphasizes algorithmic aspects of this analysis. Yet there is no generic explanation of this phenomenon in the context of the dead-time compensation.

Implementation of DTCs

There emerges an increasing number of applications, where the possibility to meet performance requirements are challenged by the presence of loop delay. Therefore, there is not only theoretical but also significant practical importance for the DTC-based controllers. From the practical point of view, implementation issues are an essential part of overall study of DTCs.

Because there is a large number of processes including dead time in chemical industry, one of the first practical uses of the control scheme based on the Smith predictor was implemented there (Lupfer and Oglesby, 1962). Although satisfactory results of the resulting control system were reported, the implementation of the plant model (including loop delay approximated by rational transfer functions) required a significant effort. The situation has changed with the development of digital equipment, so already in the early '80s some microprocessor-based industrial process controllers offered the DTC as a standard algorithm alongside the PID (e.g., (Veronesi, 2003)), allowing much simpler implementation of DTC control algorithms. This fact shows a high demand on this type of control systems in industry. Moreover, there are efficient tuning algorithms (Palmor and Shinnar, 1981; Palmor and Blau, 1994), which further extend the scope of applicability of the DTC.

As far as the classical Smith predictor is considered, the predictor block $\Pi = P_\mathrm{r}(1 - \mathrm{e}^{-sh})$ is readily implemented digitally as the difference of two transfer functions P_r and $P_\mathrm{r}\mathrm{e}^{-sh}$. This approach, however, is not applicable to the modified Smith predictor, $\Pi = \tilde{P} - P_\mathrm{r}\mathrm{e}^{-sh}$, in the case when P_r is unstable and to

the H^∞ DTC, $\Pi = \tilde{P} - \hat{P}\,\mathrm{e}^{-sh}$, in which case $\tilde{P}$ and $\hat{P}$ are always unstable. Indeed, in these situations the difference $\tilde{P} - \hat{P}\,\mathrm{e}^{-sh}$ necessarily contains unstable pole/zero cancellations. If these cancellations are performed numerically, the closed-loop system is just not internally stable. The implementation of such DTCs must therefore be based on the analytical cancellations of unstable poles and zeros of Π before it is implemented. This leads to the need to implement distributed-delay algorithms like that in (1.4).

A discretization algorithm for the implementation of DD elements was mentioned already by Manitius and Olbrot (1979), yet the issue did not draw much attention until the late '90s. The first investigation of this problem was (van Assche *et al.*, 1999), where the implementation of DD control laws via a lumped-delay approximation (discretization) procedure was addressed through a numerical example. The example demonstrated numerical instability irrespective of the discretization step. This result, see also its follow-ups (Mondié *et al.*, 2001a; Mondié and Michiels, 2003), motivated Richard (2003) to pose the problem of the reliable numerical implementation of DD controllers as one of important open problems in the control of time-delay systems. It turns out, however, that the reported problems are associated with a poor approximation method and are not inevitable, see (Mirkin, 2004). This paper proposes then a numerically stable lumped-delay approximation method. An alternative approach is to use the difference between $\tilde{P}$ and $\hat{P}\,\mathrm{e}^{-sh}$ complemented by a resetting mechanism as proposed by Mondié *et al.* (2001b). See Section 3.2 for more details. These developments, however, are purely theoretical. At the same time, it is of great interest to have a practical validation of the existing implementation schemes.

1.3 The goals of this research

Currently available methods of the design of DTCs do not provide comprehensive answers to the questions of the robustness with respect to delay mismatches. In many applications loop delays are uncertain or vary, so that DTCs should be robust to such kind of uncertainty. By the delay margins we understand maximal deviations (in both directions) of the actual plant delay for which the closed loop-system with a controller designed for a nominal value of the delay remains stable. As pointed out above, although there exist several methods for computing the delay margins, a little is known about delay sensitivity of DTC-based controllers. The purposes of this work are

- to study the mechanisms by which DTCs might become excessively sensitive to modeling uncertainties,
- to suggest possible remedies to overcome these problems, and

The robustness of DTC schemes will be studied in the framework of the H^∞ theory. Toward this end, recent solutions of the optimal control problems for time delay systems that result in "optimal" DTCs will be extensively used. These controllers include FIR blocks with unstable zero-pole cancellation. Implementation of such controllers has not been studied in details yet. The goals of the second part of this work are

- to study existing approaches to the numerical implementation of DTC-based controllers and
- to validate these approaches by laboratory experiments.

1.4 Organization of the thesis

This thesis is organized as follows:

Chapter 2 addresses possible reasons for the high sensitivity of DTCs to delay uncertainties. The dead-time tolerance of DTCs is first studied for the plant containing an integrator and a dead time. The

proposed analysis is then extended to more general plants using such tools of the classical loop shaping as the M and N circles and the Bode's gain-phase relation. Finally, somewhat more conscious guidelines for the design of DTC-based controllers are proposed.

Chapter 3 studies existing approaches to the numerical implementation of DTC-based controllers: resetting mechanism (RM) and lumped-delay approximation (LDA) approaches. The main emphasis in this chapter is placed on the second approach. It is shown that the out-of-the-box use of this approach might not be feasible. The reasons for this are presented and remedies are proposed. The proposed solutions are validated by laboratory experiments.

Chapter 4 contains concluding remarks.

Appendix A is devoted to the laboratory experiment used for the experimental validation of the proposed solutions in Chapter 3. Also, it presents the modeling of the experimental setup.

Appendix B describes some details about the Matlab implementation of DTCs control laws. It contains detailed explanations about the Simulink implementation the RM approach and auxiliary Matlab functions for both LDA and RM methods.

The results of Chapter 2 were presented at the 5th IFAC Workshop on Time Delay Systems, K.U. Leuven, Belgium, September 2004, and a paper based on these results was published in the *International Journal of Control* (vol. 80, no. 2, pp. 1316–1332, 2007).

Chapter 2

Delay margin of DTCs

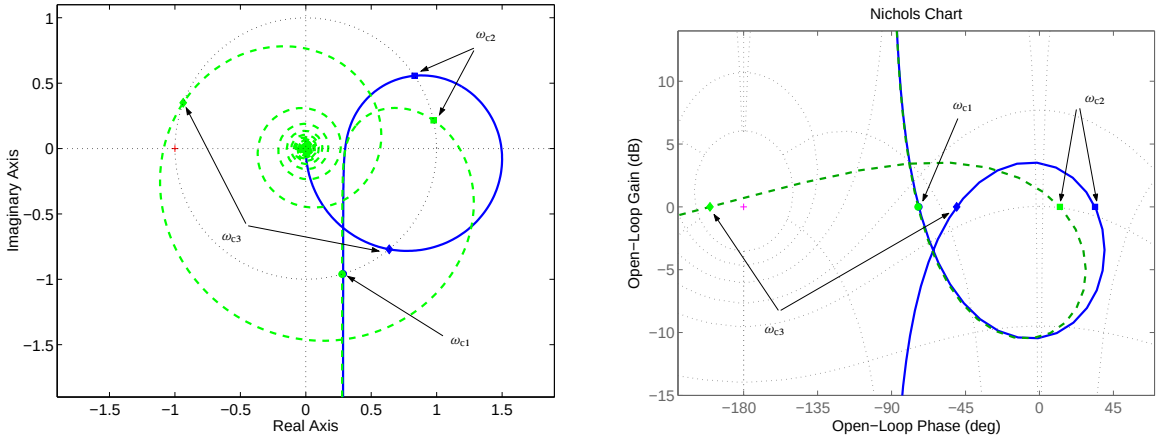
As mentioned in the previous chapter, underlying mechanisms by which DTC-based controllers might become excessively sensitive to delay uncertainty are still obscure. The purpose of this chapter is to reveal them. It will be shown that these mechanisms can be understood by using the standard Nyquist criterion reasoning. In particular, it will be shown (both by analyzing a simple yet representative special case and by general arguments) that DTC-based controllers tend to introduce multiple crossover frequencies (this feature was called the *crossover proliferation* phenomenon). This, in turn, leads to a substantial deterioration of the robustness of the resulting system to delay mismatch. A contribution of this chapter is in revealing the link between the existence of the crossover proliferation phenomenon and the sensitivity of DTCs to delay uncertainty and understanding how generic the crossover proliferation phenomenon is in the context of the dead-time compensation. The resulting conclusions lead to somewhat more consciousness guidelines for the design of DTC-based controllers (see Section 2.4).

The chapter is organized as follows. In Section 2.1 preliminary results on the definition of the delay margin notion are collected. In Section 2.2 we consider the design of simple DTC controllers for the plant containing an integrator and a dead time. This is perhaps the simplest nontrivial special case, yet it does capture many important aspects of the problem and, as such, is frequently used as a benchmark problem in the analysis of DTCs. We consider both the classical two-stage design (§2.2.1) and the H^∞ loop-shaping design (§2.2.2). Some generalizations and interpretations are then discussed in Section 2.3 and possible design guidelines are put forward in Section 2.4. Section 2.5 collects lengthly technical derivations for the results of Section 2.2.

2.1 Preliminary: delay margin (dead-time tolerance)

The purpose of this section is to review the notion of the delay margin and to emphasize some of its aspects relevant to the discussion of the delay sensitivity of DTC-based controllers. These aspects are definitely not new, although it might be difficult to pinpoint them in the literature.

Stability margins are one of the key characteristics of the classical methods based on the analysis of the frequency response of the loop gain, $L(\mathrm{j}\omega)$. Widely used stability margins are the gain margin, μ_g, which reflects the sensitivity to gain variations, and the phase margin, μ_ph, which reflects the sensitivity to phase variations. Although loop delays affect only the phase of the open-loop transfer function, the phase margin might be a poor measure of the robustness against delay variations. The reason is that the phase lag due to the delay depends on the frequency at which μ_ph is measured (called the crossover frequency, ω_c, and defined as the frequency at which the loop gain is 0 dB), whereas the phase margin notion does not take the crossover frequency into account. This calls for the introduction of a new stability margin, called the *delay margin* or *dead-time tolerance*. The delay margin, μ_d, is defined as the smallest delay variation



(a) Nyquist diagram (b) Nichols chart

Figure 2.1: Plant with several crossover frequencies: $L(s)$ (solid line) and $L(s)\mathrm{e}^{-0.5s}$ (dashed line)

destabilizing the system.

It is somehow conventional in the control literature to refer to the delay margin as the following quantity:

$$\mu_{\mathrm{d}} = \frac{\mu_{\mathrm{ph}}}{\omega_{\mathrm{c}}}. \tag{2.1}$$

Yet there are two important cases, especially from DTC analysis perspectives, for which (2.1) falls short of reflecting the delay margin. These are the cases when the high-frequency loop gain is not contractive and when there are more than one crossover frequencies.

- The condition $\lim_{\omega\to\infty}|L(\mathrm{j}\omega)| < 1$ or, more precisely, $\limsup_{\omega\to\infty}|L(\mathrm{j}\omega)| < 1$ is necessary for the closed-loop system be robust against arbitrarily small high-frequency modeling errors. The recognition of this fact can be traced back to (Willems, 1971; Barman *et al.*, 1973; Palmor, 1980) and the general proof is due to Georgiou and Smith (1993). Thus, if the condition above does not hold, e.g., for $L(s) = \frac{2s+1}{s+2}$, the delay margin is zero irrespective of μ_{ph} and ω_{c} since any loop delay destabilizes the system.

- When there are several crossover frequencies, (2.1) can also fail. To see this consider the following loop transfer function:

$$L(s) = \frac{6(s^2 + 0.2s + 0.01)}{s(s+2)^2}.$$

  The Nyquist and Nichols plots of this L are presented in Fig. 2.1 by the solid lines. This system has three crossover frequencies: $\omega_{\mathrm{c}1} = 0.0154$, $\omega_{\mathrm{c}2} = 0.746$, and $\omega_{\mathrm{c}3} = 5.24$. The phase margin here is measured at $\omega_{\mathrm{c}1}$, so that the delay margin according to (2.1) should be $\mu_{\mathrm{d}} > \frac{\pi}{2\cdot 0.0154} \approx 102$. Yet this conclusion is erroneous. This is clearly seen in the frequency-response plots of $L_{\mathrm{r}}(s)\mathrm{e}^{-0.5s}$ (dashed lines in Fig. 2.1). These curves do encircle the critical point, i.e., the destabilizing delay is actually < 0.5 sec. The reason is that the phase lag due to the delay at the largest crossover frequency, $\omega_{\mathrm{c}3}$, is much larger than that at $\omega_{\mathrm{c}1}$. Hence, when the delay is increased $L_{\mathrm{r}}(\mathrm{j}\omega_{\mathrm{c}3})$ reaches the critical point long before $L_{\mathrm{r}}(\mathrm{j}\omega_{\mathrm{c}1})$ does, even though the phase distance from the critical point in the latter case is smaller.

These situations should thus be reflected in the expression of μ_{d} in terms of the loop frequency response $L(\mathrm{j}\omega)$.

Another aspect of the definition of the delay margin, which is important in the analysis of DTCs, is related to the direction in which the delay changes. It is quite common to consider mainly the sensitivity to the increase of the loop delay. Yet this assumption is justifiable only when the nominal system is delay-free (so the negative direction makes no sense) or when only this direction is destabilizing. As will be shown in the next section, this is not what happens in systems containing DTCs, where both direction of the delay variation might be destabilizing. It is therefore important to account for the negative delay variations (delay decrease) as well.

To this end, we define two delay margins: μ_{d}^{+} and μ_{d}^{-}, which reflect the sensitivity to the increase and decrease of h, respectively. Following the discussion above, they are quantified as follows:

$$\mu_{\mathrm{d}}^{+} = \begin{cases} \min \dfrac{\mu_{\mathrm{ph},i}^{+}}{\omega_{\mathrm{c}i}} & \text{if } \limsup\limits_{\omega\to\infty} |L(\mathrm{j}\omega)| < 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.2a}$$

and

$$\mu_{\mathrm{d}}^{-} = \begin{cases} \max\left\{-h, \dfrac{\mu_{\mathrm{ph},i}^{-}}{\omega_{\mathrm{c}i}}\right\} & \text{if } \limsup\limits_{\omega\to\infty} |L(\mathrm{j}\omega)| < 1 \\ 0 & \text{otherwise,} \end{cases} \tag{2.2b}$$

where $\omega_{\mathrm{c}i}$ are crossover frequencies and $\mu_{\mathrm{ph},i}^{+} \geq 0$ and $\mu_{\mathrm{ph},i}^{-} \leq 0$ are corresponding angular distances of $L(\mathrm{j}\omega_{\mathrm{c}i})$ to the critical point $(-1, 0)$ on the Nyquist plot measured in the clockwise and counter-clockwise directions, respectively ("phase margins"). Note that the "max" operator in definition (2.2b) is used to take into account that loop delays in causal systems can be only nonnegative.

Having the quantities in (2.2), the system with a nominal delay h remains stable for all delays in the range $(h - \mu_{\mathrm{d}}^{-}, h + \mu_{\mathrm{d}}^{+})$. It should be noted that although the system will become unstable at each end of this interval (unless, possibly, $\mu_{\mathrm{d}}^{-} = h$), there might exist other stability intervals outside the interval above. Yet, arguably, these additional intervals are less important for the robustness analysis.

2.2 Control of an integrator and dead-time

In this section we consider the DTC controller design for the dead-time plant

$$P(s) = \frac{k_p}{s}\mathrm{e}^{-sh}, \qquad k_p > 0. \tag{2.3}$$

This system is simple enough to enable us to end up with closed-form solutions, while it still captures the essence of the problem and for this reason is frequently used as a benchmark problem for control of dead-time systems.

We consider below two different approaches to the DTC design for (2.3): the classical two-stage design (Mirkin and Palmor, 2005) with a static (proportional) primary controller and a direct H^{∞} loop-shaping design, which also results in a DTC configuration with a static primary controller. While the former procedure, which is a *de facto* standard in the design of DTCs, might be regarded as heuristic, the latter design is a result of a rigorous analytic procedure.

2.2.1 Two-stage design with a static primary controller

As discussed in Section 1.2, the essence of the conventional two-stage DTC design is to design a primary controller for some rational $\tilde{P}$ such that $\Pi = \tilde{P} - P_{\mathrm{r}}\mathrm{e}^{-sh}$ is stable and then implement the resulting $\tilde{C}$

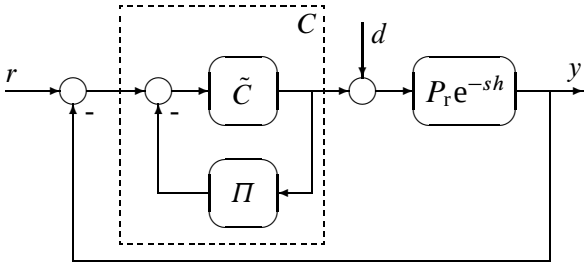
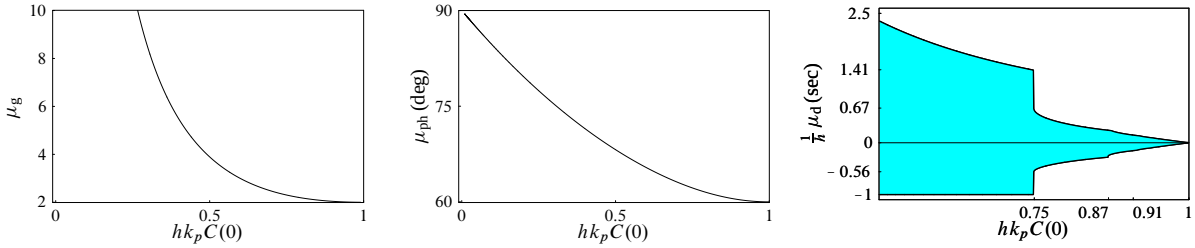


Figure 2.2: General DTC setup



(a) $\mu_{\rm g}$ vs. $hk_pC(0)$ (b) $\mu_{\rm ph}$ vs. $hk_pC(0)$ (c) $\mu_{\rm d}$ vs. $hk_pC(0)$

Figure 2.3: Stability margins vs. normalized controller gain $hk_pC(0)$ (two-stage design)

as shown in Fig. 2.2. Although $P_{\rm r}$ is not stable, the choice $\tilde{P} = P_{\rm r}$ still guarantees the stability of the DTC block Π when the latter is implemented as a distributed delay system (Mirkin and Palmor, 2005). This choice, however, has a disadvantage that when $\tilde{C}(s) = k_c$, the static gain of the overall controller is not k_c, so that the high-gain primary controller does not necessarily result in a high-gain overall controller C. To circumvent this inconvenience, we use the modification proposed by Watanabe and Ito (1981) and choose

$$\tilde{P}(s) = \frac{k_p(1 - sh)}{s},$$

which still results in a stable Π. Yet in this case $\Pi(0) = 0$ and hence $C(0) = \tilde{C}(0) = k_c$. It is worth stressing that this modification does not impose any limitation of the resulting C. In fact, it is a matter of an internal loop shifting in C to transform any controller with a conventional Smith predictor to a corresponding controller with $\tilde{P}$ as above.

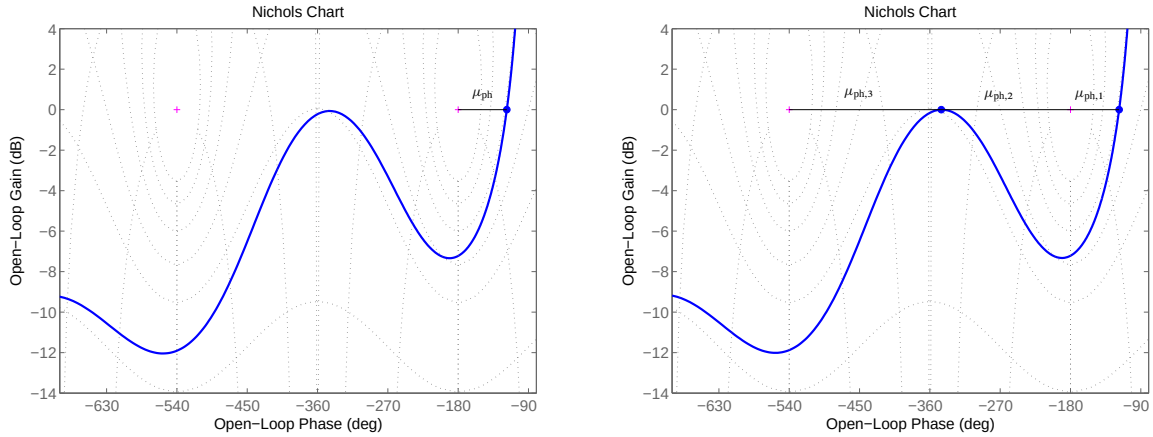
The primary controller should now be designed for the delay-free plant $\tilde{P}$. The characteristic polynomial of this system is $\chi_{\rm cl}(s) = (1 - k_ck_ph)s + k_ck_p$. Thus, the closed-loop system is stable iff

$$0 < k_c < \frac{1}{hk_p}$$

(at $hk_pk_c = 1$ the DTC internal loop is not well posed) and the larger k_c is, the higher the loop gain is and the faster the closed-loop response is (the closed-loop transfer function from r to y is now $T = e^{-sh}/((1/k_ck_p - h)s + 1)$).

One would expect that the stability margins deteriorate as k_c approaches its upper bound $1/(hk_p)$. This is indeed true as can be seen in the plots in Fig. 2.3, where the stability margins versus the normalized controller gain $hk_pk_c \in (0, 1)$ are depicted; see Section 2.5.1 for details of the derivation. Moreover, the gain and phase margins are smooth continuous functions of k_c. More surprising is the behavior of the

delay margin. One can see that at $k_c \approx \frac{0.75}{hk_p}$ the plot has a discontinuity where μ_d deteriorates dramatically by more than a factor of 2.



(a) $k_c = \frac{0.748}{hk_p}$ (b) $k_c = \frac{0.749}{hk_p}$

Figure 2.4: Nichols charts of $L(s)$ at the first crossover proliferation

To explain this phenomenon, consider the loop transfer function $L = P_\mathrm{r} C\,\mathrm{e}^{-sh}$ of the resulting system. It is readily verified that

$$L(s) = \frac{k_c k_p\,\mathrm{e}^{-sh}}{s + k_c k_p(1 - sh - \mathrm{e}^{-sh})}\,. \tag{2.4}$$

The source of the intriguing behavior of μ_d in Fig. 2.3(c) becomes apparent when the Nichols plot of the open loop for $hk_pk_c = 0.748$ and $hk_pk_c = 0.749$ are compared, see Fig. 2.4. The magnitude plots in both cases are "oscillatory", with a sequence of local resonant peaks. When $hk_pk_c = 0.748$, all these peaks lie below the 0 dB level, so that there is only one crossover frequency $\omega_\mathrm{c} = 0.773/h$ as shown in Fig. 2.4(a). The μ_d^+ is then calculated as

$$\frac{\mu_\mathrm{ph}}{\omega_\mathrm{c}} = \frac{1.086}{0.773/h} = 1.404h,$$

and μ_d^- is obviously $-h$ as the decrease of the loop delay is not destabilizing.

A small increase of k_c changes this situation dramatically as the first resonance crosses the 0 dB level. The system in Fig. 2.4(b) has now three crossover frequencies: $\omega_{\mathrm{c}1} = 0.774/h$ and $\omega_{\mathrm{c}2} \approx \omega_{\mathrm{c}3} = 5.108/h$ and both the increase and the decrease of the delay may be destabilizing (which explains the discontinuity of the negative margin in Fig. 2.3(c)). The positive normalized delay margin should then be calculated as the minimum over

$$\frac{\mu^+_{\mathrm{ph},1}}{\omega_{\mathrm{c}1}h} = \frac{1.086}{0.774} = 1.403 \quad \text{and} \quad \frac{\mu^+_{\mathrm{ph},3}}{\omega_{\mathrm{c}3}h} = \frac{3.408}{5.108} = 0.667.$$

Since $\omega_{\mathrm{c}1}$ is considerably smaller than $\omega_{\mathrm{c}3}$, the positive margin is calculated at the latter point even though the phase distance to the critical point there is larger than that at $\omega_{\mathrm{c}1}$. This explains the discontinuity in Fig. 2.3(c).

With the further increase of k_c the number of crossover frequencies continues to increase (see Fig. 2.5). The μ_d's might then be calculated at higher crossover frequencies, so that they continue to deteriorate, although not always discontinuously. The latter can be explained by the fact that the μ_ph's at the points of the touch with the 0 dB level are quite large (around $\pm\pi$), which might compensate, for a short while, the increase of the crossover frequencies.

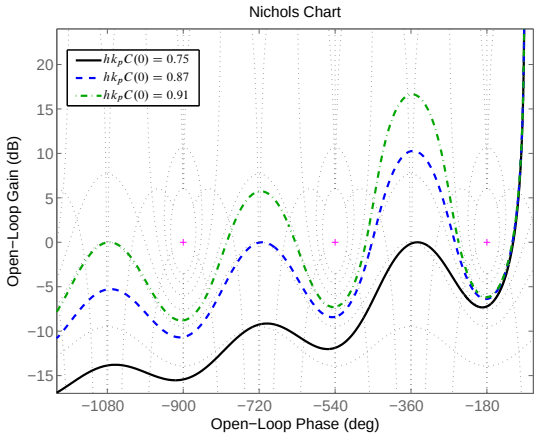


Figure 2.5: Nichols charts of $L(s)$ for different crossover proliferation (two-stage design)

When $hk_pk_c \to 1$, the loop transfer function approaches

$$L_\alpha(s) := \frac{\mathrm{e}^{-sh}}{1-\mathrm{e}^{-sh}} \tag{2.5}$$

It can be shown that the Nichols plot of $L_\alpha(\mathrm{j}\omega)$ is a discontinuous curve going along with the 0 dB M-circle. For this curve $\mu_\mathrm{g} = 2$ and $\mu_\mathrm{ph} = 60°$, which are reasonably large. On the other hand, the $\limsup|L(\mathrm{j}\omega)|$ is infinity, so that the delay margins in this case are $\mu_\mathrm{d}^+ = \mu_\mathrm{d}^- = 0$ by (2.2).

Remark 2.1 *It is worth emphasizing that the proliferation of crossover frequencies does not occurs when only the nominal delay changes. This is apparently the reason why the discontinuity of μ_d did not show up in earlier studies of the delay margin of DTCs, where the analysis was based on varying the nominal delay, see (Furutani and Araki, 1998; Michiels and Niculescu, 2003). In that case the number of the crossover frequencies cannot change.*

Remark 2.2 *Note that the positive and negative delay margins (μ_d^+ and μ_d^-) in this example are not equivalent, even after the first discontinuity point where μ_d^- becomes meaningful. There are intervals where $\mu_d^+ > \mu_d^-$ and vice versa. This fact justifies the introduction of two delay margins.*

2.2.2 H^∞ loop shaping

An alternative to the two-stage design procedure discussed above is the use of direct optimization-based approaches. In this subsection we consider the application of the H^∞ loop-shaping procedure of McFarlane and Glover (1990). The procedure is based on the maximization of the robustness radius against normalized coprime factor uncertainty (robustness in the gap metric) for the weighted plant WP. The weighting function $W(s)$ is chosen on the basis of the classical gain loop-shaping arguments and the robust stability stage actually attempts to increase stability margins without much altering the shape of $|WP(\mathrm{j}\omega)|$.

We choose to use the simplest possible weight, $W(s) = k_w$. The increase of k_w can be interpreted as requiring a higher loop gain or, alternatively, a higher bandwidth. In this case the robustness radius is maximized for $WP = \frac{k_w k_p}{s}\mathrm{e}^{-sh}$. This H^∞ optimization (see Section 2.5.2 for details) results then in a controller having the DTC structure like that depicted in Fig. 2.6 with

$$\tilde{C}(s) = \frac{k_w}{\eta} \quad \text{and} \quad \Pi(s) = k_p \frac{2\eta s + k_w k_p \eta(1-\eta^2) - (\eta^2+1)s\mathrm{e}^{-sh}}{s^2 + (k_w k_p \eta)^2},$$

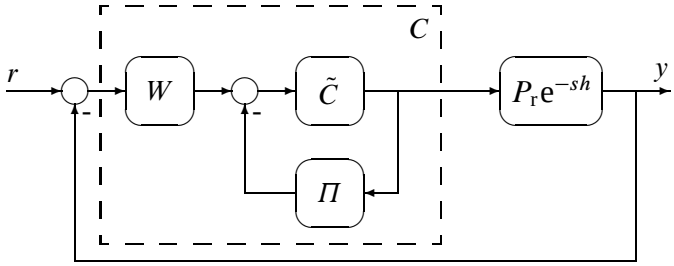


Figure 2.6: Control system setup for H^∞ loop shaping

where $\eta \in (0, 1]$ is the (unique) solution of the transcendental equation

$$2\tan\bigl(hk_pk_w\eta\bigr) = 1/\eta - \eta$$

(η is a monotonically decreasing function of hk_pk_w). It can be verified that the zeros of the denominator of Π at $\pm \mathrm{j}k_wk_p\eta$ are canceled by zeros of its numerator, so that Π is an entire function belonging to H^∞. The optimal performance level is $\gamma = \sqrt{1/\eta^2 + 1}$ and the quantity $1/\gamma \in (0, \sqrt{0.5}]$ is actually the robustness radius of the resulting system in the gap metric. It can be shown that the static gain of the overall controller

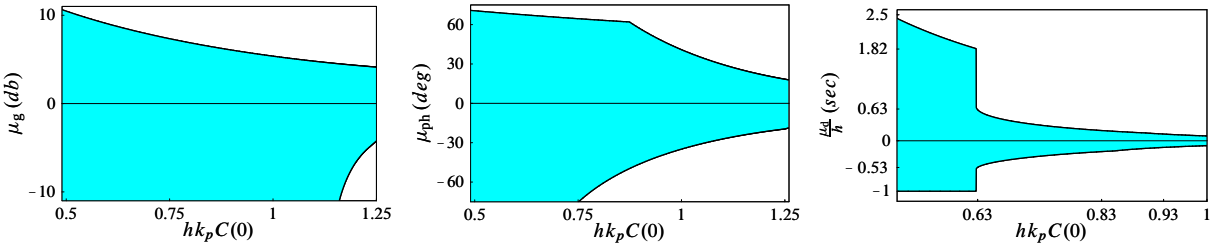
$$C(0) = k_w\eta,$$

which is an increasing function of k_w. The upper bound on the quantity $hk_pC(0)$ is now $\pi/2$, which is more than 50% larger than in the case of the two-stage design, where $hk_pC(0)$ is bounded by 1. Also, the loop transfer function is now

$$L(s) = \frac{k_p}{s}\,\frac{k_w(s^2 + (k_wk_p\eta)^2)\mathrm{e}^{-sh}}{(s + k_pk_w)^2\eta - k_pk_w(1 + \eta^2)s\mathrm{e}^{-sh}}$$

(note that its zeros at $\pm \mathrm{j}k_wk_p\eta$ are canceled by poles).

The resulting stability margins as functions of the normalized controller static gain $hk_pC(0)$ are shown in Fig. 2.7. As in the example in §2.2.1, μ_{ph} and μ_{g} are decreasing continuous (though not differentiable) functions of $C(0)$ whereas μ_{d} has a discontinuity points at $hk_pC(0) \approx 0.63$, see Fig. 2.7(c). This phenomenon can again be explained by observing that at this $C(0)$ an additional crossover frequency emerges, see Fig. 2.8. The only qualitative difference from the case studied in §2.2.1 is that now there is a weight k_w for which the controller $C(s)$ becomes unstable (although the closed-loop system is still stable). This can be seen in Fig. 2.8, where the Nichols plots of $L(\mathrm{j}\omega)$ at $hk_pC(0) = 0.83$ and $hk_pC(0) = 0.93$ encircle the critical point.



(a) μ_{g} versus $hk_pC(0)$ (b) μ_{ph} versus $hk_pC(0)$ (c) μ_{d} versus $hk_pC(0)$

Figure 2.7: Stability margins vs. normalized controller gain $hk_pC(0)$ (H^∞ loop shaping)

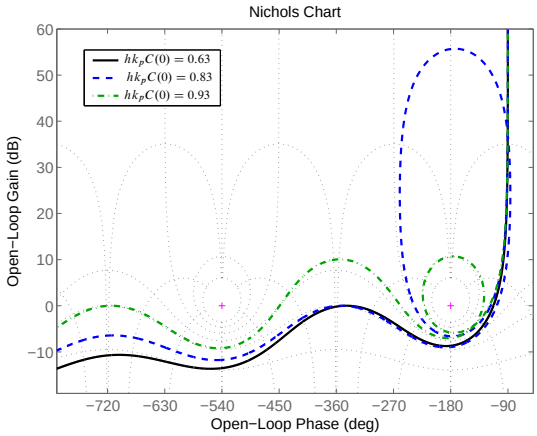


Figure 2.8: Nichols charts of $L(s)$ for different crossover proliferation (H^∞ loop-shaping design)

2.3 Generalizations

The situation described in the examples above turns out to be generic. A similar behavior (i.e., the proliferation of crossover frequencies as the requirements to the loop gain become more aggressive) takes place in all other simulations we carried out. In this section we present some explanations of the phenomenon.

2.3.1 High-gain design of Smith predictor

In the case of the classical Smith controller with high-gain design of the primary controller, the crossover proliferation phenomenon has a clear explanation. To see this, rewrite the actual loop transfer function as

$$L = \frac{P\tilde{C}}{1 + P\tilde{C}(1 - \mathrm{e}^{-sh})} = \frac{1}{1/\tilde{L} + 1 - \mathrm{e}^{-sh}},$$

where $\tilde{L} := P\tilde{C}$ is the loop gain for the design of first stage. If the primary controller $\tilde{C}$ is designed to achieve a high loop gain $\tilde{L}$, the actual loop gain

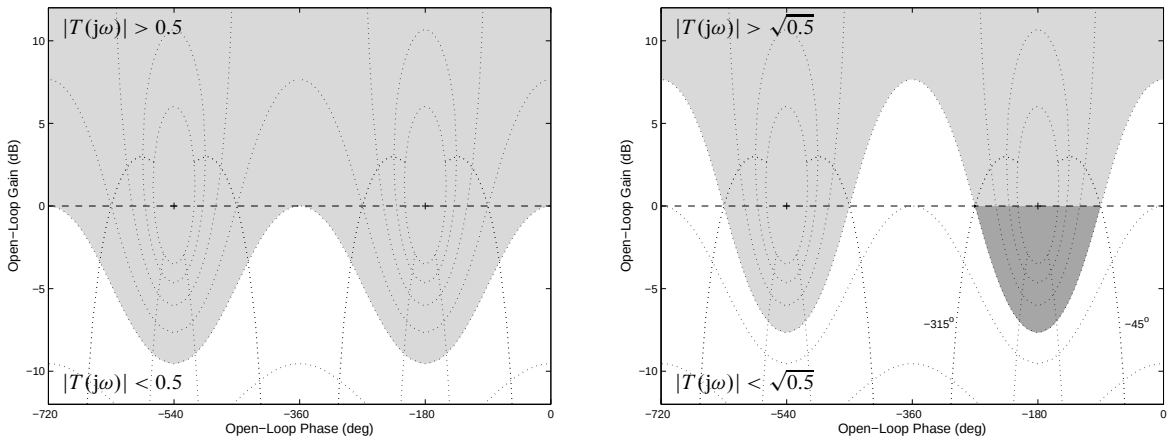
$$L \to L_\alpha = \frac{\mathrm{e}^{-sh}}{1 - \mathrm{e}^{-sh}},$$

where L_α is defined by (2.5). As discussed in the end of §2.2.1, this transfer function has infinitely many crossover frequencies and the zero delay margin. Of course, the infinite loop gain is never achievable, so L_α is never achievable either. Yet for high-gain $\tilde{L}$ we may expect that L behaves similarly, i.e., also has multiple crossover frequencies.

2.3.2 M and N circles

More insight into the crossover proliferation phenomenon can be gained by considering the relations between the frequency responses of the loop gain L and the closed-loop complementary sensitivity transfer function $T = L/(1+L)$, which are known as M and N circles (Franklin *et al.*, 2002) (on either the Nyquist or Nichols plots). To simplify the exposition, we assume here that the open loop is stable. The unstable case can be addressed by similar arguments.

The classical Smith controller design is typically based on shaping the magnitude of the closed-loop transfer function $T = L/(1 + L)$. This is simplified by the fact that in this case $T = T_\mathrm{r}\mathrm{e}^{-sh}$ for some



(a) Areas divided by the M-circle $|T(j\omega)| = 0.5$

(b) Areas divided by the M-circle $|T(j\omega)| = \sqrt{0.5}$

Figure 2.9: M and N circles as Nichols charts grid

rational T_r, so that the magnitude shaping problem is essentially finite dimensional. We thus start with conditions on $|T(j\omega)|$ guaranteeing a unique crossover frequency. To this end, consider the Nichols grid in Fig. 2.9(a), which corresponds to M and N circles. It is readily seen that the M-circle corresponding to $|T(j\omega)| = 0.5$ is almost completely (except for the tangential points where the open-loop phase is $-2\pi k$) located bellow the 0 dB level for $|L(j\omega)|$ (the dashed line). Hence, if the design keeps $|T(j\omega)| < 0.5$ for all ω larger than the first crossover, ω_{c1}, no additional crossover frequencies occur. The condition $|T(j\omega)| < 0.5$, however, effectively means that the feedback is inefficient at these frequencies. This might be rather restrictive, because the presence of the loop delay imposes limitations on the first crossover frequency. In fact, the use of DTCs is motivated by the possibility to increase the closed-loop bandwidth beyond the first crossover.

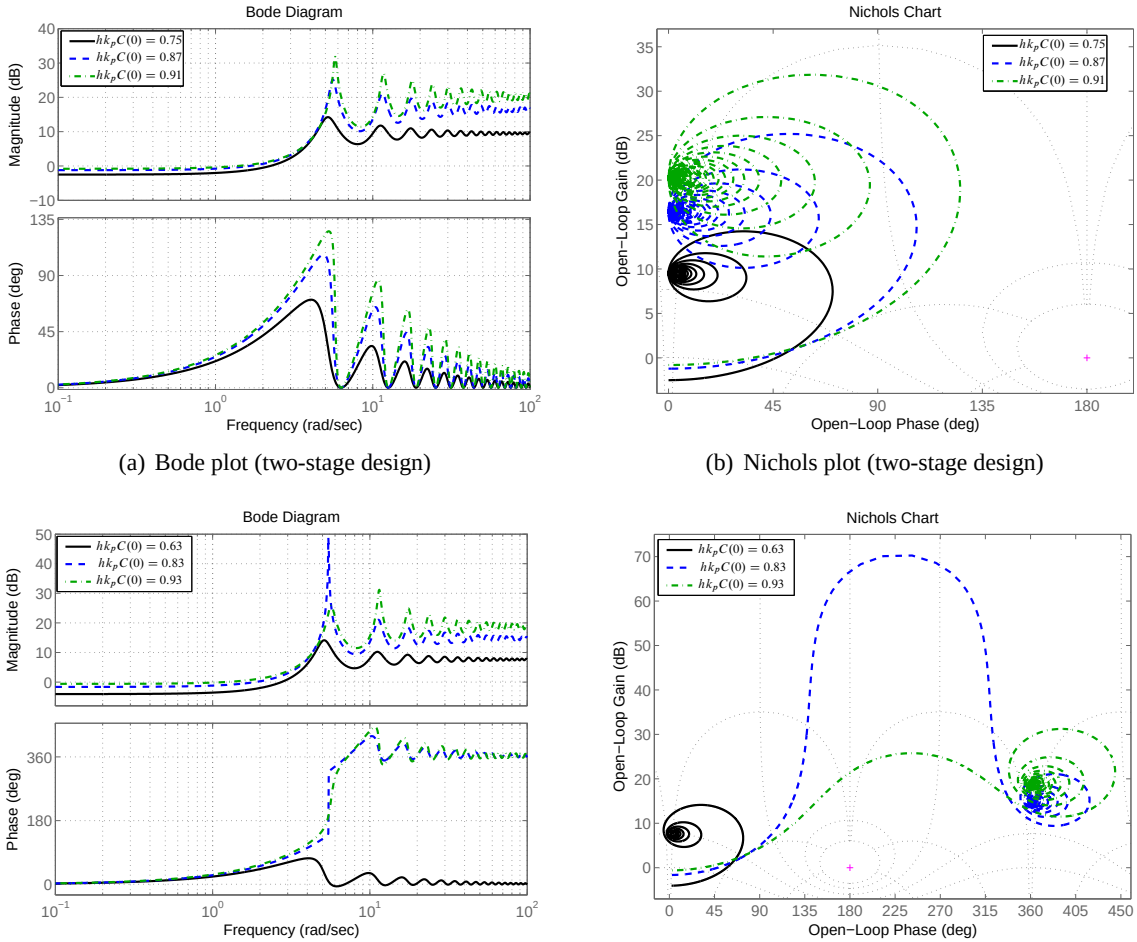
More precise conditions for the absence of additional crossovers should make use of the phase information about $T(j\omega)$. Consider, for example, what happens in the closed-loop bandwidth frequency range. Define by ω_b (the closed-loop bandwidth) the maximal frequency for which $|T(j\omega)| \geq \sqrt{0.5}$ for all $\omega \leq \omega_b$. This is equivalent to the condition that the loop frequency response $L(j\omega)$ is located in the gray regions in Fig. 2.9(b) for all $\omega \leq \omega_b$. To avoid crossover proliferation, we must keep $L(j\omega)$ in the dark gray area in Fig. 2.9(b) for all $\omega_{c1} \leq \omega \leq \omega_b$. Inspecting the corresponding N-circles, we then readily conclude that this is possible *only if* $\measuredangle T(j\omega) \in (-\frac{3}{4}\pi, -\frac{1}{4}\pi)$. Even though this condition is only necessary, it does shed light on the connection between the crossover proliferation phenomenon and aggressive shaping of $|T(j\omega)|$. Indeed, if we attempt to end up with a large (comparing to the delay h) bandwidth ω_b, the phase lag of $T(j\omega_b)$ will tend to be large as the result of the delay term $e^{-j\omega_b h}$ (remember that T must be stable). Hence, the condition $\measuredangle T(j\omega) > -\frac{3}{4}\pi$ becomes rather restrictive as the required bandwidth increases.

The arguments above can be extended to show that crossover proliferation is avoided only if

$$\measuredangle T(j\omega) > -2\pi + \arccos \tfrac{0.5}{|T(j\omega)|}, \quad \forall \omega \in (\omega_{c1}, \omega_{0.5}], \tag{2.6}$$

where by $\omega_{0.5}$ we denote the smallest frequency at which $|T(j\omega)| = 0.5$. In other words, the larger $|T(j\omega)|$ is, the wider phase range of $T(j\omega)$ should be kept off in order to avoid the appearance of the second crossover. Proceeding with these arguments, we can derive necessary conditions, in terms of $\measuredangle T$, for avoiding the further crossovers.

Note that condition (2.6) for the absence of crossover proliferation is generic in the sense that it does not depend on the controller structure. Yet it is the DTC configuration that makes an aggressive shaping



(a) Bode plot (two-stage design)

(b) Nichols plot (two-stage design)

(c) Bode plot (H^∞ loop shaping)

(d) Nichols plot (H^∞ loop shaping)

Figure 2.10: $C(s)$ for the designs in Section 2.2

of $|T(\mathrm{j}\omega)|$ possible for dead-time systems. The arguments above may imply that crossover proliferation is the price we pay for this.

2.3.3 Bode's gain-phase relation

Another possible explanation of the oscillatory behavior of the loop gain in DTC-based schemes can be deduced from the phase lead properties of resulting controllers. Åström (1977) argued that the Smith controller can be thought of as a lead network. The Bode plots in Fig. 2.10 of the controllers obtained in Section 2.2 confirm this observation. When a higher crossover is required, more phase lead should be introduced to the loop to compensate the phase lag of the delay element e^{-sh}. The phase lead, however, comes at a high price: the high-frequency gain typically increases and a large positive slope of the magnitude plot in the crossover region arises. To illustrate the latter point, consider the Bode's gain-phase integral relation (Skogestad and Postlethwaite, 1996) at $\omega = \omega_\mathrm{c}$ (we assume here that $\measuredangle C(0) = 0$):

$$\measuredangle C(\mathrm{j}\omega_\mathrm{c}) = \frac{1}{\pi}\int_{-\infty}^{\infty} \frac{\mathrm{d}\ln|C|}{\mathrm{d}\nu} \ln\coth\frac{|\nu|}{2}\mathrm{d}\nu - \sum_{i=1}^{n_z} \measuredangle\frac{z_i - \mathrm{j}\omega_\mathrm{c}}{z_i^* + \mathrm{j}\omega_\mathrm{c}} + \sum_{i=1}^{n_p} \measuredangle\frac{p_i - \mathrm{j}\omega_\mathrm{c}}{p_i^* + \mathrm{j}\omega_\mathrm{c}}, \tag{2.7}$$

where n_z and n_p are the numbers of the right-half plane poles and zeros of $C(s)$, respectively, z_i and p_i are these zeros and poles, and $\nu = \frac{\omega}{\omega_c}$. Note that the second term in the right-hand side of (2.7) is always negative while the third term there is always positive. Since $\ln\coth\frac{|\nu|}{2}$ decreases rapidly as ω deviates from ω_c, the integral in (2.7) depends mostly on the behavior of $\frac{d\ln|C|}{d\nu}$ (which is the slope of $|C|$ on the Bode plot) near ω_c. When $|C|$ is monotonic, any attempt to achieve a large positive $\measuredangle C(j\omega_c)$ (phase lead) leads to a high positive slope of the magnitude in an intendant crossover region. This implies either that the crossover frequency decreases or that the static gain of C should be decreased in order to keep the required ω_c.

It appears that DTC attempts to circumvent this problem by introducing "fast" oscillations of the magnitude and phase of the controller. In this case a phase lead does not necessarily mean that the slope of $|C|$ is positive over a large frequency range. Yet this also implies that phase leads is "localized" to a very narrow band. When even this strategy cannot provide a sufficient phase lead, DTC introduces unstable poles, so that the phase is led by the third term in the right-hand side of (2.7), see the diagram in Fig. 2.10(c).

Remark 2.3 *As already mentioned in the Introduction, the fact that the use of the Smith predictor and related controllers might give rise to multiple magnitude peaks of the loop frequency response was noticed in many early studies; see, e.g., (Åström, 1977; Georgiou and Smith, 1992). Yet in most of these studies the appearance of such peaks was not linked to the deterioration of the delay margin. To the best of my knowledge, the only paper discussing the connection between the crossover proliferation and robustness in general is (Horowitz, 1983). It is argued there that the proliferation of the crossover regions should result in a non-robust design. Horowitz (1983), however, did not address the robustness to delay variations and, actually, questioned a general applicability of the DTC-based design with which we disagree. Adam* et al. *(2000) noticed the link between the delay margin of the Smith controller and crossover proliferation, yet did not address the underlying reasons of the latter and whether it is generic in the context of DTC.*

2.4 Some design guidelines

The arguments above suggest that the design of DTC indeed tends to become extremely sensitive to delay uncertainty when aggressive control strategies are used. This is true both for the two-stage design and for direct optimization-based approaches. Moreover, in the former case it might be hard to reveal the degradation of the dead-time robustness in the first stage (the design of the primary controller for $\tilde{P}$) only.

As a possible remedy, the following steps may be considered:

- When a large delay margin is required, it is important to prevent the proliferation of the crossover frequencies. For example, in the problem considered in §2.2.1 such a consideration would impose the following constraint of the admissible gain of $\tilde{C}$: $k_c < 0.749/(hk_p)$. Clearly, this condition imposes limitations on the achievable closed-loop bandwidth. In other words, it rules out "aggressive" controllers. The experience shows that even under this limitation the use of DTCs enables one to end up with controllers having higher gain than in the case when finite-dimensional controllers are designed for dead-time systems, provided such finite-dimensional controllers have complexity, compatible with that of the primary controller.

- When the primary controller is designed in the first stage, not only the magnitude but also the phase of the closed-loop transfer function T should be accounted for. This effectively implies that the design of the primary controller should take into account the phase lag caused by the loop delay. In a sense, this furthers the arguments of Palmor (1980), who emphasized that the primary controller should not be designed "as if there were no delays" at all (because this might lead to disastrous results).

- One may argue that when k_c in Subsection 2.2.1 approaches $0.748/(hk_p)$, the maximal local peak of $|C(\mathrm{j}\omega)|$ becomes too close to 0 dB so that even a small gain increase will lead to the appearance of two additional crossover frequencies and, consequently, a dramatic deterioration of μ_d. This situation can be prevented by requiring that the local peaks be "sufficiently" far from the 0 dB level. To this end, a new quantity, which we call the *crossover proliferation margin* μ_{cp}, may be introduced:

  $$\mu_\mathrm{cp} := -\sup_{\omega>\omega_\mathrm{c},\mathrm{d}|L(\mathrm{j}\omega)|/\mathrm{d}\omega=0} |L(\mathrm{j}\omega)| \text{ (in dB)},$$

  where ω_c is the maximal "legal" crossover frequency. This quantity is not a stability margin like the gain or phase margins, it does not measure the distance to the critical point. Rather, it is a measure of safety from the delay margin point of view.

Note that the limitation on the number of the crossover frequencies can quite easily be incorporated in classical design procedures for $\tilde{C}$ (PID tuning, loop shaping, etc). It is, however, not completely clear how/whether this consideration can be incorporated into optimization-based approaches. The discussion in the second paragraph of Section 2.3 suggests that not only the gain, but also the phase of the closed-loop system might need to be shaped toward this end.

It is worth stressing that there might be situations when the presence of multiple crossover frequencies is beneficial (i.e., in the control of flexible structures (Nordin and Gutman, 1995; Kidron and Yaniv, 1995)). In such situations, it might make more sense to look at the second, the third, etc peaks. We believe, however, that the facts that the proliferation of crossover frequencies inevitably leads to a deterioration of the delay margin and that the use of DTCs tends to introduce this proliferation (especially when high-gain primary controllers are used) should be fully appreciated in the design of dead-time compensators.

2.5 Technical derivations

2.5.1 Modified Smith predictor

In this section some details about constructing the graphs of §2.2.1 are provided. The starting point here is the loop transfer function in (2.4), i.e.,

$$L(s) = \frac{k_p k_c \mathrm{e}^{-sh}}{s + k_p k_c (1 - sh - \mathrm{e}^{-sh})}$$

(remember that the closed-loop system is stable iff $0 < k_c < \frac{1}{hk_p}$). Denoting $\tilde{s} := sh$ and

$$\varkappa := \frac{1}{k_p k_c h} - 1 \geq 0,$$

the loop transfer function can be rewritten as follows:

$$L(\tilde{s}) = \frac{\mathrm{e}^{-\tilde{s}}}{1 + \varkappa\tilde{s} - \mathrm{e}^{-\tilde{s}}}. \tag{2.8}$$

The frequency response (in terms of the normalized frequency $\tilde{\omega} := \omega h$) of the loop transfer function is then

$$\begin{aligned} L(\mathrm{j}\tilde{\omega}) &= \frac{\mathrm{e}^{-\mathrm{j}\tilde{\omega}}}{1 + \mathrm{j}\varkappa\tilde{\omega} - \mathrm{e}^{-\mathrm{j}\tilde{\omega}}} = \frac{\cos\tilde{\omega} - \mathrm{j}\sin\tilde{\omega}}{1 - \cos\tilde{\omega} + \mathrm{j}(\varkappa\tilde{\omega} + \sin\tilde{\omega})} \\ &= \frac{1}{(1 + \mathrm{j}\varkappa\tilde{\omega})\mathrm{e}^{\mathrm{j}\tilde{\omega}} - 1} = \frac{1}{\cos\tilde{\omega} - \varkappa\tilde{\omega}\sin\tilde{\omega} - 1 + \mathrm{j}(\sin\tilde{\omega} + \varkappa\tilde{\omega}\cos\tilde{\omega})}, \end{aligned}$$

which is nonzero and finite $\forall\tilde{\omega} \in (0, \infty)$.

Crossover frequencies

Crossover frequencies are the frequencies ω_c at which $|L(j\omega_c)| = 1$. Applying to the loop transfer function above, we have:

$$\begin{aligned} |L(j\tilde{\omega})|^2 &= \frac{1}{|1-\cos\tilde{\omega}+j(\varkappa\tilde{\omega}+\sin\tilde{\omega})|^2} \\ &= \frac{1}{(1-\cos\tilde{\omega})^2+(\varkappa\tilde{\omega}+\sin\tilde{\omega})^2} \\ &= \frac{1}{2-2\cos\tilde{\omega}+\varkappa^2\tilde{\omega}^2+2\varkappa\tilde{\omega}\sin\tilde{\omega}} \end{aligned}$$

Thus, the (normalized) crossover frequencies satisfy

$$\tilde{\omega}_c^2\varkappa^2 + 2\tilde{\omega}_c\sin\tilde{\omega}_c\varkappa + (1-2\cos\tilde{\omega}_c) = 0$$

or, equivalently, $\varkappa$'s should satisfy

$$\varkappa = \frac{-\sin\tilde{\omega}_c \pm \sqrt{\sin^2\tilde{\omega}_c + 2\cos\tilde{\omega}_c - 1}}{\tilde{\omega}_c} = \frac{-\sin\tilde{\omega}_c \pm \sqrt{\cos\tilde{\omega}_c(2-\cos\tilde{\omega}_c)}}{\tilde{\omega}_c}. \tag{2.9}$$

Taking into account that $\varkappa > 0$, any crossover frequency must satisfy:

$$\begin{cases} \cos\tilde{\omega}_c \geq 0 & \text{if } \sin\tilde{\omega}_c \leq 0 \\ \cos\tilde{\omega}_c \geq \frac{1}{2} & \text{if } \sin\tilde{\omega}_c > 0 \end{cases}$$

or, equivalently,

$$\tilde{\omega}_c \in \left(0, \tfrac{1}{3}\pi\right] \cup \left[(-\tfrac{1}{2}+2l)\pi, (\tfrac{1}{3}+2l)\pi\right], \quad l = 1, 2, \ldots \tag{2.10}$$

which can also be stated as $\tilde{\omega}_c \notin (\frac{1}{3}\pi + 2\pi l, \frac{3}{2}\pi + 2\pi l)$, and for any $\tilde{\omega}_c$ satisfying (2.10) there exists at least one $\varkappa$ for which this $\tilde{\omega}_c$ is a crossover frequency. Note that the "+" version of (2.9) produces an admissible (i.e., non-negative) $\varkappa$ for all frequencies in (2.10), whereas the "−" version of (2.9) produce an admissible $\varkappa$ only for

$$\tilde{\omega}_c \in \left[(-\tfrac{1}{2}+2l)\pi, (-\tfrac{1}{3}+2l)\pi\right], \quad l = 1, 2, \ldots.$$

The dependence of $\varkappa$ on $\tilde{\omega}_c$ is shown in Fig. 2.11. It is seen that for sufficiently large $\varkappa$ (equivalently, small k_c), namely for $\varkappa > 0.335078$, there is only one crossover frequency, in the interval $(0, \pi/3]$, for each value of $\varkappa$. Then, for $0.150774 < \varkappa \leq 0.335078$, two additional crossover frequencies appear in the interval $[3\pi/2, 7\pi/3]$. Reducing $\varkappa$ further, $0.097406 < \varkappa \leq 0.150774$, gives rise to the appearance of an additional pair of crossover frequencies in $[7\pi/2, 13\pi/3]$. This process is repeated for more and more intervals of $\varkappa$ so that as $\varkappa \to 0$ the number of crossover frequencies approaches infinity.

Note that at each interval of frequencies $[(3/2+2l)\pi, (7/3+2l)\pi]$ the curves in Fig. 2.11 fall from their peaks in two directions downward: to the left and to the right. Each of these directions corresponds to a crossover frequency. For the reason that will become apparent later on we call the right side the *positive edge* and the left side the *negative edge*. The former edge contains only the "+" part of (2.9) starting from the peak frequency (e.g., $\tilde{\omega}_c = 5.10751$ for the first interval) and going toward $(7/3+2l)\pi$. The negative edge contains both the "+" part of (2.9) (from $(3/2+2l)\pi$ to the peak) and all its "−" part.

Phase margin

As follows from (2.9), at crossover frequencies

$$\varkappa\tilde{\omega}_c + \sin\tilde{\omega}_c = \pm\sqrt{\cos\tilde{\omega}_c(2-\cos\tilde{\omega}_c)} = \pm\sqrt{1-(1-\cos\tilde{\omega}_c)^2}.$$

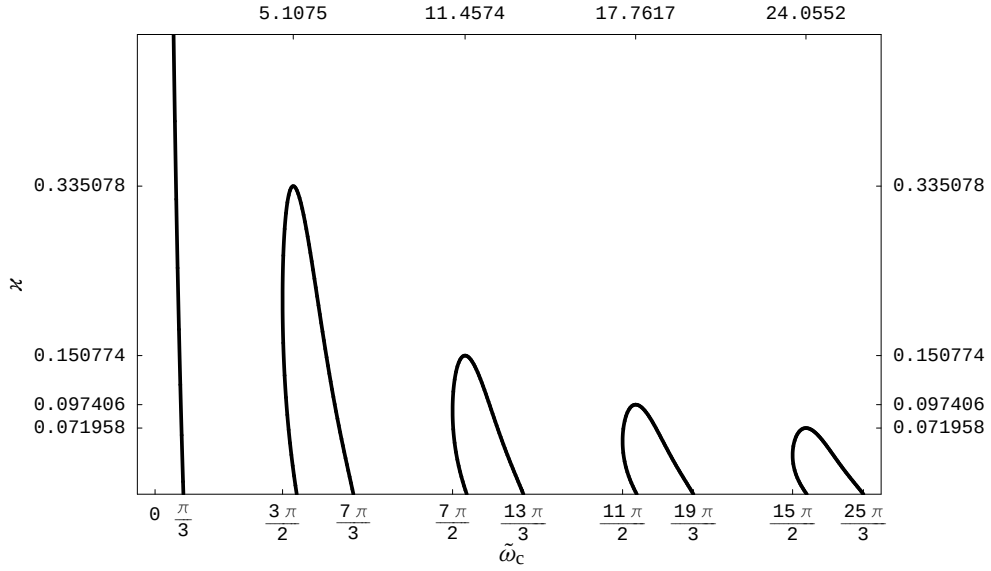


Figure 2.11: $\varkappa$ vs. $\tilde{\omega}_c$

Thus,

$$\begin{aligned}\arg\big(L(j\tilde{\omega}_c)\big) &= -\tilde{\omega}_c - \arg\big(1-\cos\tilde{\omega}_c + j(\varkappa\tilde{\omega}_c + \sin\tilde{\omega}_c)\big)\\ &= -\tilde{\omega}_c - \arg\big(1-\cos\tilde{\omega}_c \pm j\sqrt{1-(1-\cos\tilde{\omega}_c)^2}\big)\\ &= -\tilde{\omega}_c \mp \arctan\left(\sqrt{\frac{1}{(1-\cos\tilde{\omega}_c)^2}-1}\right). \end{aligned} \tag{2.11}$$

The analysis of the sensitivity of the closed-loop system to phase variations (the phase margin) depends on the edge on which the corresponding crossover frequency is located. For the frequencies located on the positive edge, a phase lag is destabilizing, whereas for those on the negative edge a phase lead eventually leads to the crossing of the critical point. For this reason, the analysis of μ_{ph} must be split according to the edge in Fig. 2.11.

- ***l*th positive edge:** In this case we are confined with the "$-$" part of (2.11) and regard the phase margin as μ^+_{ph} (in the lag direction). The closest critical point is now the one with the phase $(2l+1)\pi$, so that for every $l = 0, 1, \ldots$

$$\mu^+_{\text{ph},l} = (2l+1)\pi - \tilde{\omega}_{c,l} - \arctan\left(\sqrt{\frac{1}{(1-\cos\tilde{\omega}_{c,l})^2}-1}\right) \tag{2.12}$$

  and $\tilde{\omega}_{c,l} \in [\tilde{\omega}_{\text{peak},l}, (1/3+2l)\pi]$, where $\tilde{\omega}_{\text{peak},l}$ is the frequency at which the lth peak occur. Note that $\tilde{\omega}_{\text{peak},0} = 0$ and for $l > 0$

$$\tilde{\omega}_{\text{peak}} = \big[\ 5.10751\ \ 11.4574\ \ 17.7617\ \ 24.0552\ \ \ldots\ \big],$$

  cf. the upper abscissa in Fig. 2.11. Note that $\mu^+_{\text{ph},l} \in \big[\frac{\pi}{3}, (2l+1)\pi + \arg(L(j\tilde{\omega}_{\text{peak},l}))\big]$ for all l.

- ***l*th negative edge:** The situation in this case is more complicated as both signs in (2.11) should be accounted for. In both cases the phase margin is regarded as μ^-_{ph} (in the lead direction) to the closest

critical point, which is now the one with the phase $(2l-1)\pi$. So for every $l = 1, 2, \ldots$ we have:

$$\mu^-_{\text{ph},l} = \begin{cases} (2l-1)\pi - \tilde{\omega}_{c,l} - \arctan\left(\sqrt{\dfrac{1}{(1-\cos\tilde{\omega}_{c,l})^2} - 1}\right) & \text{if } \varkappa \in \left[\frac{2}{(2l+1)\pi}, \varkappa_{\text{peak},l}\right] \\ (2l-1)\pi - \tilde{\omega}_{c,l} + \arctan\left(\sqrt{\dfrac{1}{(1-\cos\tilde{\omega}_{c,l})^2} - 1}\right) & \text{if } \varkappa \in \left[0, \frac{2}{(2l+1)\pi}\right] \end{cases} \tag{2.13}$$

where $\varkappa_{\text{peak},l}$ is the $\varkappa$ corresponding to the lth peak. Some straightforward (at least with the help of Mathematica) calculations yield that

$$\varkappa_{\text{peak}} = \left[\, 0.335078 \;\; 0.150774 \;\; 0.097406 \;\; 0.0719582 \;\; \ldots \,\right],$$

cf. the left ordinate in Fig. 2.11.

The phase margin (for each $\varkappa$) in the "lag direction", μ^+_{ph}, should now be calculated as the minimum over all $\mu^+_{\text{ph},l}$, $l = 0, 1, \ldots$. Similarly, the phase margin in the "lead direction", μ^-_{ph}, is the maximum over all $\mu^-_{\text{ph},l}$, $l = 1, 2, \ldots$ (which are negative). The calculations are simplified due to the fact that the angular distance to the critical point is decreasing as a function of l or, equivalently, of the crossover frequency. This claim is proved below.

Lemma 2.1 *For every $\varkappa > 0$ and all corresponding crossover indices $l \geq 0$,*

$$-\mu^-_{ph,l} < \mu^+_{ph,l} < -\mu^-_{ph,l+1} < \mu^+_{ph,l+1}$$

(with some abuse of notation we assume here that $\mu^-_{ph,0} = \frac{\pi}{3}$).

Proof: The proof exploits the connection between open- and closed-loop properties known as the M-circles (Franklin *et al.*, 2002). Note that the closed-loop transfer function for the loop gain as in (2.8) is

$$T(\tilde{s}) = \frac{L(\tilde{s})}{1 + L(\tilde{s})} = \frac{\mathrm{e}^{-\tilde{s}}}{\varkappa\tilde{s} + 1}.$$

The magnitude of the frequency response of this transfer function, $|T(\mathrm{j}\tilde{\omega})|$, is a decreasing function of the normalized frequency $\tilde{\omega}$. This means that as $\tilde{\omega}$ increases, the level of the M-circles crossed by the Nyquist (or Nichols) plot of $L(\mathrm{j}\tilde{\omega})$ decreases. The statement of the Lemma follows now from the facts that *(a)* the angular distance from M-circles to the critical point decreases as M-levels increase; *(b)* the crossover frequencies increase with l; and *(c)* the crossover frequency corresponding to the lth positive edge is strictly larger than that corresponding to the lth negative edge in Fig. 2.11. ■

Lemma 2.1 leads to the following proposition, which is the main result about the phase margin (which is $\mu_{\text{ph}} = \min\{\mu^+_{\text{ph}}, -\mu^-_{\text{ph}}\}$) of the system in §2.2.1 (this μ_{ph} is plotted in Fig. 2.3(b) vs. $k_p k_c h = \frac{1}{\varkappa+1}$.).

Proposition 2.1 *Given a $\varkappa > 0$, the phase margin, μ_{ph}, of the modified Smith predictor with a proportional controller is given by*

$$\mu_{ph} = \pi - \tilde{\omega}_{c1} - \arctan\left(\sqrt{\frac{1}{(1-\cos\tilde{\omega}_{c1})^2} - 1}\right) \quad \textit{(in rad)},$$

where $\tilde{\omega}_{c1}$ is the unique solution of (2.9)*, corresponding to the "+" sign, in the interval $\tilde{\omega} \in (0, \pi/3]$ (it is actually the first crossover frequency).*

Delay margin

By (2.2), the delay margins are calculated my finding extrema over all possible $\mu^{+}_{\mathrm{ph},l}/\tilde{\omega}_{c,l}$ and $\mu^{-}_{\mathrm{ph},l}/\tilde{\omega}_{c,l}$. There is no much to be done here analytically, as the resulting formulae are quite complicated. The plot in Fig. 2.3(c) is obtained by the numerical comparison over the first 100 crossover intervals.

Gain margin

The gain margin is calculated as $1/|L(\mathrm{j}\tilde{\omega})|$ for those $\tilde{\omega}$ for which $\Im(L(\mathrm{j}\tilde{\omega})) = 0$ and $\Re(L(\mathrm{j}\tilde{\omega})) \leq 0$. The first condition above is equivalent to $\sin\tilde{\omega} + \varkappa\tilde{\omega}\cos\tilde{\omega} = 0$ or

$$\varkappa = -\frac{\sin\tilde{\omega}}{\tilde{\omega}\cos\tilde{\omega}} = -\frac{\tan\tilde{\omega}}{\tilde{\omega}}. \tag{2.14}$$

Since $\varkappa > 0$, the equation above is solvable iff $\tan\tilde{\omega} > 0$. Substituting the $\varkappa$ above to the formula for $L(\mathrm{j}\tilde{\omega})$, we have that for all $\tilde{\omega}$ for which $\Im(L(\mathrm{j}\tilde{\omega})) = 0$:

$$L(\mathrm{j}\tilde{\omega}) = \frac{\cos\tilde{\omega}}{1-\cos\tilde{\omega}},$$

which is negative iff $\cos\tilde{\omega} < \frac{1}{2}$. Combining this condition with the condition $\tan\tilde{\omega} \leq 0$, we end up with the condition

$$\tilde{\omega} \in \left((-\tfrac{3}{2}+2l)\pi, (-1+2l)\pi\right] \cup \left((-\tfrac{1}{2}+2l)\pi, (-\tfrac{1}{3}+2l)\pi\right], \quad l = 1, 2, \ldots$$

It is readily verified (e.g., by plotting the corresponding graphs) that for a given $\varkappa$ there is infinite number of solutions of the equation (2.14) and these solutions asymptotically approach the sequence πl (the larger $\varkappa$ is, the faster the convergence is). Moreover, the distance from solutions of (2.14) to the corresponding πl becomes smaller as l grows. This, in turn, yields that the maximal $|L(\mathrm{j}\tilde{\omega})|$ among all points where $L(\mathrm{j}\tilde{\omega})$ intersects the negative real semi-axis is is at the first intersection, i.e., in the interval $\tilde{\omega} \in (\pi/2, \pi]$ (a rigorous proof follows the M-circle arguments of the proof of Lemma 2.1). Thus, the following proposition can be formulated:

Proposition 2.2 *Given a $\varkappa > 0$, the gain margin, μ_g, of the modified Smith predictor with a proportional controller is given by*

$$\mu_g = 1 - \frac{1}{\cos\tilde{\omega}_\phi} \geq 2,$$

where $\tilde{\omega}_\phi$ is the unique solution of (2.14) *in the interval $\tilde{\omega} \in (\pi/2, \pi]$ (it is actually the first phase crossover frequency).*

2.5.2 H^∞ loop shaping

In this section the derivation of the H^∞ loop-shaping formulae for the plant in Section 2.2 is presented. We start with the general solution (§2.5.2) and then address the special case in §2.5.2.

General formulae

The central technical step in the H^∞ loop-shaping procedure (McFarlane and Glover, 1990) is the solution of the gap optimization problem for the cascade connection of the plant $P(s)$ and the weighting function $W(s)$. We assume hereafter that this connection, $P_\mathrm{a} = PW$, is strictly proper and is given in terms of its stabilizable and detectable state-space realization as follows:

$$P_\mathrm{a}(s) = \left[\begin{array}{c|c} A & B \\ \hline C & 0 \end{array}\right].$$

Solutions for such a system are available in the literature; see (Dym *et al.*, 1995) for SISO systems and (Tadmor, 1997a) for general MIMO systems. Yet the available solutions are not readily cast in the DTC form. For this reason, the solution procedure of (Mirkin, 2003, Section III) (see also (Meinsma *et al.*, 2002)) is applied here to the gap optimization problem.

The starting point for this solution is the parametrization of all delay-free controllers. The later is based (McFarlane and Glover, 1990) on the stabilizing solutions $X \geq 0$ and $Y \geq 0$ of the following H^2 algebraic Riccati equations (AREs):

$$A'X + XA - XBB'X + C'C = 0 \quad \text{and} \quad AY + YA' - YC'CY + BB' = 0$$

(these solutions always exist). Then, given a performance level $\gamma > \gamma_{\text{opt}} := \sqrt{1 + \rho(XY)}$, the set of all γ-suboptimal controllers $C(s)$ is given as

$$C(s) = \big(G_{0,12}(s) + G_{0,11}(s)Q(s)\big)\big(G_{0,22}(s) + G_{0,21}(s)Q(s)\big)^{-1},$$

where Q is any stable transfer function satisfying $\|Q\|_\infty < \sqrt{\gamma^2 - 1}$ and

$$G_0(s) = \left[\begin{array}{c|cc} A - BB'X & ZB & ZYC' \\ \hline -B'X & I & 0 \\ C & 0 & I \end{array}\right],$$

where $Z := \gamma^2((\gamma^2 - 1)I - YX)^{-1}$ is well defined.

The solution of the dead-time version of this problem in (Mirkin, 2003; Meinsma *et al.*, 2002) is based on the extraction of dead-time controllers from the delay-free parametrization above. The direct application of the formulae there yields that the problem with the loop delay h is solvable iff $\Sigma_{22}(t)$ is nonsingular for all $t \in [0, h]$, where

$$\Sigma(t) = \left[\begin{array}{cc} \Sigma_{11}(t) & \Sigma_{12}(t) \\ \Sigma_{21}(t) & \Sigma_{22}(t) \end{array}\right] := \exp\left(\left[\begin{array}{cc} A - (I - Z)BB'X & \frac{1}{\gamma^2 - 1}ZYC'CYZ' \\ -XBB'X & -A' + XBB'(I - Z') \end{array}\right] t\right)$$

(hereafter, we write Σ to denote $\Sigma(h)$). If this condition holds, then the set of all controllers solving the problem for the augmented plant $P_{\text{a}}(s)\text{e}^{-sh}$ is given in the DTC form depicted in Fig. 2.6 for (mind the negative feedback)

$$\tilde{C}(s) = -\big(G_{h,12}(s) + G_{h,11}(s)Q(s)\big)\big(G_{h,22}(s) + G_{h,21}(s)Q(s)\big)^{-1},$$

where Q is any stable transfer function satisfying $\|Q\|_\infty < \sqrt{\gamma^2 - 1}$ and

$$G_h(s) = \left[\begin{array}{c|cc} A - BB'X & (Z + \Sigma_{12}\Sigma_{22}^{-1}X)B & \Sigma_{22}^{-\prime}ZYC' \\ \hline -B'X & I & 0 \\ C(\Sigma_{22}' - \frac{1}{\gamma^2 - 1}YZ'\Sigma_{21}') & 0 & I \end{array}\right],$$

and

$$\Pi(s) = \boldsymbol{\pi}_h\left\{\text{e}^{-sh}\left[\begin{array}{cc|c} A - (I - Z)BB'X & \frac{1}{\gamma^2 - 1}ZYC'CYZ' & ZB \\ -XBB'X & -A' + XBB'(I - Z') & -XB \\ \hline C & \frac{1}{\gamma^2 - 1}CYZ' & 0 \end{array}\right]\right\},$$

where $\boldsymbol{\pi}_h\{\cdot\}$ is the h-completion operator (Mirkin, 2003) defined as

$$\boldsymbol{\pi}_h\left\{\text{e}^{-sh}\left[\begin{array}{c|c} A & B \\ \hline C & 0 \end{array}\right]\right\} := \left[\begin{array}{c|c} A & \text{e}^{-Ah}B \\ \hline C & 0 \end{array}\right] - \text{e}^{-sh}\left[\begin{array}{c|c} A & B \\ \hline C & 0 \end{array}\right] = \left[\begin{array}{c|c} A & B \\ \hline C\,\text{e}^{-Ah} & 0 \end{array}\right] - \text{e}^{-sh}\left[\begin{array}{c|c} A & B \\ \hline C & 0 \end{array}\right].$$

In particular, the so-called *central primary controller*, i.e., the one corresponding to $Q = 0$, is

$$\tilde{C}(s) = G_{h,12}(s)G_{h,22}(s)^{-1} = \left[\begin{array}{c|c} A - BB'X - \Sigma_{22}^{-\prime}ZYC'C(\Sigma_{22}' - \frac{1}{\gamma^2-1}YZ'\Sigma_{21}') & \Sigma_{22}^{-\prime}ZYC' \\ \hline B'X & 0 \end{array}\right].$$

Note that the optimal level of γ is the one for which $\Sigma_{22}(t)$ is nonsingular for all $t \in [0, h)$ and becomes singular at $t = h$. Thus, as γ approaches the optimal level, the formula for $\tilde{C}(s)$ above becomes poorly defined since Σ_{22} is not invertible. This problem can be resolved by considering the descriptor representation of the central controller. Namely, the state-space equation of $\tilde{C}$,

$$\dot{x}_{\mathrm{c}} = \big(A - BB'X - \Sigma_{22}^{-\prime}ZYC'C(\Sigma_{22}' - \tfrac{1}{\gamma^2-1}YZ'\Sigma_{21}')\big)x_{\mathrm{c}} + \Sigma_{22}^{-\prime}ZYC'y,$$

can be rewritten as

$$\Sigma_{22}'\dot{x}_{\mathrm{c}} = \big(\Sigma_{22}'(A - BB'X) - ZYC'C(\Sigma_{22}' - \tfrac{1}{\gamma^2-1}YZ'\Sigma_{21}')\big)x_{\mathrm{c}} + ZYC'y,$$

which may make sense even for singular Σ_{22}. In particular, when A is scalar, $\Sigma_{22} = 0$ and the equation above reduces to the static equation

$$0 = \tfrac{1}{\gamma^2-1}ZYC'CYZ'\Sigma_{21}'x_{\mathrm{c}} + ZYC'y$$

which, in turn, leads to the following formula for the central primary controller:

$$\tilde{C}(s) = -(\gamma^2 - 1)\frac{BX}{CYZ'\Sigma_{21}'}. \tag{2.15}$$

This system is well defined as for any meaningful problem formulation $CY \neq 0$ (otherwise, either $C = 0$ or $B = 0$), $Z \neq 0$ since the minimal performance for the delay problem is strictly smaller than that in the delay-free case, and $\Sigma_{21} \neq 0$ since otherwise the matrix exponential Σ would be singular (which is clearly impossible). This formula will be used in the next subsection.

Finally, the controller for $P(s)$ (remember, the controller $C(s)$ above is designed for $P_{\mathrm{a}} = PW$) is $W(s)C(s)$.

Integrator with delay

The H^∞ loop-shaping design addressed in §2.2.2 is based on the solution of the gap robustness optimization for the augmented plant

$$P_{\mathrm{a}}(s) = \frac{k_p k_w}{s}\mathrm{e}^{-sh} = \frac{k_{\mathrm{a}}}{s}\,\mathrm{e}^{-sh}$$

where with no loss of generality we assume that $k_{\mathrm{a}} := k_p k_w > 0$. A state-space realization of this P_{a} is

$$P_{\mathrm{a}}(s) = \left[\begin{array}{c|c} 0 & \sqrt{k_{\mathrm{a}}} \\ \hline \sqrt{k_{\mathrm{a}}} & 0 \end{array}\right],$$

which leads to the following AREs:

$$-k_{\mathrm{a}}X^2 + k_{\mathrm{a}} = 0 \quad \text{and} \quad -k_{\mathrm{a}}Y^2 + k_{\mathrm{a}} = 0,$$

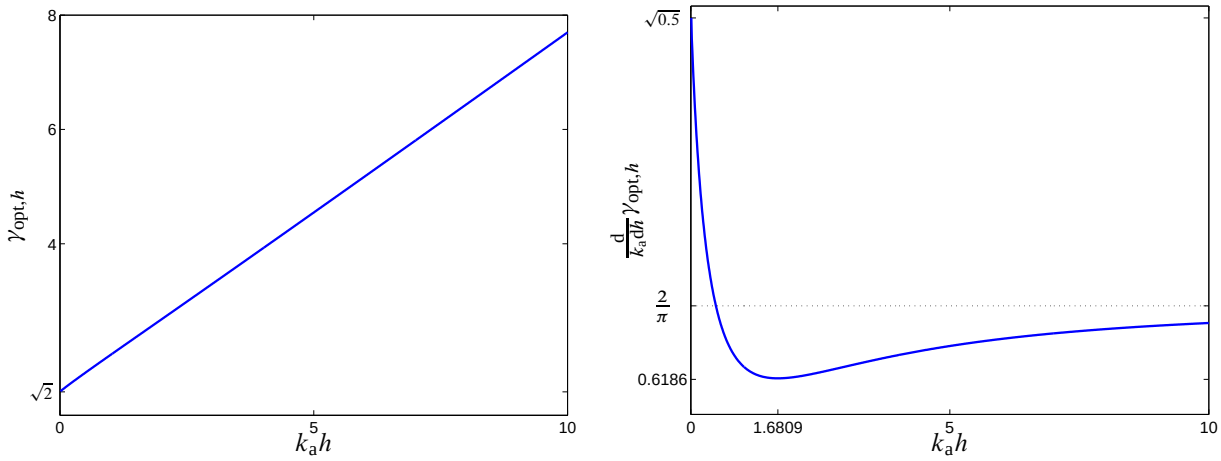
from which $X = Y = 1$, $\gamma_{\mathrm{opt}} = \sqrt{2}$ (for the delay-free problem), and $Z = \frac{\gamma^2}{\gamma^2-2}$. Then for every $\gamma > \sqrt{2}$

$$\begin{aligned}\Sigma(t) &= \exp\left(\begin{bmatrix} \frac{2}{\gamma^2-2} & \frac{\gamma^4}{(\gamma^2-2)^2(\gamma^2-1)} \\ -1 & \frac{2}{\gamma^2-2} \end{bmatrix} k_{\mathrm{a}}t\right) \\ &= \frac{\sqrt{\gamma^2-1}}{\gamma^2-2}\begin{bmatrix} 2 & \frac{\gamma^4}{(\gamma^2-2)(\gamma^2-1)} \\ -\gamma^2+2 & -2 \end{bmatrix}\sin\Big(\frac{k_{\mathrm{a}}t}{\sqrt{\gamma^2-1}}\Big) + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\cos\Big(\frac{k_{\mathrm{a}}t}{\sqrt{\gamma^2-1}}\Big).\end{aligned}$$

Therefore, the infimum of γ's for which the problem is solvable, $\gamma_{\mathrm{opt},h}$, is the maximal γ satisfying the following transcendental equation:

$$\cos\Big(\frac{k_\mathrm{a}h}{\sqrt{\gamma^2-1}}\Big) = 2\frac{\sqrt{\gamma^2-1}}{\gamma^2-2}\sin\Big(\frac{k_\mathrm{a}h}{\sqrt{\gamma^2-1}}\Big).$$

The dependence of $\gamma_{\mathrm{opt},h}$ on normalized h is depicted in Fig. 2.12(a). It looks like a straight line, yet it is not. This is seen from the plot of the derivative of $\gamma_{\mathrm{opt},h}$ with respect to $k_\mathrm{a}h$ in Fig. 2.12(b), which is not a constant. This derivative changes from $\sqrt{0.5} \approx 0.7071$ at $h = 0$ to $\frac{2}{\pi} \approx 0.6366$ as $h \to \infty$, with the minimum of ≈ 0.6186 at $h \approx 1.6809/k_\mathrm{a}$.



(a) $\gamma_{\mathrm{opt},h}$ vs. $k_\mathrm{a}h$ (b) $\frac{\mathrm{d}}{k_\mathrm{a}\mathrm{d}h}\gamma_{\mathrm{opt},h}$ vs. $k_\mathrm{a}h$

Figure 2.12: Optimal performance as a function of the loop delay h

It is convenient to introduce a new variable at this stage:

$$\eta := \frac{1}{\sqrt{\gamma^2-1}} \in (0,1)$$

(note that η corresponding to $\gamma_{\mathrm{opt},h}$ must satisfy the equation $2\tan(k_\mathrm{a}\eta h) = 1/\eta - \eta$). In this case $Z = \frac{1+\eta^2}{1-\eta^2}$ and also

$$\Sigma \xrightarrow{\gamma\to\gamma_{\mathrm{opt},h}} \begin{bmatrix} \frac{4\eta}{1+\eta^2} & \frac{\eta(1+\eta^2)}{1-\eta^2} \\ -\frac{1-\eta^2}{\eta(1+\eta^2)} & 0 \end{bmatrix}.$$

Then, the optimal primary controller defined by (2.15) is

$$\tilde{C}(s) = \frac{1}{\eta}$$

and the DTC block is

$$\begin{aligned} \Pi(s) &= \boldsymbol{\pi}_h\left\{\mathrm{e}^{-sh}\left[\begin{array}{cc|c} \frac{\eta^2}{1-\eta^2}k_\mathrm{a} & \frac{\eta^2(1+\eta^2)^2}{(1-\eta^2)^2}k_\mathrm{a} & \frac{1+\eta^2}{1-\eta^2}\sqrt{k_\mathrm{a}} \\ -k_\mathrm{a} & -\frac{\eta^2}{1-\eta^2}k_\mathrm{a} & -\sqrt{k_\mathrm{a}} \\ \hline \sqrt{k_\mathrm{a}} & \frac{\eta^2(1+\eta^2)}{1-\eta^2}\sqrt{k_\mathrm{a}} & 0 \end{array}\right]\right\} \\ &= \boldsymbol{\pi}_h\left\{\mathrm{e}^{-sh}\frac{(1+\eta^2)k_\mathrm{a}s}{s^2+\eta^2k_\mathrm{a}^2}\right\} = \left(\frac{2\eta s+\eta(1-\eta^2)k_\mathrm{a}}{s^2+\eta^2k_\mathrm{a}^2} - \mathrm{e}^{-sh}\frac{(1+\eta^2)s}{s^2+\eta^2k_\mathrm{a}^2}\right)k_\mathrm{a}. \end{aligned}$$

It is readily seen that $\Pi(0) = 1/\eta - \eta$, so that the static gain of the overall controller $C(s)$ is $C(0) = \frac{\tilde{C}(0)}{1+\tilde{C}(0)\Pi(0)} = \eta$.

Finally, the formulae of §2.2.2 are obtained by the substitution $k_{\mathrm{a}} = k_p k_w$ and the incorporation the weighting function $W(s) = k_w$ into the controller (it multiplies $\tilde{C}(s)$ and divides $\Pi(s)$).

Chapter 3

Implementation of controllers including FIR blocks

As in other engineering fields, the design of control systems can be divided into two stages: the analytical design and the implementation (using existing hardware). Previous chapter discussed the first stage for dead-time systems. In this chapter, some aspects of the second stage for controllers including DTCs are studied.

As far as rational regulators are considered, there is a wide range of literature devoted to their implementation using analog and, especially, digital equipment. The implementation of regulators including infinite-dimensional parts is considerably less scrutinized. One of the central objectives of this chapter is to discuss a number of methods used to implement FIR block with unstable zero-pole cancellation. Of course, it is not possible to elucidated all problems connected to the FIR block implementation, but a number of important problems will be discussed. Another objective of this chapter is to show that DTC control laws are a feasible option when control of delayed plants is discussed. In other words we will show that DTC control laws can be applied to delayed plants as well as be implemented using existing hardware and used to control existing laboratory plant. Also we will discuss possible pros and cons of different implementation methods.

The discussion of the implementation issues associated with DTC-based controllers in this chapter is based on the servo control system designed for a laboratory DC motor, which is a part of the experimental setup described in Appendix A. In Section 3.1, the controller design for this system is discussed and the use of DTC-based control laws is justified. In Section 3.2 some known implementation methods for DTCs containing FIR elements are reviewed. Section 3.3 studies one of these methods, lumped-delay approximation, demonstrates that its application might result in infeasible implementations, reveals the reasons of these problems, and proposes remedies. Finally, in Section 3.4.1, DTC-based controllers are successfully implemented for the control of more complicated laboratory pendulum experiments.

3.1 Servo system for a delayed DC motor

In this section we consider the controller design for a DC motor, which is a part of the laboratory experiment described in Appendix A. The transfer function of the DC motor is

$$P_{\mathrm{m}}(s) = \frac{41.085}{s(0.71s + 1)}. \tag{3.1}$$

We consider the standard one-degree-of-freedom feedback control configuration and pose the following frequency-domain design specifications for the resulting control system:

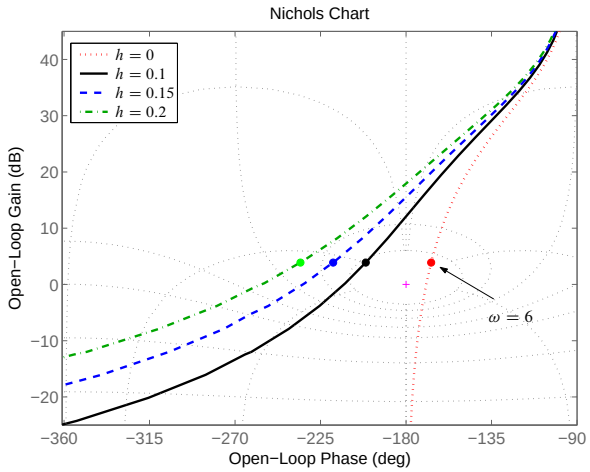


Figure 3.1: Nichols charts of $P_{\mathrm{m}}(s)\mathrm{e}^{-sh}$ with different delays

1. the regulator must include an integrator,
2. the (first) gain crossover frequency is $\omega_{\mathrm{c1}} = 6\,\mathrm{rad/sec}$,
3. the phase margin at the gain crossover frequency (ω_{c1}) is $\mu_{\mathrm{ph1}} = 0.6\,\mathrm{rad} \approx 34.4^o$.

These specifications are sufficiently transparent to enable meaningful comparison of controllers designed by different methods. At the same time, they do reflect typical requirements to the design of servo controllers. The first requirement guarantees that the steady-state tracking error for step reference signals and any constant disturbance is zero. The second requirement is taken to have the rise time of approximately $0.15 \div 0.17\,\mathrm{sec}$, which is approximately 33% higher than the rise time of the open loop plant. Note that the required crossover frequency is slightly smaller than that of the plant itself (for which $\omega_{\mathrm{c}} = 7.54$). The phase margin specification reflects the requirement to keep the loop frequency response far from the critical point, which, in turn, should result in relatively small overshoot and a reasonably good level of robustness to plant uncertainty. In fact, the second and third specifications together can be interpreted as imposing the following interpolation constraint on the loop transfer function $L(s)$:

$$L(\mathrm{j}6) = \mathrm{e}^{\mathrm{j}(0.6-\pi)} = -\mathrm{e}^{\mathrm{j}0.6}.$$

Also, they may be thought of as bringing the delay margin to the 0.1 sec level.

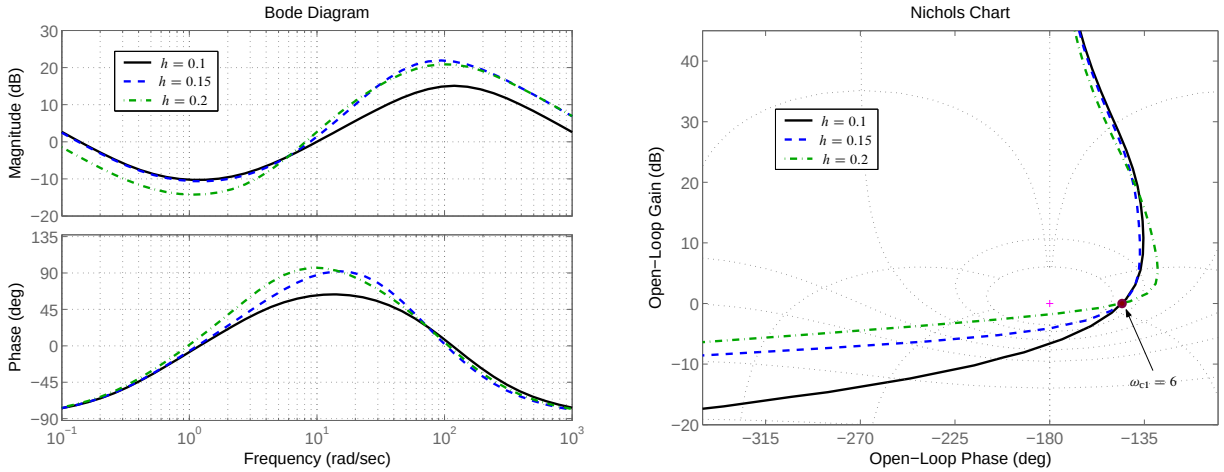
Delays to this system are introduced artificially, by delaying the shaft angle measurement in the control loop by means of software. Throughout this chapter, we consider three different loop delays from the set $\{0.1, 0.15, 0.2\}$ (in seconds). Fig. 3.1 presents the Nichols charts of these delayed plants $P_{\mathrm{m}}(s)\mathrm{e}^{-sh}$ with the frequency responses at the required crossover frequency 6 rad/sec marked by bold points. It will be demonstrated below that the delays above can be considered large for given plant and performance specifications.

3.1.1 Classical loop shaping

To highlight main problems, arising in the controller design for our system in the presence of loop delays, consider first the application of the classical loop-shaping approach, in which the required specifications are imposed by simple lead-lag networks. The motivation for the use of this approach is intelligible—transparent design and simple implementation.

The logic of the loop shaping for the systems in Fig. 3.1 is simple. First of all, a proportional gain $\frac{1}{|P_{\rm m}(j\omega_{\rm c1})|} \approx 0.64$ is added to obtain the required crossover frequency. Second, a lag controller with an integral action has to be added to meet the first specification. This lag controller can always be chosen to keep the required crossover frequency. The controller we have after these steps is clearly not sufficient because even without the lag controller the loop phase does not meet the phase margin specification. This can be clearly seen from the plots in Fig. 3.1, in the delayed cases the closed-loop system is even not stable. Hence, in the third stage a lead network has to be designed to attain the required phase margin. This step is in general highly nontrivial. The reason[1] is that any phase lead brought about by the standard lead controller comes at the expense of an increase of the controller gain at frequencies just above $\omega_{\rm c1}$. In general, the larger the required phase lead, the higher is the positive gain slope at the crossover frequency. This puts the loop to the danger of approaching, or even encircling, the critical point at the phase crossover. We are therefore limited in the maximal phase lead. Finally, at the fourth stage a low-pass filter can be added to limit the controller bandwidth.

After performing the four steps described above, the controllers and the loop frequency responses, depicted in Figs. 3.2, were obtained. The inspection of Fig. 3.2(a) shows that the larger the loop delay is,



(a) Controllers (b) Loops transfer functions

Figure 3.2: Classical loop-shaping designs for $P_{\rm m}(s)\mathrm{e}^{-sh}$

the higher phase lead at $\omega_{\rm c1}$ is added by the controller. The price is also clear there: the positive slope of the controller magnitude at $\omega_{\rm c1}$ increases with the delay. The outcome of this increased slop is seen clearly from the Nichols chart of the resulting loop gain: the loop gain at frequencies above $\omega_{\rm c1}$ increases and dangerously approaches the critical point from below as h grows.

It is worth emphasizing that the design of the lead network adopted here is based on the standard real lead network described in any introductory feedback control textbook. This is definitely not the best choice and indeed there are many advanced methods, based on a delicate play with complex lead controllers, (skew) notch and anti-notch filters, etc. More sophisticated methods, however, are not readily formalizable, they are usually ad hoc and rely heavily upon the designer experience. At the same time, the main purpose of this subsection is to highlight problems, which one faces when attempting to meet the specifications above, rather than to present the most sophisticated loop-shaping solution. The use of the basic toolkit appears to be sufficient towards this end.

[1]The reduction of the low-frequency gain can be discussed here as well.

3.1.2 DTC design via the H^∞ loop shaping

We are now in the position to apply the H^∞ loop-shaping approach described in the previous chapter to our DC motor servo control problem. Remember that in the H^∞ loop-shaping procedure the controller is split into two parts, $W(s)$ and $C_2(s)$. The first part is chosen using standard (magnitude) loop-shaping arguments without explicitly taking into account the phase (stability margin) requirements. The second part, $C_2(s)$, is then designed analytically to maximize an appropriately defined robustness radius of the augmented system $P_\mathrm{m}(s)W(s)$. Finally, the resulting controller is implemented as the cascade of W and C_2, i.e., in the form $C(s) = W(s)C_2(s)$.

The design procedure is started with the lag network $W = k\frac{\tau s+1}{s}$. It guarantees that the resulting controller contains an integrator and allows to meet the gain crossover frequency requirement by varying k (the increase of k leads to higher gain crossover frequency). We chose τ canceling the pole of P_m, so that the resulting weighted transfer function is the double integrator $WP_\mathrm{m} = \frac{k}{s^2}$. The next step is to meet the phase margin requirement. In all cases the phase margin was insufficient, so that a lead network was added to increase it. Also in the last two cases we added a low pass filter to increase the high-frequency roll off. The corner frequencies of these low pass filters were chosen to be far from the gain crossover frequency to avoid considerable reduction of the phase margin.

As we know, the H^∞ loop shaping does not give an analytical solution for the given specifications. The solution procedure is therefore iterative. Since the design rationale here is transparent, a rough solution can be obtained after a few iterations. After that, the final solution is achieved by very small changes in the weighting transfer function. The following weighing functions were finally chosen:

$$W_1(s) = \frac{1.15(s+1.4079)(s+6.61)}{s(s+10)}, \tag{3.2a}$$

$$W_2(s) = \frac{147(s+1.4079)(s+4.4)}{s(s+29)(s+30)}, \tag{3.2b}$$

$$W_3(s) = \frac{360(s+1.4079)(s+3.25)}{s(s+35)(s+40)} \tag{3.2c}$$

for $h = 0.1$, $h = 0.15$, and $h = 0.2$, respectively. These weighted transfer functions yield the following primary controllers and DTCs:

$$\tilde{C}_1(s) = \frac{2443.9674(s+10.08)(s+2.348)}{(s+676.2)(s+55.18)(s+5.981)},$$

$$\Pi_1(s) = \boldsymbol{\pi}_h\left\{\frac{-72.1344(s-10)(s+6.61)s^2\mathrm{e}^{-0.1s}}{(s+10.1)(s+3.412)(s-3.412)(s-10.1)(s^2+13.73)}\right\},$$

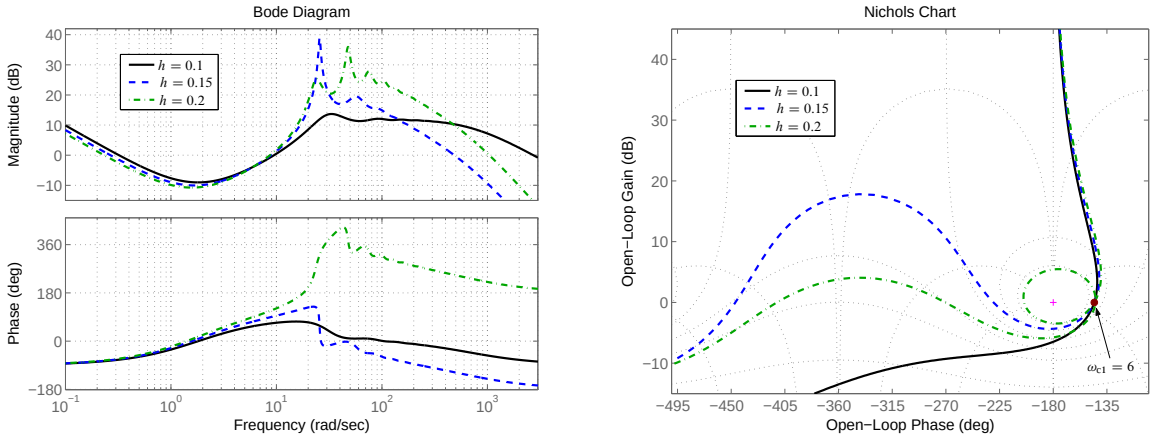
$$\tilde{C}_2(s) = \frac{2827.8652(s+2.024)(s^2+59.14s+875.6)}{(s+3139)(s-2379)(s+21.46)(s+1.942)},$$

$$\Pi_2(s) = \boldsymbol{\pi}_h\left\{\frac{-9028.4927(s-30)(s-29)(s+4.4)s^2\mathrm{e}^{-0.15s}}{(s-2.863)(s+2.863)(s^2-59.14s+875.6)(s^2+59.14s+875.6)(s^2+13.77)}\right\},$$

$$\tilde{C}_3(s) = \frac{3644.0338(s+39.63)(s+35.54)(s+1.797)}{(s+2.547e004)(s-2.469e004)(s+31.66)(s+1.796)},$$

$$\Pi_3(s) = \boldsymbol{\pi}_h\left\{\frac{21604.5959(s-40)(s-35)(s+3.25)s^2\mathrm{e}^{-0.2s}}{(s+39.63)(s+35.54)(s-35.54)(s-39.63)(s-2.471)(s+2.471)(s^2+14.18)}\right\}.$$

The Bode diagrams of these controllers are depicted in Fig. 3.3(a). The resulting open-loop transfer functions (see Fig. 3.3(b)) show behavior similar to that of the delayed integral example from the previous chapter: the increase of the loop delay causes "oscillating" behavior of the open loop magnitude because the regulator has to provide more and more phase lead in the crossover region. In accordance with what



(a) Regulators (b) Open loop transfer functions

Figure 3.3: H^∞ loop-shaping designs for $P_\mathrm{m}(s)\mathrm{e}^{-sh}$

we saw in the Chapter 2, it brings the multiplication of gain crossover frequencies (from 1 in the first case to 5 in the last) and a significant decrease of the delay margin (from ± 0.1sec in the first case to ± 0.022 sec in the last). It is worth mentioning that the controller in the third case is by itself unstable, resulting in an encirclement of the critical point by the reculting loop to maintain the closed-loop stability.

Similarly to the classical loop-shaping design, the H^∞ loop-shaping design here might not be the best possible design. Our primary goal here was to show its transparent logic. Maybe the unstable controller in the third case is worth reconsidering. Yet the three resulting open loops present typical cases for system controlled by DTC regulators. We therefore prefer to keep it in this form for the implementation part.

The procedure above yields the control system setup shown in Fig. 3.4, where delayed plant $P_\mathrm{r}e^{-sh}$ is controlled by a regulator C. This overall regulator C consists of three parts: a rational weighting transfer

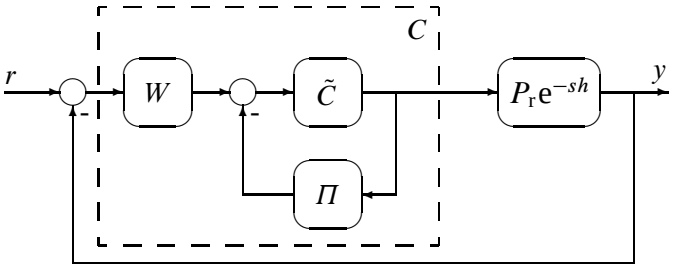


Figure 3.4: Control system setup for H^∞ loop shaping

function W, a rational primary controller $\tilde{C}$ and an irrational predictor block Π. The implementation of W and $\tilde{C}$ bears no problems in general because these transfer functions are rational. The implementation of the irrational predictor block Π, which involves unstable pole/zero cancellations, might not be trivial (see the discussion in the next section). Conceptually, the simplest solution to overcome this problem might be to approximate the predictor block Π by a rational transfer function. To this end, several methods can be used. For example, a Padé approximation of the continuous-time delay with some ad hoc adjustments that guarantee that all unstable poles are canceled after the approximation can be used. Another simple method is curve fitting. After the rational approximation of Π has been obtained, any conventional method can be used to implement it. Consider Fig. 3.5, which shows the Nichols charts of the open loop transfer functions (the case of $h = 0.15$) for analytically calculated DTC and its 10th order approximation. It is

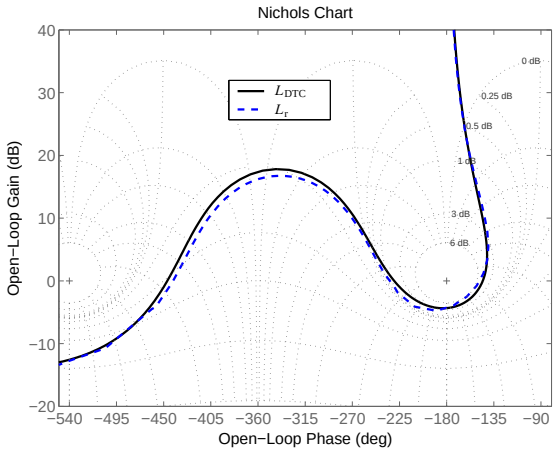


Figure 3.5: Loop transfer function for DTC controller and its rational approximation

definitely a feasible option to use, although such a method might have some drawbacks. For example, the resulting approximation of Π might be of a very high order. This work studies alternative implementation methods described below.

3.2 Analog implementation of DTC controllers

The irrational part of the controller in Fig. 3.4, i.e., its DTC block, can be in general described as

$$\begin{aligned}\Pi(s) &= \pi_h\{\hat{P}(s)\mathrm{e}^{-sh}\} := \tilde{P}(s) - \hat{P}(s)\mathrm{e}^{-sh} \\ &= C(sI-A)^{-1}\mathrm{e}^{-Ah}B - C(sI-A)^{-1}B\,\mathrm{e}^{-sh}\end{aligned} \tag{3.3}$$

for some rational $\hat{P}(s) = C(sI-A)^{-1}B$. Such blocks are conventionally referred to as *distributed-delay* (DD) blocks owing to their distributed-delay form:

$$\begin{aligned}\Pi(s) &= C\,\mathrm{e}^{-Ah}(sI-A)^{-1}\big(I-\mathrm{e}^{-(sI-A)h}\big)B \\ &= \int_0^h C\,\mathrm{e}^{-A(h-\sigma)}B\,\mathrm{e}^{-s\sigma}\,\mathrm{d}\sigma.\end{aligned} \tag{3.4}$$

Whereas the implementation of rational primary controllers is well-understood now, that of the irrational DD part might not be. If the transfer matrix $\hat{P}$ is stable, then so is $\tilde{P}$ and the DD element can be implemented as the difference between two stable transfer matrices $\tilde{P} - \hat{P}\mathrm{e}^{-sh}$, much like the classical Smith predictor. For unstable $\hat{P}$, however, the difference above contains unstable pole-zero cancellations and hence is not directly suitable for the implementation. The use of form (3.4), in which unstable poles of $\hat{P}$ and $\tilde{P}$ are canceled analytically, enables one to circumvent this problem, yet this form is not readily implementable using standard hardware.

As discussed in §1.2.3, there are essentially two approaches available in the literature to implement $\Pi(s)$: lumped delay approximations of the DD form in (3.4) (Mirkin, 2004) and the implementation of the difference $\tilde{P} - \hat{P}\mathrm{e}^{-sh}$ by incorporating a nonlinear resetting mechanism (Mondié *et al.*, 2001b). The primary purpose of this section is to review these approaches.

3.2.1 Lumped-delay approximations of distributed-delay elements

Lumped-delay approximation (LDA) approaches are motivated by the Riemann sum approximation of integrals. Assuming a uniform partitioning of the interval $[0, h]$ with the step $\frac{h}{\nu}$ we have:

$$\int_0^h C\,\mathrm{e}^{-A(h-\sigma)}B\,\mathrm{e}^{-s\sigma}\,\mathrm{d}\sigma \approx \frac{h}{\nu}\sum_{i=0}^{\nu}\eta_i C\,\mathrm{e}^{-A(1-i/\nu)h}B\,\mathrm{e}^{-sih/\nu}, \tag{3.5}$$

where real constants η_i depend on the approximation method. For example, the rectangular approximation yields $\eta_i = 1$, $i = 0, \ldots, \nu-1$, and $\eta_\nu = 0$; the trapezoidal approximation yields $\eta_i = 1$, $i = 1, \ldots, \nu-1$, and $\eta_0 = \eta_\nu = \frac{1}{2}$; etc. For each $s \in \mathbb{C}$ approximation (3.5) converges as $\nu \to \infty$. Yet it does not converge as a function of s. This might cause severe instability problems, as shown by van Assche *et al.* (1999) via numerical simulations. The reason, found in (Mirkin, 2004), is that approximation (3.5) does not converge in the high-frequency range. In fact, the left-hand side of (3.5) is strictly proper whereas the right-hand side is not for all ν.

To resolve this problem, Mirkin (2004) proposed a modification, guaranteeing that the approximation is always strictly proper. The idea can be illustrated as follows. It is clear that for any $\tau > 0$ and rational $G(s) = C(sI - A)^{-1}B$

$$\begin{aligned}
\Pi(s) &= \frac{\tau s+1}{\tau s+1}\boldsymbol{\pi}_h\{G(s)\mathrm{e}^{-sh}\} \\
&= \frac{1}{\tau s+1}\big((\tau s+1)(G_\alpha(s) - G(s)\mathrm{e}^{-sh})\big) \\
&= \frac{1}{\tau s+1}\left(\tau C\,\mathrm{e}^{-Ah}B - \tau CB\mathrm{e}^{-sh} + \int_0^h C(I+\tau A)\mathrm{e}^{-A(h-\sigma)}B\,\mathrm{e}^{-s\sigma}\,\mathrm{d}\sigma\right),
\end{aligned}$$

where $G_\alpha(s) = C\,\mathrm{e}^{-Ah}(sI - A)^{-1}B$. The approximation of the DD part in the parentheses can be then performed using the Riemann sum, resulting in the *strictly proper* (finite bandwidth) LDA

$$\begin{aligned}
\Pi(s) &\approx \frac{1}{\tau s+1}\left(\tau C\,\mathrm{e}^{-Ah}B - \tau CB\mathrm{e}^{-sh} + \frac{h}{\nu}\sum_{i=0}^{\nu}\eta_i C(I+\tau A)\mathrm{e}^{-A(1-i/\nu)h}B\,\mathrm{e}^{-sih/\nu}\right) \\
&= \frac{1}{\tau s+1}\sum_{i=0}^{\nu}\Pi_i\,\mathrm{e}^{-sih/\nu},
\end{aligned} \tag{3.6}$$

where the matrices Π_i are defined as

$$\Pi_i = \begin{cases} C(\frac{h}{\nu}\eta_i(I+\tau A) + \tau I)\mathrm{e}^{-Ah}B & \text{if } i = 0, \\ C(\frac{h}{\nu}\eta_i(I+\tau A) - \tau I)B & \text{if } i = \nu, \\ \frac{h}{\nu}\eta_i C(I+\tau A)\mathrm{e}^{-A(1-i/\nu)h}B & \text{otherwise.} \end{cases}$$

The block-diagram of this lumped-delay system is presented in Fig. 3.6. This scheme can be easily implemented by the dSPACE controller used in the laboratory provided the number of the delay blocks required there is not too big.

3.2.2 Implementation of distributed-delay elements using resetting mechanism

An alternative approach, called the *resetting mechanism* (RM), to the stable implementation of DD element was suggested in (Mondié *et al.*, 2001b). The idea is to use the lumped delay form (3.3), or in the time domain:

$$\begin{cases} \dot{x}(t) = Ax(t) + \mathrm{e}^{-Ah}Bu(t) - Bu(t-h), \\ y(t) = Cx(t), \end{cases} \tag{3.7}$$

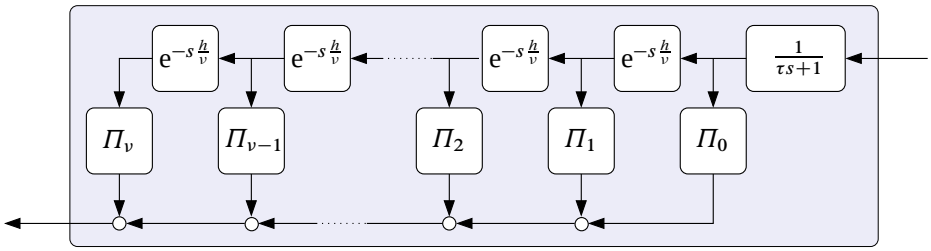


Figure 3.6: Lumped-delay approximation (LDA) of $\Pi(s)$

in combination with a periodic reset of the state vector x. The reset of the state vector clearly prevents the hidden modes in (3.7) to give rise to an unbounded grow in x. The straightforward reset of x, however, would alter the block $\Pi(s)$. To circumvent this problem, Mondié *et al.* (2001b) proposed to switch between two identical systems of the form (3.7) as shown in Fig. 3.7. Here the switch period is h and the

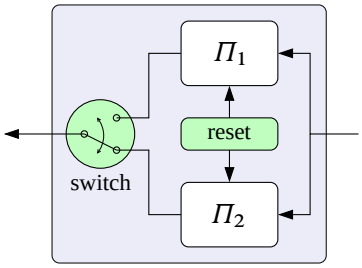


Figure 3.7: Resetting mechanism (RM) setup for implementing $\Pi(s)$

reset period for each system is $2h$ and shifted by h relative to each other. The scheme makes use the fact that (3.7), in fact, describes an FIR (finite impulse response) system the impulse response of which has support in $[0, h]$. This implies that only the input history of the length h affects the output of Π. Consider one working cycle of the system in Fig. 3.7 in the time interval $t \in [2kh, 2(k+1)h]$ for some integer k. At the beginning of this cycle, the output of Π is formed using the output of Π_1 and the second system is reset. By the middle of the cycle, at $t = (2k+1)h$, the state vector of Π_2 accumulates all history necessary to produce y. At this point, we switch the outputs, so that now the output of Π is now formed using the output of Π_2. At the same moment, the state vector of Π_1 is reset and starts to accumulate the history without affecting the output of Π. This accumulation is finished by the end of the cycle, at $t = 2(k+1)h$, and we can start the next cycle.

This scheme is conceptually simple and numerically stable, yet it might be complicated for the implementation using Simulink. The reason is that we need to incorporate a periodic state reset into Simulink blocks. It turns out that the only Simulink transfer function supporting this option is the integrator block. For this reason, systems of the form (3.7) have to be built from these elementary blocks. For example, in the SISO case the observer form of the state-space realization in (3.7) can be used as shown in Fig. 3.8 for

$$G(s) = \frac{b_{n-1}s^{n-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0} \quad \text{and} \quad G_\alpha(s) = \frac{\tilde{b}_{n-1}s^{n-1} + \cdots + \tilde{b}_1 s + \tilde{b}_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0}.$$

Resets of all integrator blocks here should be synchronized. Since this scheme cannot be applied to SIMO case, we have to use more general state-space form as shown in Fig. 3.9. A transfer matrix $I\frac{1}{s}$ is replaced here by the dash box. Since it is not possible to reset transfer matrix in Simulink, we split a bus signal to scalar ones and each signal connected to integrator block which could be reset, after then scalar signals are combined to bus again.

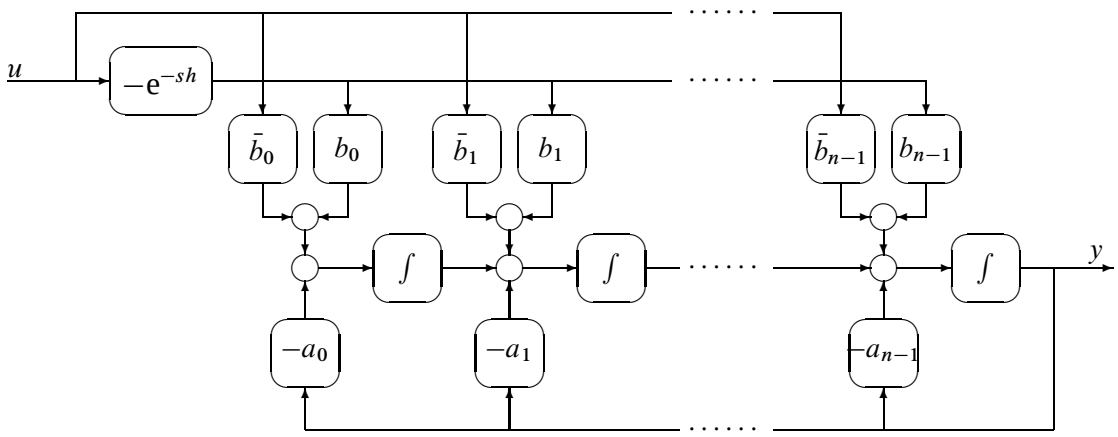


Figure 3.8: Realization of block Π using observer form

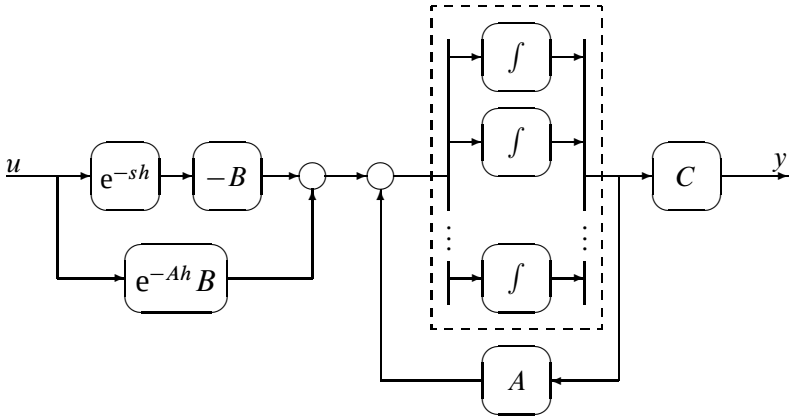


Figure 3.9: Realization of block Π using state-space form

3.2.3 Some comparisons

Both methods described above can be used for the DTC implementation.

- The LDA appear to be well-suited for low level programming controller based on micro-processors. Indeed, since this implementation scheme uses only static gains, delays and only one low pass filter, it can be a right choice for such a kind of hardware. The only problem could come from the choice of the discretization step (caused, for example, by high sensitivity of block Π to approximation or requirement of highly precision approximation), because the number of divisions is proportional to the number of logical operations needed to calculate the output signal.

- The second method has more complicated scheme and seems more suitable for controllers supporting high level programming languages, like Matlab/Simulink.

It is unclear at this point what other advantages and disadvantages can a designer have by choosing either of those implementation methods. To clarify this, some experiments have to be carried out. The rest of this chapter is devoted to the experimental validation of these methods and the analysis of their properties.

It is worth emphasizing that the comparison between these two implementation methods is complicated by the fact that the second method produces nonlinear controllers. Hence, the analytical calculation of the approximation error in the second method appears to be impossible. Also, the RM method of

(Mondié *et al.*, 2001b) has no tuning parameters, it either works or not and in the latter case it is unclear how the accuracy can be improved. For these reasons, the focus below is on the LDA method, the approximation errors in which can be quantified in the framework of the linear systems theory.

3.3 Analysis of the LDA method

Return now to the three DTC controllers designed in Section 3.1. They represent three typical forms of DTC controllers: decreasing loop gain with a single crossover frequency ($h = 0.1$ and weighting function (3.2a)), stable controller with multiple crossover frequencies ($h = 0.15$ and weighting function (3.2b)), and unstable controller with multiple crossover frequencies ($h = 0.2$ and weighting function (3.2c)). The purpose of this section is to address the implementation of these controllers using the LDA method presented in the previous section. Namely, we consider the implementation of the controller of the form depicted in Fig. 3.4 via "out of the box" use of the approximation technique from §3.2.1.

3.3.1 Out of the box implementation

We start with the literal implementation of (3.6). It has two design parameters that can be adjusted: the number of subintervals ν (h/ν is the discretization step) and the time constant of the low-pass filter τ. The rationale in the choice of ν is clear: its increase improves the approximation accuracy of block Π, but also increases the computational load. One would therefore prefer to keep ν as small as possible, provided it guarantees sufficiently high approximation accuracy. It turns out that the choice of τ does not have a crucial effect on the approximation accuracy and it is quite easy to pick the best τ for a given ν. We therefore concentrate here on the choice of ν.

There may be several approaches to measure the accuracy of the LDA $\Pi_{\mathrm{a}}(s)$ of $\Pi(s)$. Because both Π and its approximations are stable, an obvious choice is to measure the approximation accuracy by any system norm of the approximation error $\Pi - \Pi_{\mathrm{a}}$. This approach, however, would not take properties of the resulting closed-loop system into account. We then consider another approach, which is more in the line of the developments in the previous chapter. Namely, it is proposed here to measure the approximation accuracy by the deterioration of the delay margin of the resulting feedback system. On the one hand, this measure does take into account some important properties of the closed-loop system (like robustness and crossover frequencies). On the other hand, it is readily calculable. Speaking formally, let μ_{d} and $\mu_{\mathrm{d,a}}$ be the delay margins of the distributed-delay control law and its LDA, respectively. Then, the approximation accuracy is defined as

$$\delta = \left| 1 - \frac{\mu_{\mathrm{d,a}}}{\mu_{\mathrm{d}}} \right| .$$

In the DC motor example considered here we adopt the approximation level of $\delta \leq 0.001$, which corresponds to the delay margin deterioration of at most 0.1%. We then seek for a minimal ν guaranteeing this approximation accuracy level.

The results, presented in Table 3.1, are surprising. When h changes from 0.1 to 0.15 (50% increase),

h	0.1	0.15	0.2
ν	7	435	5460

Table 3.1: ν for LDA of Π for different h

the number of the discretization steps increases by more than the factor of 60. Furthermore, doubling the loop delay leads to the increase of ν by the factor of 780. One may argue that that there is no need in such accurate accuracy level ($\delta \leq 0.001$) and it should be possible to reduce ν without considerable loss

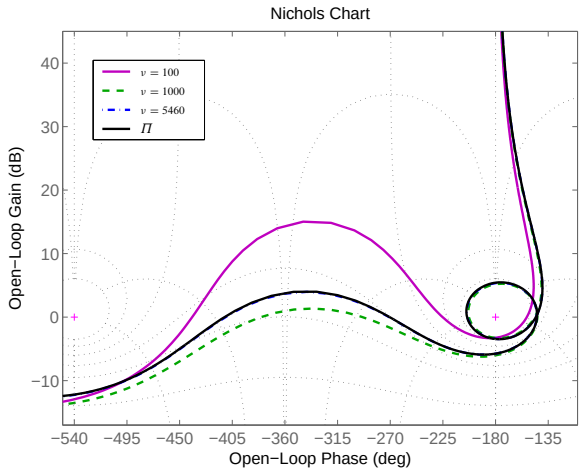


Figure 3.10: Nichols plot of L for Π and its approximations for different ν

of accuracy. This is not the case, however. To see this, let us look at the Nichols plot of the open loop depicted in Fig. 3.10, which shows the effect of the discretization step on the resulting open loop when $h = 0.2$. One can see that when $\nu = 100$ the resulting closed loop is actually unstable, when $\nu = 1000$ closed loop is stable, but the difference between analytically calculated open loop and its approximation is substantial. Only as ν reaches 5460, the approximation of Π becomes accurate enough to make the analytical and approximated loops virtually indistinguishable.

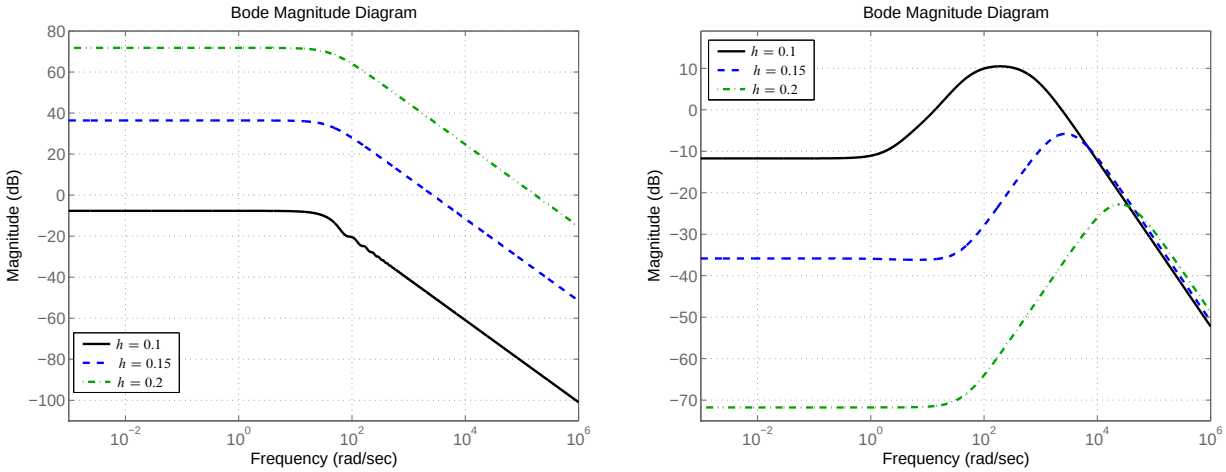
To conclude this subsection, the out of the box implementation of the LDA algorithm does not yield satisfactory results. The number of the discretization steps guaranteeing an acceptable approximation accuracy increases rapidly with the loop delay.

3.3.2 Inaccuracy mechanisms

To understand the reason of the vast increase of ν with the delay, let us return to the scheme of the DTC controller. As mentioned in §3.2, the controller (see Fig. 3.4) is the cascade of the rational weighting transfer function $W(s)$ and a closed-loop system with the loop transfer function $\tilde{C}\Pi$. The main point relevant to our discussion is that the closed-loop part cannot be implemented as one transfer function because the predictor Π is not rational and must be approximated. We therefore should analyze this part as the feedback interconnection of two different systems.

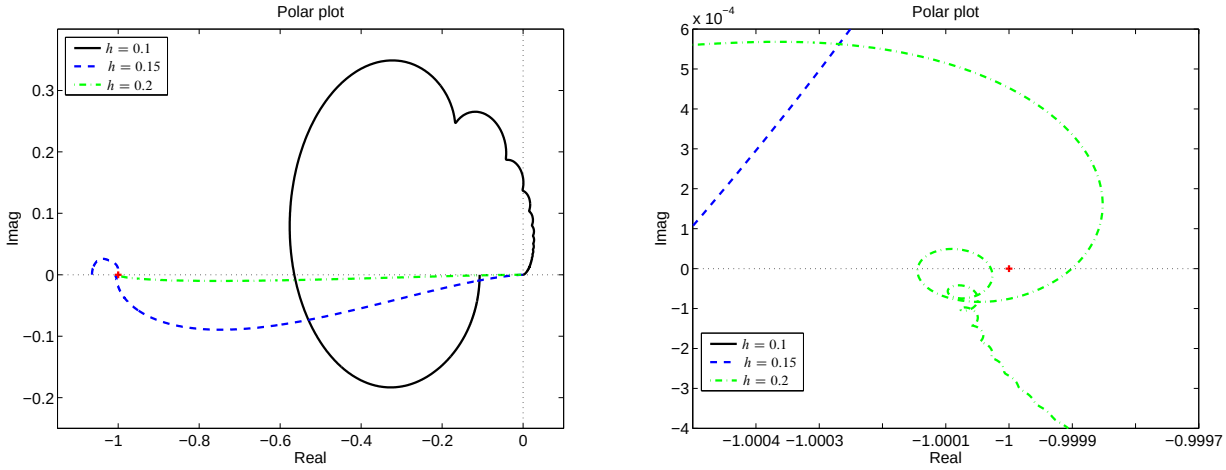
Consider each component the feedback part separately. Bode diagrams of the predictors Π and the central regulators $\tilde{C}$ for three studied cases depicted in Fig. 3.11. One can see that as h increases, the magnitude of the predictor block Π in the low-frequency range increases, whereas the magnitude of the primary controller $\tilde{C}$ decreases. The magnitude difference between them increases from approximately a factor of 2 (when $h = 0.1$) through ≈ 3000 ($h = 0.15$) to a factor of $\approx 10^7$ (for $h = 0.2$). This fact clearly gives rise to more significant round-off errors when the feedback part of the controller is implemented (because of the need to deal with signals having huge difference in their amplitudes).

To continue, consider the loop transfer function $\tilde{C}\Pi$, the polar plot of which is depicted in Fig. 3.12. A striking property of this loop transfer function is the deterioration of stability margins as h increases. The modulus margin μ_{mod} (defined as the minimal radius of the circle centered at $(-1, 0)$ and tangent to the polar plot of the loop transfer function) is representative in this respect: $\mu_{\text{mod}} = \{0.43, 4.9 \cdot 10^{-4}, 2.77 \cdot 10^{-5}\}$ in the cases when $h = \{0.1, 0.15, 0.2\}$, respectively. As we know, small stability margins make the closed-loop system sensitive to modeling mismatches. This is exactly what happens in our case as h increases:



(a) Predictor block Π (b) Primary controller $\tilde{C}$

Figure 3.11: Bode plots of $\tilde{C}$ and Π for different delays



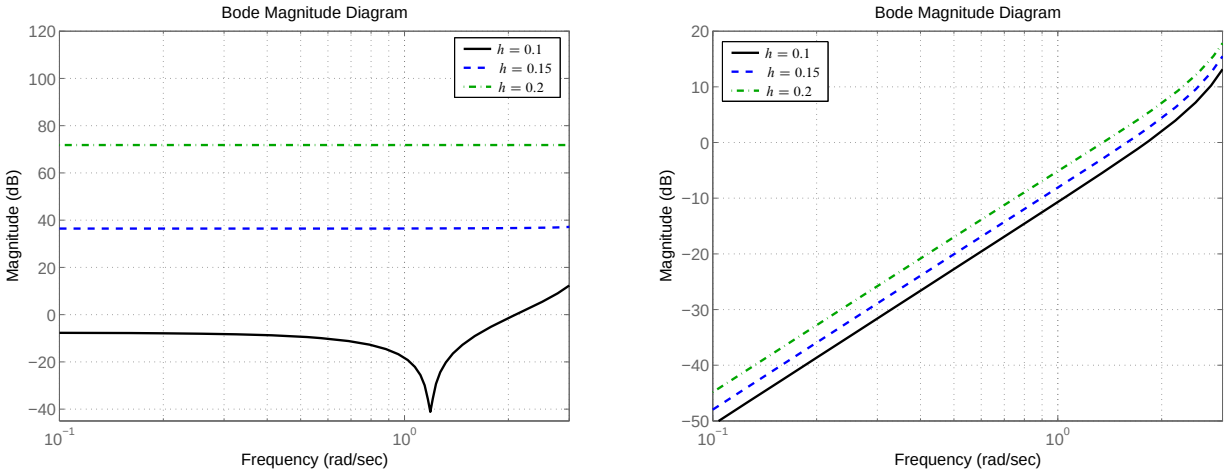
(a) Full-size plot (b) Polar plot enlarged at the $(-1, 0)$ region

Figure 3.12: Frequency response of $\tilde{C}\Pi$ for resulting regulators

stability margins decrease and we have to take more subintervals ν to satisfy more demanding requirements on the approximation accuracy. Combined with the increasing magnitude difference in involved signals discussed above, this fact causes the dramatic increase in the required number of steps.

The next step in the analysis of the problem reported in §3.3.1 is to understand the reason for the high magnitude of predictor frequency response. To this end, note that according to (3.3), the predictor consists of two parts: $\Pi = \tilde{P}(s) - \hat{P}(s)\mathrm{e}^{-sh}$. The Bode diagrams of each of these terms in the low-frequency range are shown in Figs. 3.13. These plots show clearly that the high magnitude of $\Pi(\mathrm{j}\omega)$ in the low-frequency range is caused by $\tilde{P}$. Inspecting $\tilde{P}$ and $\hat{P}$ in (3.3), one can see that the difference between them is the presence of the matrix exponential term e^{-Ah} in the "B" part of $\tilde{P}$. Elements of a matrix exponential might indeed become very large if its argument matrix has eigenvalues with large positive real parts. In our case, this might happen if A, which is the "A" matrix of both $\tilde{P}$ and $\hat{P}$, has "very negative" eigenvalues.

As shown in §2.5.2, the "A" matrix of $\tilde{P}$ and $\hat{P}$ is Hamiltonian, i.e., its eigenvalues are symmetric with



(a) $|\tilde{P}(\mathrm{j}\omega)|$ (b) $|\hat{P}(\mathrm{j}\omega)|$

Figure 3.13: Bode plots of the components of $\Pi = \tilde{P} - \hat{P}\,\mathrm{e}^{-sh}$ in the low-frequency range

respect to the imaginary axis (Zhou *et al.*, 1995). This means that the matrix A in (3.3) necessarily contains eigenvalues with negative real parts (unless it has only jω-axis eigenvalues, of course). The eigenvalues for our examples are shown in Tab. 3.2. One can see that there is a significant difference between the

h	eigenvalues of Ah			
0.1	$\pm$j0.37	$\pm$0.34	$\pm$1.01	
0.15	$\pm$j0.56	$\pm$0.43	$\pm$4.44 $\pm$ j0.16	
0.2	$\pm$j0.75	$\pm$0.49	$\pm$7.1	$\pm$7.93

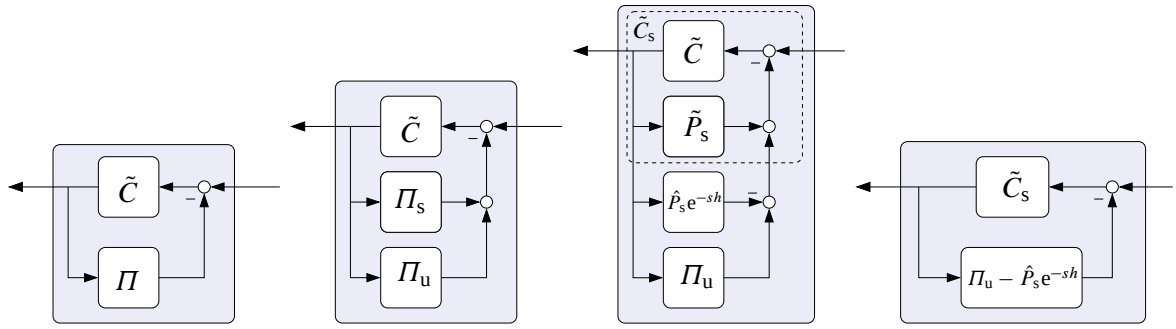
Table 3.2: Eigenvalues of resulting Hamiltonian matrices

rightmost eigenvalues of $-Ah$. The exponents of these eigenvalues are 1.4, 84.4, and 2762.8 for $h = 0.1$ $h = 0.15$, and $h = 0.2$, respectively. This leads to the increase of the amplitude of the predictor. Thus, the problem with the implementation of DTCs using LDA method is caused by the presence of fast stable pole(s) in the predictor blocks.

It is also worth mentioning that the high value of $-Ah$ can be the source of numerical problems in the computation of the controller. The computation of the matrix exponential of a matrix containing large positive eigenvalues is numerically unreliable (Golub and Van Loan, 1996). Although there are different approaches to improve the accuracy of calculation in this case (Golub and Van Loan, 1996), the problem still presents. Another aspect of this problem is the presence of the negative counterparts of large positive eigenvalues. The matrix exponential of a Hamiltonian matrix can then result in a matrix with a large condition number, so that the matrix might become close to singular. This causes unreliable results, then the coefficients of the resulting controller might make no sense.

3.3.3 Balancing controller loop: loop shifting

As mentioned in the previous paragraph, the problem with the implementation of DD elements using LDA is caused by *stable* poles of $\hat{P}$. At the same time, the need in the DD implementation of DTCs stems from the presence of *unstable* poles of $\hat{P}$. Indeed, the use of the DD form of $\Pi = \tilde{P} - \hat{P}\,\mathrm{e}^{-sh}$ (and, consequently, its LDAs) instead of the implementation of Π as the difference of two transfer functions is



(a) Original configuration (b) Split of Π (c) Embedding $\tilde{P}_\mathrm{s}$ into $\tilde{C}_\mathrm{s}$ (d) Final configuration

Figure 3.14: Loop shifting stages

caused by the need to prevent unstable pole-zero cancelations in the latter implementation. Yet, as stable poles of $\hat{P}$ are not a part of this cause, there is no need to include them into the procedure. This fact leads us to the idea of applying the LDA only to the part of Π including unstable poles of $\hat{P}$ and $\tilde{P}$.

The modified implementation is presented as the four steps below (see Fig. 3.14).

(a) Compute $\tilde{C}$ and Π.

(b) This stage deals with

$$\Pi = \tilde{P}(s) - \hat{P}(s)\mathrm{e}^{-sh} = C\,\mathrm{e}^{-Ah}(sI - A)^{-1}B - C(sI - A)^{-1}B\,\mathrm{e}^{-sh}.$$

There always exist (Golub and Van Loan, 1996) a similarity transformation such that the state matrix can be split to two matrices: A_s that includes the stable (open LHP) part of the spectrum of A and A_u that includes the anti-stable (closed RHP) part. In other words, there exists a nonsingular matrix T such that

$$\hat{P}(s) = \left[\begin{array}{c|c} A & B \\ \hline C & 0 \end{array}\right] = \left[\begin{array}{c|c} T^{-1}AT & T^{-1}B \\ \hline CT & 0 \end{array}\right] = \left[\begin{array}{cc|c} A_\mathrm{s} & 0 & B_\mathrm{s} \\ 0 & A_\mathrm{u} & B_\mathrm{u} \\ \hline C_\mathrm{s} & C_\mathrm{u} & 0 \end{array}\right] = \left[\begin{array}{c|c} A_\mathrm{s} & B_\mathrm{s} \\ \hline C_\mathrm{s} & 0 \end{array}\right] + \left[\begin{array}{c|c} A_\mathrm{u} & B_\mathrm{u} \\ \hline C_\mathrm{u} & 0 \end{array}\right]$$
$$=: \hat{P}_\mathrm{s}(s) + \hat{P}_\mathrm{u}(s)$$

where $\hat{P}_\mathrm{s}$ and $\hat{P}_\mathrm{u}$ are the stable and anti-stable parts of $\hat{P}$, respectively. It results in

$$\Pi = \tilde{P} - \hat{P}\,\mathrm{e}^{-sh} = (\tilde{P}_\mathrm{s} - \hat{P}_\mathrm{s}\mathrm{e}^{-sh}) + (\tilde{P}_\mathrm{u} - \hat{P}_\mathrm{u}\mathrm{e}^{-sh}) =: \Pi_\mathrm{s} + \Pi_\mathrm{u},$$

where Π_s (based on $\hat{P}_\mathrm{s}$) can be implemented as the difference of two transfer functions, whereas Π_u (based on $\hat{P}_\mathrm{u}$) should be implemented as a DD element using the LDA approach.

(c) Group the rational components $\tilde{C}$ and $\tilde{P}_\mathrm{s}$ and find a new primary controller $\tilde{C}_\mathrm{s} := \tilde{C}/(1 + \tilde{C}\,\tilde{P}_\mathrm{s})$.

(d) Implement the resulting controller.

Hereafter, this procedure will be referred to as the *loop shifting* since it rearranges the internal loop of the controller. The new loop (Fig. 3.14(d)) consists of the rational controller $\tilde{C}_\mathrm{s}$ and the irrational block

$$\Pi_\mathrm{irr}(s) := \Pi_\mathrm{u}(s) - \hat{P}_\mathrm{s}(s)\mathrm{e}^{-sh}.$$

The latter block contains a DD element Π_u and a delayed stable strictly proper system $\hat{P}_\mathrm{s}(s)\mathrm{e}^{-sh}$. When

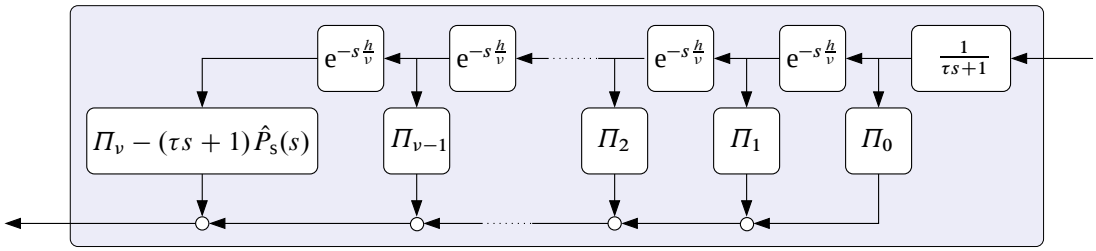
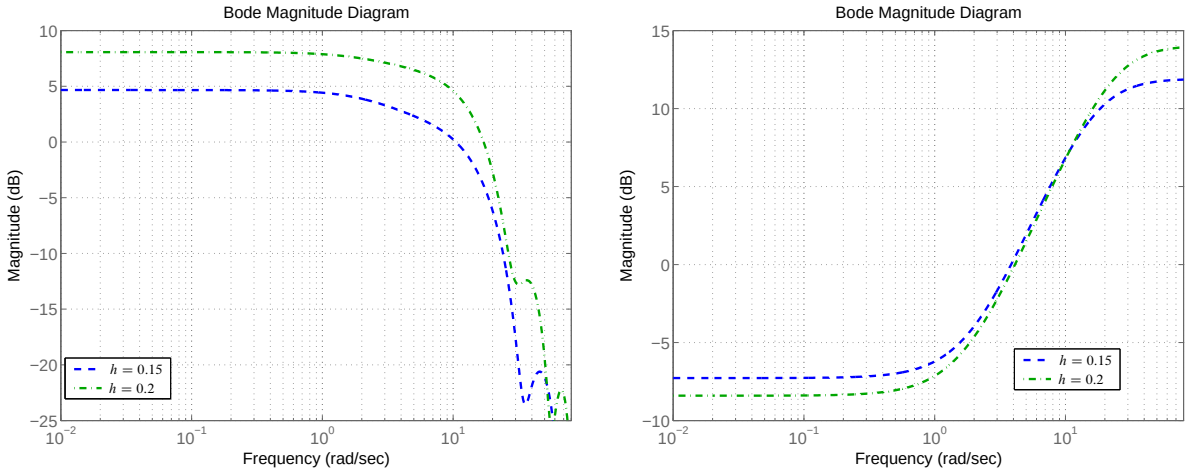


Figure 3.15: Lumped-delay approximation (LDA) of $\Pi_{\text{irr}}(s)$

the former is implemented by the LDA approach, the latter is naturally added to the last term of (3.6), the one corresponding to $i = \nu$. This results in the modification of the implementation scheme in Fig. 3.6 depicted in Fig. 3.15, where the coefficients Π_i correspond to the approximation of Π_u.

Consider how the application of the loop shifting procedure affects the controllers designed in §3.1.2. Fig. 3.16 shows the magnitudes of $\Pi_{\text{irr}}(j\omega)$ for $h = 0.15$ and $h = 0.2$ (i.e., for the problematic delays



(a) Magnitude of $\Pi_{\text{irr}}(j\omega)$ (b) Magnitude of $\tilde{C}_s$

Figure 3.16: Internal loop components after loop shifting

in §3.3.1). Comparing these plots with those in Fig. 3.11(a), one can see a significant decrease in the magnitude of the block to be approximated. This is clearly caused by the fact that Π_{irr} includes the matrix exponential of only a stable matrix, $e^{-A_u h}$. Thus, the application of the loop shifting idea results in more balanced internal controller loops. Also, the loop shifting leads to an increase in stability margins of the new controller loop. Fig. 3.17 shows the Nyquist plots of the resulting loop transfer functions, $\Pi_{\text{irr}}\tilde{C}_s$. The modulus margins under $h = 0.15$ and $h = 0.2$ are now $\mu_{\text{mod}} = 0.105$ and $\mu_{\text{mod}} = 0.355$, respectively, which is the increase by factors of ≈ 200 and ≈ 12800 comparing with the margins for the out of the box implementation. This implies that the sensitivity of the new loop to the approximation errors is lower now.

With the observations above, it is not a surprise now that the number of the discretization steps ν required to achieve at most 0.1% reduction in the resulting delay margin decreases considerably comparing with that in §3.3.1. The comparison is presented in Table 3.3 and demonstrates clearly the advantage of the loop shifting procedure proposed above. For $h = 0.15$ the requirements on ν can be relaxed by a factor of 17.4, while for $h = 0.2$—by a factor of 105, i.e., by more than two orders of magnitude.

Figs. 3.18–3.20 show the responses of the resulted closed-loop systems to the unit step reference signal. For small delays (Fig. 3.18) one can see a good match between numerical simulation results and

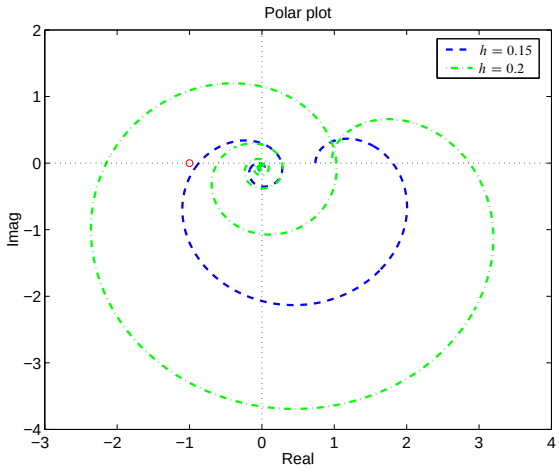
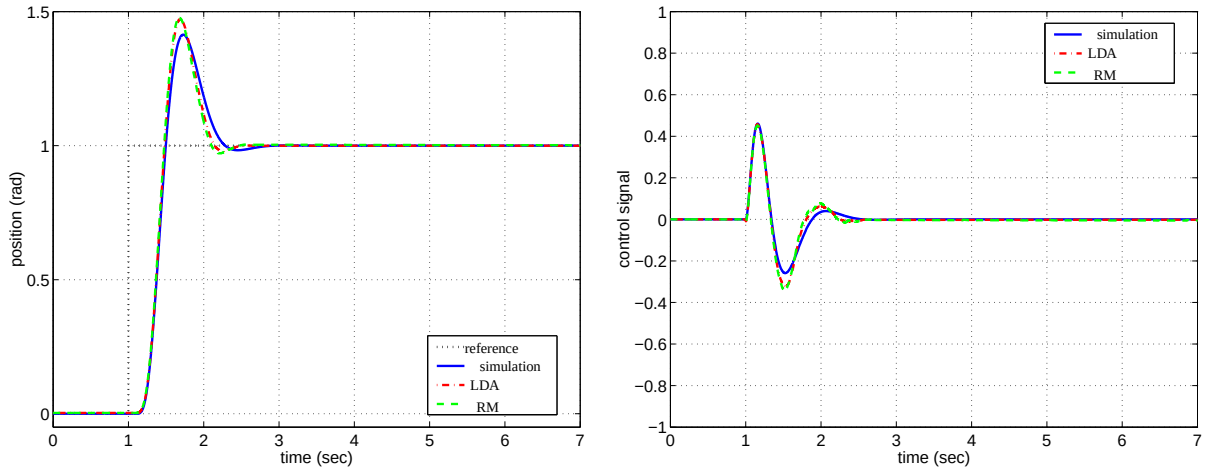


Figure 3.17: Frequency response of $\Pi_{\text{irr}}\tilde{C}_{\text{s}}$

h	0.15	0.2
$\nu_{\text{out of the box}}$	435	5460
$\nu_{\text{loop shifting}}$	25	52

Table 3.3: ν for LDA of Π for different h (loop shifting)



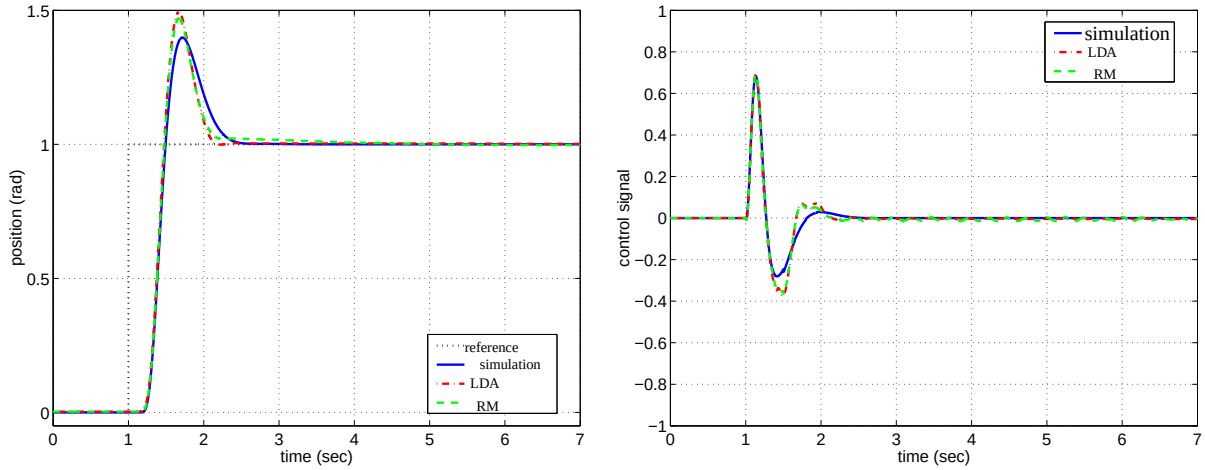
(a) Motor angular position

(b) Control signal

Figure 3.18: Step responses for $h = 0.1$

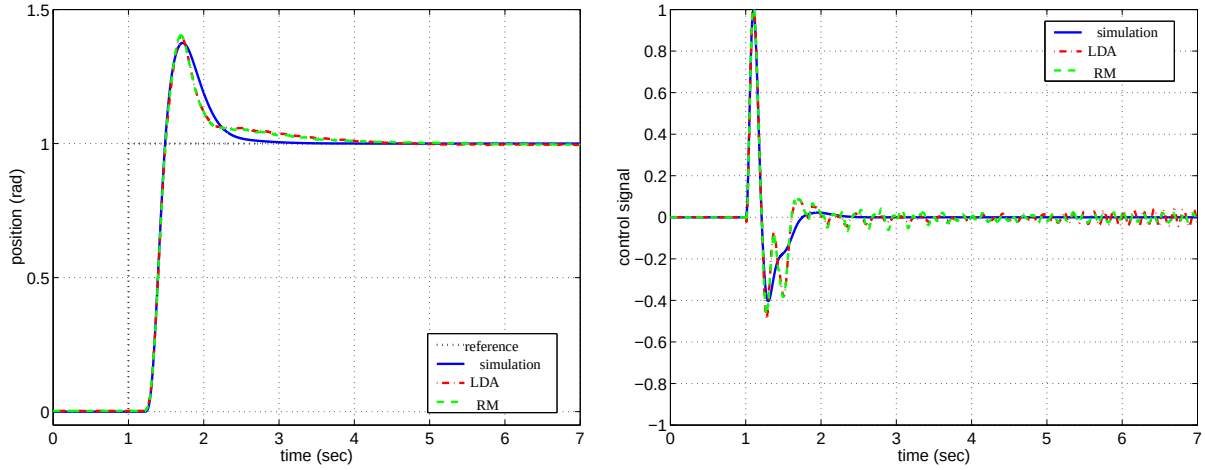
the response of the real experimental system. At $h = 0.15$ and, especially, $h = 0.2$ significant steady-state noise is present. It can be explained by hight high-frequency gains of the designed controllers in these cases (cf. Fig. 3.3). Consequently, the controllers are more sensitive to high-frequency modeling mismatches and measurement noise.

Remark 3.1 *Figs. 3.18–3.20 present not only the LDA, but also the RM implementation. In both cases the loop-shifting procedure is used. More detailed description of the RM implementation via Simulink is*



(a) Motor angular position (b) Control signal

Figure 3.19: Step responses for $h = 0.15$



(a) Motor angular position (b) Control signal

Figure 3.20: Step responses for $h = 0.2$

presented in Section B.1 in Appendix B. One can see that the RM implementation results are very close to the LDA results.

3.3.4 Fast stable dynamics of the H^∞ DTC

We saw that the loop shifting procedure proposed in the previous subsection is quite effective in improving the approximation accuracy of the LDA approach. In particular, it prevents matrix exponentials of matrices with large positive eigenvalues from appearing in the DD element. These exponentials are still a part of the rational primary controller $\tilde{C}_s$. It is therefore important to prevent such eigenvalues to appear at the controller design stage. This, in turn, requires the understanding of the source of these eigenvalues. In this subsection this issue will be addressed.

The H^∞ loop-shaping formulae from §2.5.2 are presented in terms of a state-space realization of the plant. These formulae are quite complicated, so that it might be difficult to trace the poles of $\hat{P}$ in

their terms. We thus use here an alternative expression for $\hat{P}$ derived in (Mirkin, 2003). To this end, remember that the H^∞ loop-shaping procedure is based on the solution of a standard H^∞ problem. More specifically, if $P_\mathrm{a} = P_\mathrm{m}W$ is the cascade of the plant and the weighting function, then the optimization problem is formulated for the generalized plant (McFarlane and Glover, 1990)

$$G = \begin{bmatrix} \begin{bmatrix} 0 \\ \tilde{M}_\mathrm{a}^{-1} \end{bmatrix} & \begin{bmatrix} I \\ P_\mathrm{a} \end{bmatrix} \\ \tilde{M}_\mathrm{a}^{-1} & P_\mathrm{a} \end{bmatrix},$$

where $\tilde{M}_\mathrm{a} \in RH^\infty$ is the denominator of the *normalized* left coprime factorization $P_\mathrm{a} = \tilde{M}_\mathrm{a}^{-1}\tilde{N}_\mathrm{a}$. The factorization is said to be normalized (McFarlane and Glover, 1990) if

$$\begin{bmatrix} \tilde{N}_\mathrm{a}(s) & \tilde{M}_\mathrm{a}(s) \end{bmatrix} \begin{bmatrix} \tilde{N}_\mathrm{a}^\sim(s) \\ \tilde{M}_\mathrm{a}^\sim(s) \end{bmatrix} = \tilde{N}_\mathrm{a}(s)\tilde{N}_\mathrm{a}^\sim(s) + \tilde{M}_\mathrm{a}(s)\tilde{M}_\mathrm{a}^\sim(s) = I,$$

where the *conjugate* of $G(s) = D + C(sI - A)^{-1}B$ is defined as $G^\sim(s) := G'(-s) := D' - C'(sI + A')^{-1}B'$ (in the scalar case $G^\sim(s) = G(-s)$).

As shown by Mirkin (2003), the H^∞ DTC block for each achievable performance level $\gamma > 1$ is based upon the following system:

$$\hat{P} = \mathcal{F}_u(G, \gamma^{-2}G_{11}^\sim), \tag{3.8}$$

where the upper linear fractional transformation is defined as

$$\mathcal{F}_u\left(\begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{bmatrix}, \Omega\right) := \Phi_{22} + \Phi_{21}(I - \Omega\Phi_{11})^{-1}\Omega\Phi_{12}.$$

Thus, (3.8) rewrites as follows:

$$\begin{aligned} \hat{P} &= P_\mathrm{a} + \gamma^{-2}\tilde{M}_\mathrm{a}^{-1}\left(I - \gamma^{-2}\begin{bmatrix} 0 & (\tilde{M}_\mathrm{a}^\sim)^{-1} \end{bmatrix}\begin{bmatrix} 0 \\ \tilde{M}^{-1} \end{bmatrix}\right)^{-1}\begin{bmatrix} 0 & (\tilde{M}_\mathrm{a}^\sim)^{-1} \end{bmatrix}\begin{bmatrix} I \\ P_\mathrm{a} \end{bmatrix} \\ &= P_\mathrm{a} + \gamma^{-2}\tilde{M}_\mathrm{a}^{-1}\big(I - \gamma^{-2}(\tilde{M}_\mathrm{a}^\sim)^{-1}\tilde{M}_\mathrm{a}^{-1}\big)^{-1}(\tilde{M}_\mathrm{a}^\sim)^{-1}P_\mathrm{a} \\ &= \Big(I + \gamma^{-2}(\tilde{M}_\mathrm{a}^\sim\tilde{M}_\mathrm{a})^{-1}\big(I - \gamma^{-2}(\tilde{M}_\mathrm{a}^\sim\tilde{M}_\mathrm{a})^{-1}\big)^{-1}\Big)P_\mathrm{a} \\ &= \big(I - \gamma^{-2}(\tilde{M}_\mathrm{a}^\sim\tilde{M}_\mathrm{a})^{-1}\big)^{-1}P_\mathrm{a}. \end{aligned}$$

It follows from the fact that the coprime factorization of P_a is normalized that

$$P_\mathrm{a}P_\mathrm{a}^\sim + I = (\tilde{M}_\mathrm{a}^\sim\tilde{M}_\mathrm{a})^{-1}.$$

Hence, we finally obtain:

$$\hat{P} = \big((1 - \gamma^{-2})I - \gamma^{-2}P_\mathrm{a}P_\mathrm{a}^\sim\big)^{-1}P_\mathrm{a} = \gamma^2\big((\gamma^2 - 1)I - P_\mathrm{a}P_\mathrm{a}^\sim\big)^{-1}P_\mathrm{a}.$$

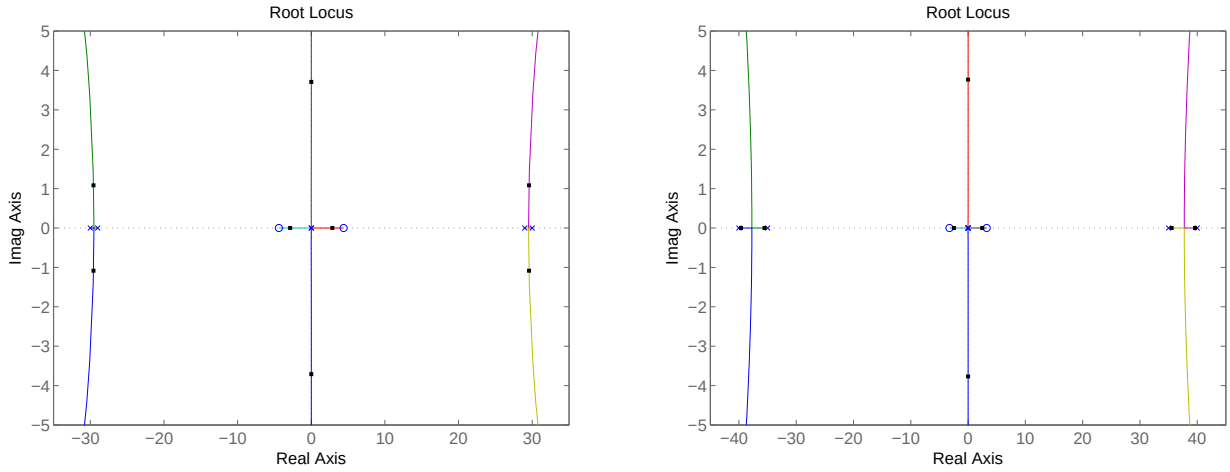
Thus, the poles of $\hat{P}$, which are the eigenvalues of the Hamiltonian matrix that we study here, satisfy the equation

$$\gamma^2 - 1 - P_\mathrm{m}(s)W(s)W(-s)P_\mathrm{m}(-s) = 0.$$

This can be equivalently presented in the following form:

$$1 - kP_\mathrm{m}(s)W(s)W(-s)P_\mathrm{m}(-s) = 0, \qquad \text{where } k := \frac{1}{\gamma^2 - 1} > 0. \tag{3.9}$$

This is reminiscent of the symmetric root-locus form (Bryson, 1999) that appears in the LQR control modulo the sign of the "feedback" gain k. When γ decreases from $+\infty$ to the optimal $\gamma_{\mathrm{opt},h} > 1$, the gain k increases from 0 to the finite value $k_{\mathrm{max},h} := 1/\sqrt{\gamma_{\mathrm{opt},h}^2 - 1}$. According to standard root-locus rules (Franklin *et al.*, 2002), when k varies from 0 to $+\infty$ the roots of (3.9) start at the poles of $P_{\mathrm{m}}(s)W(s)W(-s)P_{\mathrm{m}}(-s)$ and then some of them approach the zeros of this transfer function and the others go to the infinity along with the asymptotes centered at the origin and directed according to the pole excess of $P_{\mathrm{m}}(s)W(s)W(-s)P_{\mathrm{m}}(-s)$. Depending on whether the degree of $P_{\mathrm{m}}W$ is odd or even, the root locus is a "0°" (negative) or "180°" (positive) root locus, respectively. Although k never reaches $+\infty$, the root locus associated with (3.9) is informative because it shows the trend.



(a) $P_{\mathrm{a}} = P_{\mathrm{m}}W_2$ ($h = 0.15$) (b) $P_{\mathrm{a}} = P_{\mathrm{m}}W_3$ ($h = 0.2$)

Figure 3.21: Root loci for the poles of $\hat{P}$

Consider the root loci in Fig. 3.21, which are obtained for the systems $P_{\mathrm{m}}W_2$ and $P_{\mathrm{m}}W_3$, where the plant P_{m} is given by (3.1) and the weighting functions W_2 and W_3—by (3.2b) and (3.2c), respectively. In both cases the pole excess of P_{a} is 3, so that the loci are standard positive symmetric loci having 6 asymptotes with the angles $\pm 30°, \pm 90°, \pm 150°$. Small black rectangles on the loci denote the points corresponding to $k_{\mathrm{max},h}$. It is readily seen that the two leftmost poles are caused by the leftmost poles of P_{a}, which are actually the poles of W_i. Indeed, the two leftmost branches are always located to the left from the second fastest pole of W_2 and W_3 (at -29 and -35, respectively). This implies that the problems in approximating the H^∞ DTC for our system are caused by the two fast poles of the weighting functions.

3.3.5 The use of non-proper weights

Recognizing that the fast stable eigenvalues of the Hamiltonian matrix are caused by the weighting functions, W, it is then natural to attempt to prevent these eigenvalues to appear via a different choice of W. To this end, note that the main reason for adding these fast poles to W is to ensure that the high-frequency gain of $W(\mathrm{j}\omega)$ and, consequently, of the resulting controller, is not too high.

In this subsection a different approach to limit the high-frequency gain of the controller is proposed. Instead of imposing this via W, one may attempt to add a low-pass filter *after* the H^∞ loop-shaping procedure is completed, much like in the classical loop shaping. This implies that in the design stage the weighing function might be chosen without the high-frequency gain considerations. The weighting function needs not even to be proper. Indeed, the method requires only the auxiliary plant, $P_{\mathrm{a}} := P_{\mathrm{m}}W$, to be proper.

Technically, the proposed modification is quite simple. It contains only two modification comparing with the procedure described in §3.1.2:

1. the rationale for the choice of W does not include the high-frequency gain considerations,
2. the final controller C in Fig. 3.4 is augmented not only by W, but also by a low-pass filter F.

The choice of the low-pass filter F is rather straightforward, although it might happen that a required F harms the resulting controller too much by altering the loop in the high-frequency range. In this case, one needs to reconsider the weighting function W, which might not be trivial.

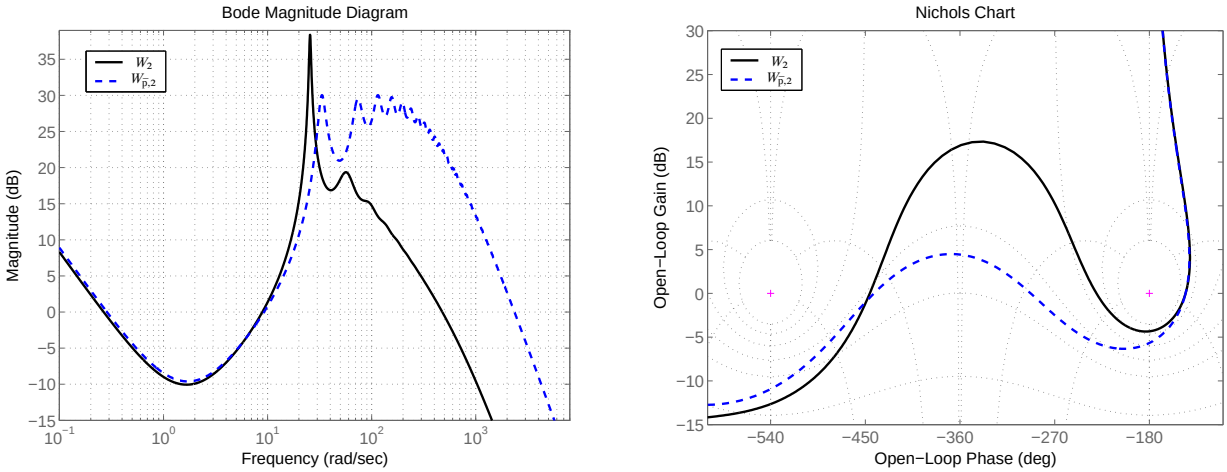
Applying this procedure to the DC motor servo-controller design for $h = 0.15$ and $h = 0.2$, the following modifications of the weighting functions and the corresponding low-pass filters were chosen:

$$W_{\bar{\mathrm{p}}2}(s) = \frac{0.115(s+1.4079)(s+5.291)}{s}, \quad F_2(s) = \frac{1}{(0.0065s+1)^2},$$
$$W_{\bar{\mathrm{p}}3}(s) = \frac{0.179(s+1.4079)(s+3.8462)}{s}, \quad F_3(s) = \frac{1}{(0.003s+1)^2}.$$

These functions result in the following controllers:

$$\tilde{C}_{\bar{\mathrm{p}}2}(s) = \frac{2138.6896(s+2.095)}{(s+15.55)(s+699.2)}, \qquad \Pi_{\bar{\mathrm{p}}2}(s) = \pi_h\left\{\frac{7.3378(s+5.291)s^2\,\mathrm{e}^{-0.15s}}{(s-3.054)(s+3.054)(s^2+13.98)}\right\},$$
$$\tilde{C}_{\bar{\mathrm{p}}3}(s) = \frac{2753.9629(s+1.913)}{(s+19.73)(s+706.8)}, \qquad \Pi_{\bar{\mathrm{p}}3}(s) = \pi_h\left\{\frac{11.0496(s+3.846)s^2\,\mathrm{e}^{-0.2s}}{(s-2.693)(s+2.693)(s^2+14.23)}\right\}.$$

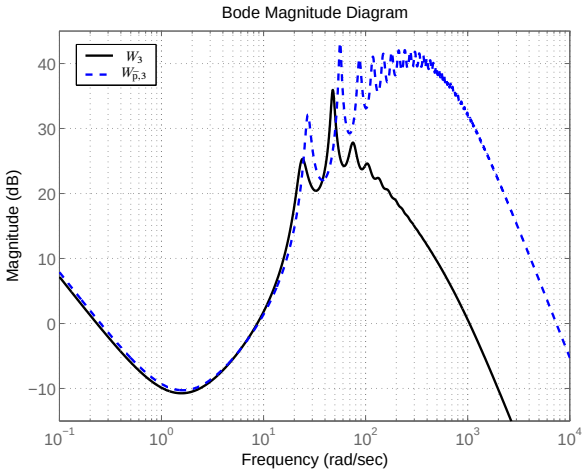
Comparing these weighting functions with those in (3.2b) and (3.2c), we can see that the lag (PI) parts remain unchanged, whereas the lead parts are replaced with ideal PD elements and additional low-pass poles are omitted. In the choice of the PD part the main criterion was the reproduction of the low-frequency gain of the resulting controller (this implicitly guarantees that the performance specifications of Section 3.1 are met). The resulting controllers and loop frequency responses are shown in Figs. 3.22 and



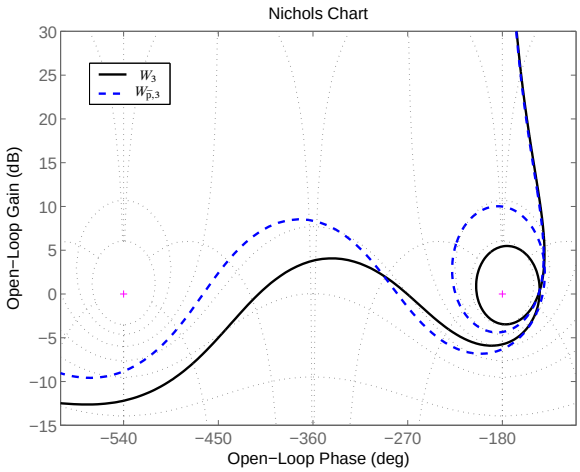
(a) Controller magnitude (b) Loop frequency response

Figure 3.22: Designs with strictly proper (W_2) and non-proper ($W_{\bar{\mathrm{p}}2}$) weights ($h = 0.15$)

3.23 together with the frequency responses for the weighting functions from §3.1.2. Up to the frequency $\omega = 10$ rad/sec there is a good match of the resulting controllers and, consequently, the loop frequency



(a) Controller magnitude

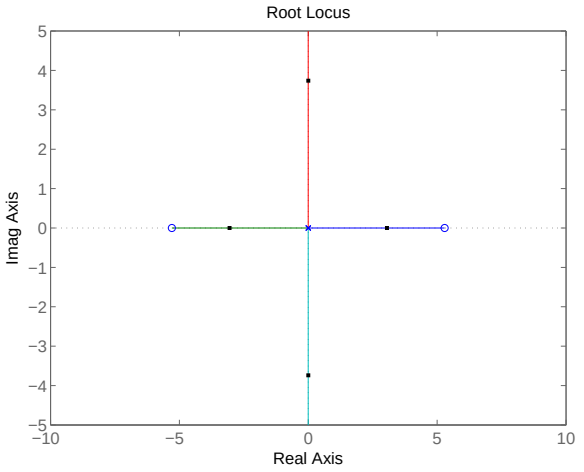


(b) Loop frequency response

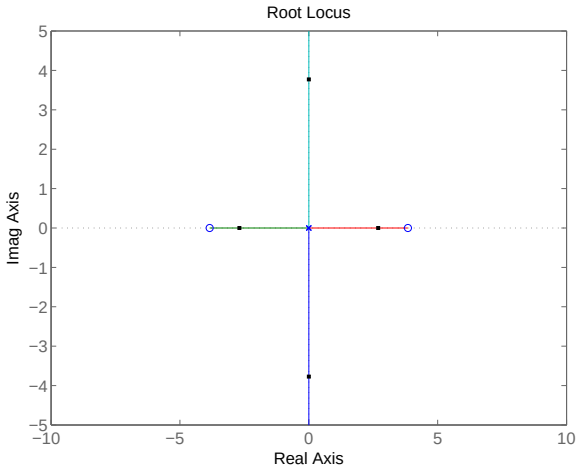
Figure 3.23: Designs with strictly proper (W_3) and non-proper ($W_{\bar{p},3}$) weights ($h = 0.2$)

responses. The difference is in the high-frequency range: controllers designed with non-proper weights have higher gain there than those designed for strictly proper weights.

The advantage of the use of non-proper W is that it prevents the fast stable eigenvalues to appear in the "A" matrix of the resulting DTC. This is clearly seen in the root loci of the resulted design, shown in Fig. 3.24. The pole excess of $P_\mathrm{a} = P_\mathrm{m} W_{\bar{p}i}$ is now 1, so that there is only a pair of asymptotes going



(a) $P_\mathrm{a} = P_\mathrm{m} W_{\bar{p},2}$ ($h = 0.15$)



(b) $P_\mathrm{a} = P_\mathrm{m} W_{\bar{p},3}$ ($h = 0.2$)

Figure 3.24: Root loci for the poles of $\hat{P}$ with non-proper weighting functions

along with the jω-axis and the real parts of the poles of $\hat{P}$ never go to the left of the leftmost zeros of $W_{\bar{p}i}$ (-5.291 and -0.864 for $i = 2$ and $i = 3$, respectively). Small black rectangles on the loci again denote the points corresponding to the maximal achievable gain, $k_{\max,h}$. As a result, the approximation of the DD block is simplified, which is reflected in Table 3.4.

A negative outcome of the procedure described above is the significant increase of the high-frequency gain of the resulting controller. Although this is not a part of the specifications posed in Section 3.1, such controllers might be undesirable because they would excite high-frequency unmodeled dynamics

h	0.15	0.2
$\nu_{\text{out of the box}}$	435	5460
$\nu_{\text{loop shifting}}$	25	52
$\nu_{\text{non-proper weight}}$	17	26

Table 3.4: ν for LDA of Π for different h (non-proper weights)

and measurement noise. This indeed happens in the case of $h = 0.2$, the control signal in steady state experiences rather visible oscillations. This renders the control system in Fig. 3.23 completely impractical.

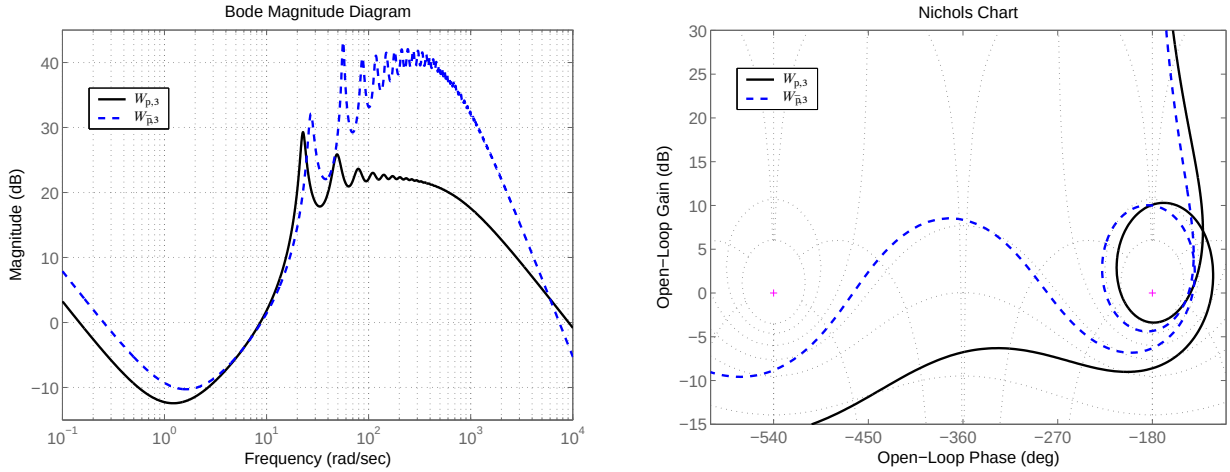
To decrease the bandwidth, an additional pole is introduced to the weighting transfer function $W_{\bar{\text{p}},3}$. The resulting weight and low-pass filter for the new design are:

$$W_{\text{p},3}(s) = \frac{2.626(s+1.4079)(s+1.335)}{s(s+6.667)}, \quad F_{\text{p}}(s) = \frac{1}{0.005s+1}$$

and the controller contains

$$\tilde{C}_{\text{p},3}(s) = \frac{3478.9654(s+7.495)(s+1.062)}{(s+1.201)(s^2+709.9s+2.284\cdot 10^5)},$$
$$\Pi_{\text{p},3}(s) = \boldsymbol{\pi}_h\left\{\frac{157.7264(s-6.667)(s+1.335)s^2\,\mathrm{e}^{-0.2s}}{(s+7.677)(s-7.677)(s+1.285)(s-1.285)(s^2+16.15)}\right\}.$$

This controller and the resulting loop are presented in Fig. 3.25. It is seen that the controller designed



(a) Controller magnitude

(b) Loop frequency response

Figure 3.25: Designs with proper ($W_{\text{p},3}$) and non-proper ($W_{\bar{\text{p}},3}$) weights ($h = 0.2$)

for $W_{\text{p},3}$ is less aggressive in the high-frequency range than that designed for $W_{\bar{\text{p}},3}$. At the same time, the use of $W_{\text{p},3}$ results also in a decrease of the low-frequency gain of the controller, although the gain in the crossover range remains effectively the same. The addition of a low-pass pole to the weighting function does not give rise to the problems reported in §3.3.4 because this additional pole is not as fast as the poles of W_3 in (3.2c). This is confirmed by the corresponding root locus in Fig. 3.26, which shows that the fastest stable pole of $\hat{P}$ is always located to the right from the "−8" level for all $k < k_{\max,h}$ (cf. the black rectangles in Fig. 3.26). Step responses of this experiment are shown in Figs. 3.27 and 3.28. The results are very similar to those in Figs. 3.19 and 3.20, respectively. Like in the previous examples, both LDA and RM methods are used.

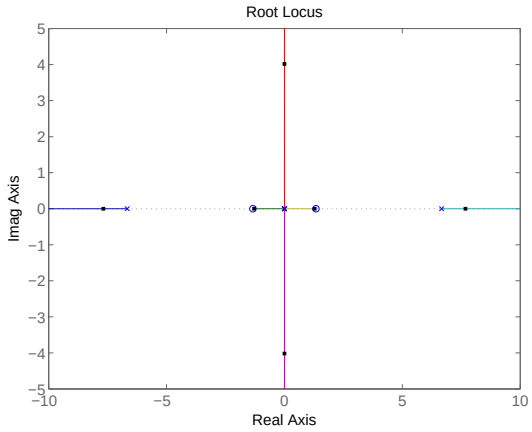
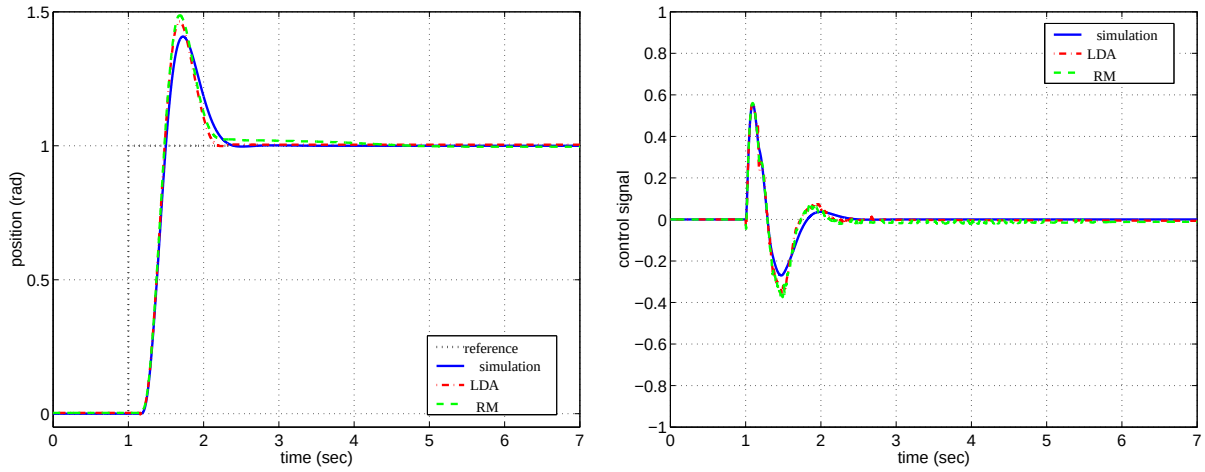


Figure 3.26: Root loci for the poles of $\hat{P}$ with proper weighting function $W_{\mathrm{p},3}$ ($h = 0.2$)



(a) Motor angular position

(b) Control signal

Figure 3.27: Step responses for $h = 0.15$ (design with $W_{\mathrm{p}2}^{-}$)

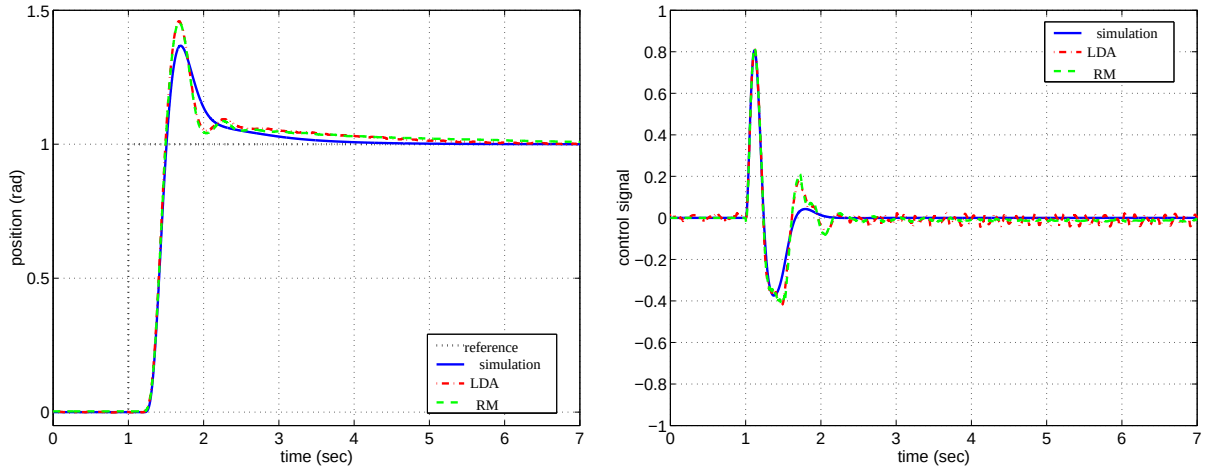
3.4 Control of the laboratory pendulum

So far, we have studied servo-controllers designed for a DC motor with loop delays. In this section two more challenging examples are addressed in brief. We consider the (complete) pendulum experiment, which is described in Appendix A, in both "down" (crane) and "up" (inverted pendulum) positions an show that DTCs with FIR blocks can be successfully implemented in these cases as well.

From the control viewpoint, the plants here have one control input and two measurable outputs: the motor angular position y and the pendulum angle θ. Controllers, thus, have two components,

$$C(s) = \begin{bmatrix} C_y(s) & C_\theta(s) \end{bmatrix},$$

acting on each of the two measurements. In both cases the input (control) delay, $h = 0.1$ for the crane and $h = 0.2$ for the inverted pendulum, is introduced artificially. Specifications include the internal stability of the closed-loop system, an integral action in the motor channel, and a reasonable crossover frequency. In the implementation, loop shifting was always used to reduce the sensitivity of the approximation of the resulting DTC blocks Π.



(a) Motor angular position

(b) Control signal

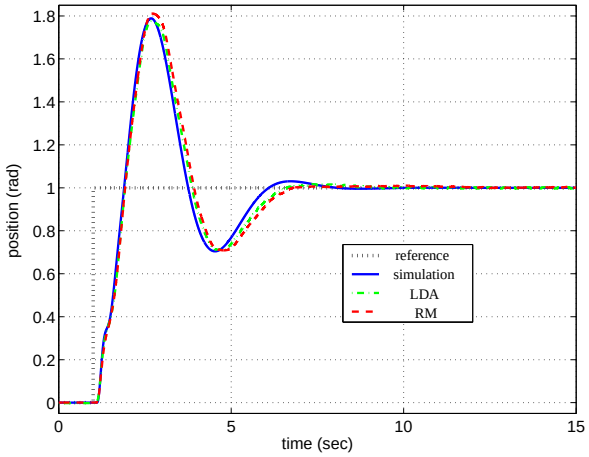
Figure 3.28: Step responses for $h = 0.2$ (design with $W_{\mathrm{p},3}$)

3.4.1 Pendulum

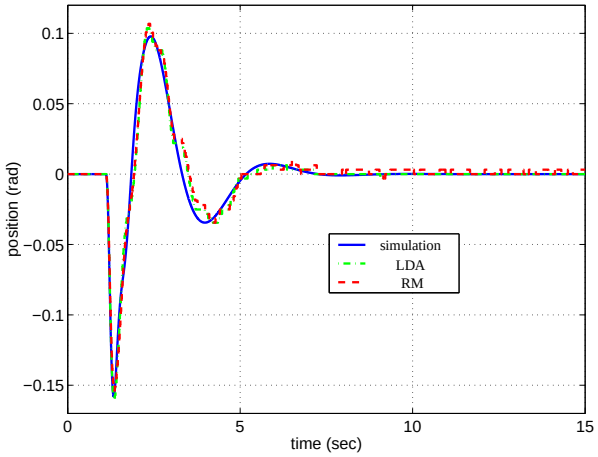
The transfer function of this system is given by (A.9). This is a stable system having a pair of lightly damped poles (pendulum) with the natural frequency $\omega_{\mathrm{n}} \approx 6.2$. The problem is to dampen the oscillations caused by these poles while maintaining reasonably good servo characteristics of the motor channel. The resulting time responses are presented in Fig. 3.29.

3.4.2 Inverted pendulum

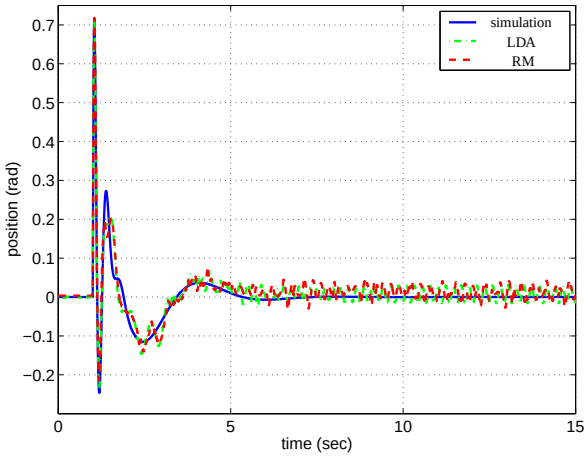
The transfer function of this system is given by (A.10). This is an unstable non-minimum phase system. The problem is to stabilize the inverted pendulum while maintaining reasonably good servo characteristics of the motor channel. This is a quite challenging task, taking into account the delay of $h = 0.2$ in the control channel. The resulting time responses, presented in Fig. 3.30, show that the use of a DTC-based controller makes these goals feasible.



(a) Motor angular position y

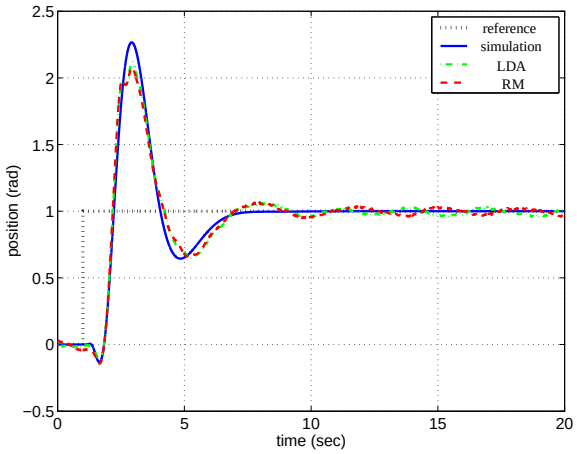


(b) Pendulum angle θ

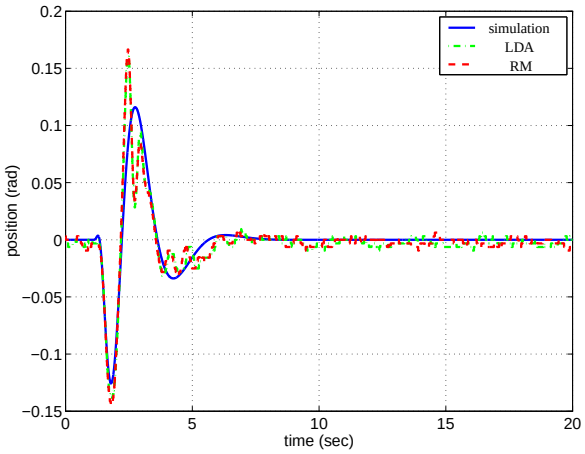


(c) Control signal

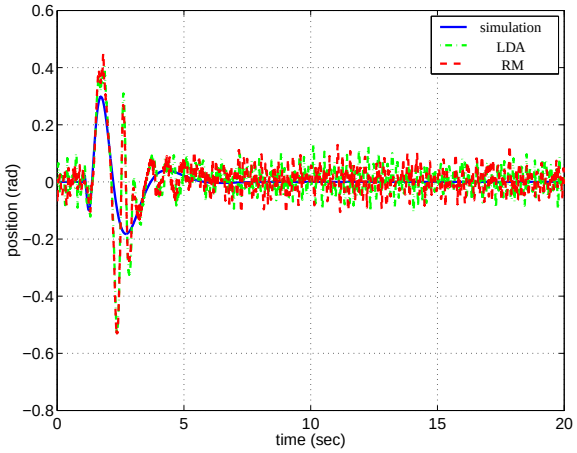
Figure 3.29: Step responses for the pendulum experiment ($h = 0.1$)



(a) Motor angular position y



(b) Pendulum angle θ



(c) Control signal

Figure 3.30: Step responses for the inverted pendulum experiment ($h = 0.2$)

Chapter 4

Conclusions and future research

This work has studied some aspects of the robust control of dead-time systems using dead-time compensators (DTCs). The main emphasis has been placed on the robustness of these controllers to uncertainties in the loop delay and on the robust implementation of the resulting (irrational) controllers.

It is known that DTC-based controllers are prone to be sensitive to variations of the loop delay. Reasons of this sensitivity, however, are not well studied in the control literature. Chapter 2 presents one of the first attempts to explain this phenomenon. Its contributions can be summarized as follows.

- The idea to investigate *underlying reasons* for weak robustness of DTCs to uncertainties in the loop delay via the classical Nyquist criterion arguments has been put forward.
- Using this approach, the delay margin of DTCs has been demonstrated to be a discontinuous function of the system parameters. More precisely, there may be points in the parameter space at which the delay margin deteriorates dramatically.
- The parametric discontinuity has been linked with the *crossover proliferation* phenomenon, which is an increase of the number of crossover regions of the resulted loop frequency response. The crossover proliferation has been shown to be triggered by the use of DTCs.
- Design guidelines to avoid crossover proliferation have been proposed.

Dead-time compensators are intrinsically infinite-dimensional systems that frequently (e.g., when the plant is unstable) involve distributed-delay (DD) elements. The implementation of DD elements using digital hardware might not be straightforward. Chapter 3 presents apparently the first investigation and practical validation of the feasibility of implementation methods available in the literature. The main contributions here are as follows.

- It has been demonstrated that the out-of-the-box use of the lumped-delay approximation method might not be feasible for the implementation even on relatively powerful hardware. The reasons for this have been demonstrated to be the ill-posedness of the internal loop of the controller and a significant growth of the gain of the DTC block, which, in turn, is caused by its fast stable modes.
- It has been shown that implementation problems can be substantially alleviated by rearranging the internal loop (loop shifting). The proposed loop shifting procedure is both conceptually and computationally simple. It is based on extracting stable rational part, which is the main source of numerical problems, from the DTC block and absorbing this part into the primary controller.
- In addition, the source of the fast stable modes of the DTC block under the H^∞ loop-shaping design has been shown to be the presence of fast stable modes of the weighting function. As a possible

remedy, the use of non-proper weighted functions has been proposed, which might further improve the robustness of the controller internal loop.

The proposed solutions have been validated by laboratory experiments (DC motor and pendulum experiments).

Some important questions related to the delay robustness of DTCs and the implementation of DD control laws are left unaddressed. It, for example, is not completely clear how the avoidance of the crossover proliferation can be incorporated to analytic design procedures (though some insight that could be helpful was provided in Section 2.3). Indeed, most analytic methods are formulated in terms of the closed-loop transfer functions, whereas the crossover notion is intrinsically open loop. It follows from the argumentation in Section 2.3 that it might be required to shape not only the magnitude of the closed-loop transfer function, but also its phase. This might be a non-trivial problem. DD controllers can, in principle, be approximated by rational transfer functions. Such an approximation, however, is not the conventional one described in §1.1.2 because it should maintain the exact cancellation of unstable poles and zeros of the DTC block. The pole-zero cancellation requirement should add interpolation constraints to rational approximation methods. These issues may be interesting subjects of future research.

Appendix A

Laboratory experiment

A.1 Experiment description

The testbed used to validate control algorithms is a laboratory pendulum. It is a 4-order SIMO system which can be used as a pendulum (a system with a pair of lightly damped poles) or as an inverted pendulum (non-minimum phase unstable system). When the pendulum is taken off, the resulting system is a DC motor with an axial load.

General view of the experiment system is depicted in Figs. A.1 and A.2. The pendulum, 1, is

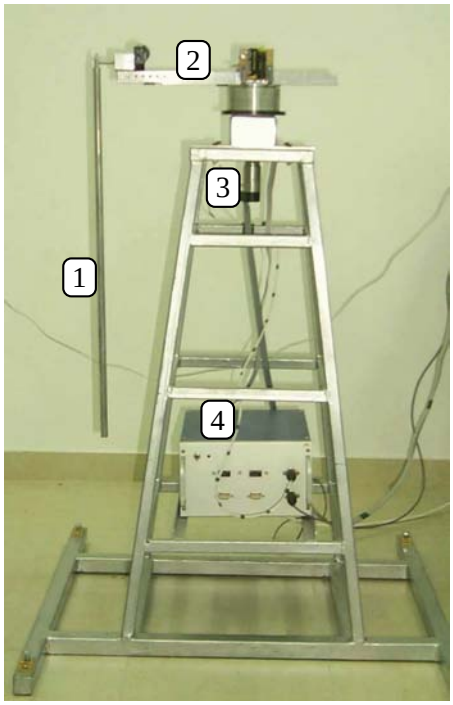


Figure A.1: The experiment system

mounted on a revolving platform, 2, driven by a DC motor, 3. More detailed view of the upper part is shown in Fig. A.3. Two variables are measured: the first encoder, 7, mounted on the motor shaft measures the angle of the platform. The second encoder, 6, mounted on the pendulum rotation shaft measures the pendulum angle. To avoid problems with wires transferring signals from the pendulum encoder and to

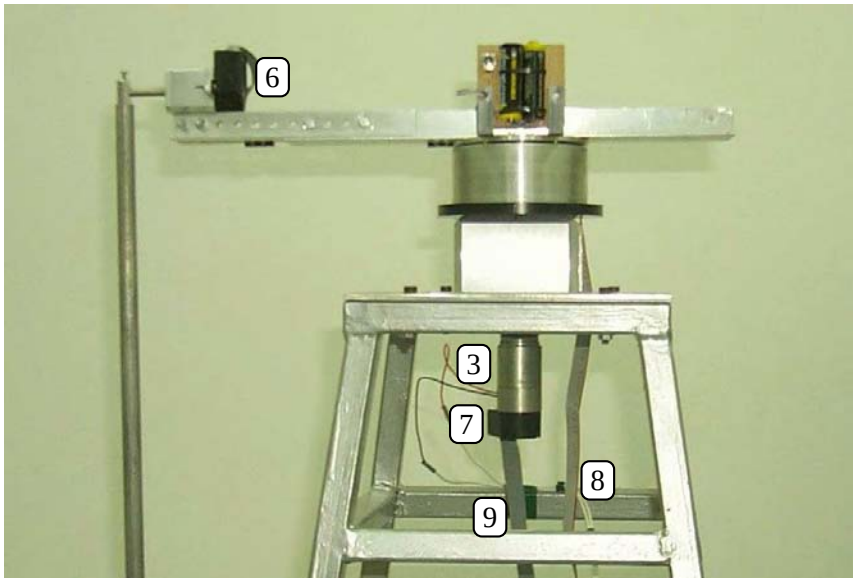
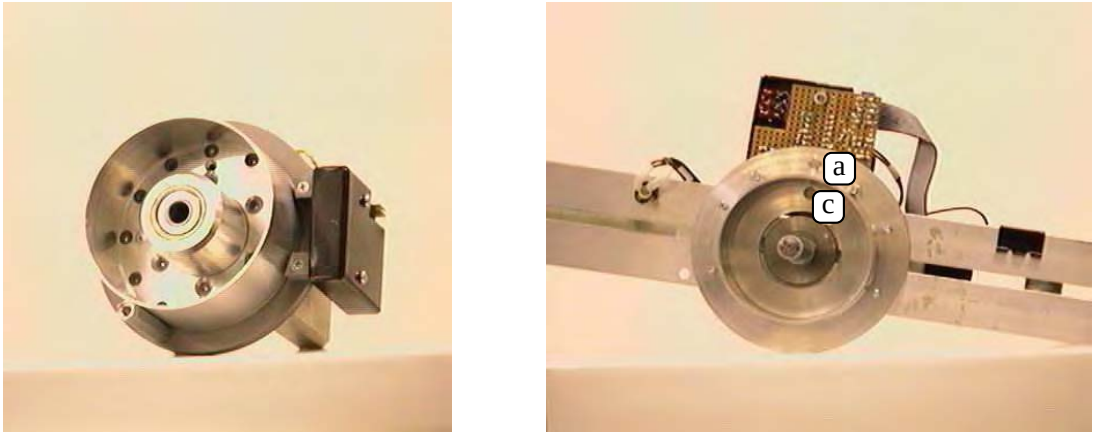


Figure A.2: The experiment system (zoomed in)



(a) Static part (b) Revolving part

Figure A.3: Disassembled upper system part

ensure free platform rotation, wireless signal transfer method is used. The encoder has two lines, therefore two signal transmitters—c and a—are used to transfer signal from revolving (Fig. A.3(a)) to static (Fig. A.3(a)) parts. Two rings of receivers installed in the static part ensure unremitting signal receiving. To avoid mixing of two signals, the inner room of the static part (see Fig. A.3(a)) is divided to two parts by thin wall (see Fig. A.3(b)).

The system is controlled by a DSP-based controller. The controller has two inputs—8 and 9—from encoders and one output goes to the amplifier 4 supplying power to DC motor. Matlab Simulink with additional DSP interface blockset is used to construct the control system block scheme and load it into the DSP-based controller. Another dSPACE program, "dSPACE ControlDesk," is used to monitor the system, register the data, and change some chosen parameters of control system during work.

A.2 Systems modeling

The modeling of the experiment, which includes both mechanical and electric parts, is divided into three stages:

1. the motion equations of the mechanical part are derived,
2. the equations of the electrical part are added to obtained motion equations, and
3. the resulting nonlinear equations of motion are linearized to obtain the plant model.

It is important to emphasize that a part of model parameters can be neither found analytically nor measured. For this reason, the parameters of the model in the "down" position (i.e., the linearization around the stable equilibrium point) is derived experimentally, via an identification procedure. This is possible because the pendulum dynamics are stable in this case. Having this model, equations for the linearized model in the "up" position are derived analytically, by changing the direction of the gravity force to its opposite value.

A.2.1 Equations of motion of the mechanical part

The mechanical part of experiment system depicted in Fig. A.1 can be sketched as shown in Fig. A.4. The

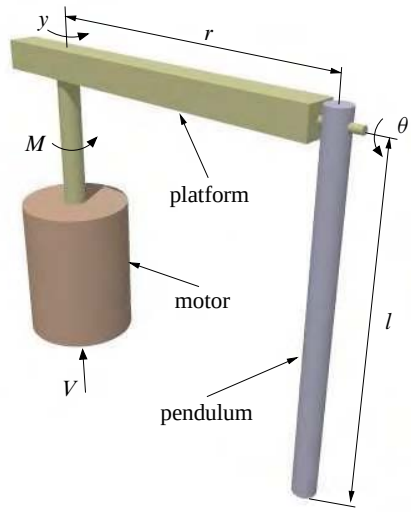


Figure A.4: Sketch of the experiment system

system input is the voltage (V) applied to the DC motor. The motor generates a torque (M). The system has two measured outputs: the first encoder measures the angle y of the revolving pendulum platform and the second encoder measures the angle θ of the pendulum. The system has two-degrees-of-freedom (the set of generalized coordinates is the angle y and θ).

Let us derive Lagrange's differential equations of motion (Meirovitch, 1975). The Lagrange's equation of motion for the jth degree of freedom, $j = 1, \ldots, n$, is defined as follows:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{q}_j}\right) - \frac{\partial L}{\partial q_j} + \frac{\partial \Phi}{\partial \dot{q}_j} = Q_j, \tag{A.1}$$

where the Lagrangian $L := T - V$, where T and V are the kinetic and potential energy, respectively, Φ is a dissipation function, Q is an applied force, and q is a generalized coordinate.

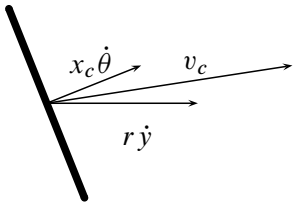


Figure A.5: v_c sketch

The pendulum platform, which is driven by the DC motor, is assumed to be viscously damped. Also, it can be assumed that the pendulum motion is viscously damped by a bearing friction. Thus, the dissipation function in our case is

$$\Phi = \frac{b\dot{y}^2}{2} + \frac{\zeta\dot{\theta}^2}{2},$$

where b and ζ are the damping coefficients of the platform and the pendulum, respectively. Hence,

$$\frac{\partial\Phi}{\partial\dot{y}} = b\dot{y} \quad \text{and} \quad \frac{\partial\Phi}{\partial\dot{\theta}} = \zeta\dot{\theta}.$$

The kinetic system energy is

$$T = \frac{1}{2}I_{\mathrm{r}}\dot{y}^2 + \frac{1}{2}I_{\mathrm{c}}\dot{\theta}^2 + \frac{1}{2}mv_{\mathrm{c}}^2,$$

where (see Fig. A.5)

$$v_{\mathrm{c}}^2 = (x_{\mathrm{c}}\dot{\theta})^2 + (r\dot{y})^2 + 2rx_{\mathrm{c}}\dot{\theta}\dot{y}\cos\theta,$$

I_{r} is the moment of inertia of the pendulum platform about its rotation axis, I_{c} is the moment of inertia of the pendulum about its centroid, v_{c} is the linear velocity of the pendulum centroid, m is the pendulum mass, x_{c} is the distance from the pendulum rotation axis to its centroid ($x_{\mathrm{c}} = 0.5l$), r is the distance from the pendulum rotation plane to the platform rotation axis, and l is the pendulum length. The potential system energy is

$$V = -mgx_{\mathrm{c}}\cos\theta.$$

The virtual work is $\delta W = M\delta y$, so that the resulting system Lagrangian is given by the following:

$$L = T - V = \frac{1}{2}I_{\mathrm{r}}\dot{y}^2 + \frac{1}{2}I_{\mathrm{c}}\dot{\theta}^2 + \frac{1}{2}m((x_{\mathrm{c}}\dot{\theta})^2 + (r\dot{y})^2 + 2rx_{\mathrm{c}}\dot{\theta}\dot{y}\cos\theta) + mgx_{\mathrm{c}}\cos\theta.$$

The derivatives appearing in the Lagrange's equations are

$$\begin{aligned}
\frac{\partial L}{\partial\dot{y}} &= I_{\mathrm{r}}\dot{y} + ml_1^2\dot{y} + mx_{\mathrm{c}}r\dot{\theta}\cos\theta \\
\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial\dot{y}}\right) &= (I_{\mathrm{r}} + mr^2)\ddot{y} + mx_{\mathrm{c}}r\ddot{\theta}\cos\theta - mx_{\mathrm{c}}r\dot{\theta}^2\sin\theta \\
\frac{\partial L}{\partial y} &= 0 \\
\frac{\partial L}{\partial\dot{\theta}} &= I_{\mathrm{c}}\dot{\theta} + mx_{\mathrm{c}}^2\dot{\theta} + mx_{\mathrm{c}}r\dot{y}\cos\theta \\
\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial\dot{\theta}}\right) &= I_{\mathrm{c}}\ddot{\theta} + mx_{\mathrm{c}}^2\ddot{\theta} + mx_{\mathrm{c}}r\ddot{y}\cos\theta - mx_{\mathrm{c}}r\dot{y}\dot{\theta}\sin\theta \\
\frac{\partial L}{\partial\theta} &= -mrx_{\mathrm{c}}\dot{y}\dot{\theta}\sin\theta - mgx_{\mathrm{c}}\sin\theta.
\end{aligned}$$

Substituting these expressions into (A.1) for $j = 1, 2$ we end up with the following equations of motion:

$$(I_{\mathrm{r}} + mr^2)\ddot{y} + mx_{\mathrm{c}}r\ddot{\theta}\cos\theta - mx_{\mathrm{c}}r\dot{\theta}^2\sin\theta + b\dot{y} = M \tag{A.2a}$$

$$(I_{\mathrm{c}} + mx_{\mathrm{c}}^2)\ddot{\theta} + mx_{\mathrm{c}}r\ddot{y}\cos\theta + mgx_{\mathrm{c}}\sin\theta + \zeta\dot{\theta} = 0, \tag{A.2b}$$

where the moment of inertia of the pendulum about its centroid is $I_{\mathrm{c}} = \frac{1}{3}ml^2$, the pendulum weight is $m = 0.2\,(\mathrm{kg})$, the pendulum length is $l = 0.665\,(\mathrm{m})$, the distance from platform rotational axis to the pendulum rotational plane is $r = 0.265\,(\mathrm{m})$, and $g \approx 9.81\,(\frac{\mathrm{m}}{\mathrm{s}^2})$ is the standard gravity.

A.2.2 Equations of motion of the electro-mechanical part

The next step is to derive the dynamic equation of the electro-mechanical part. Dynamics of DC motors are actually well studied (Dorf and Bishop, 2001). The torque M_{r} generated by a motor is proportional to the armature current $i(t)$,

$$M_{\mathrm{r}} = K_{\mathrm{m}}i,$$

where[1] $K_{\mathrm{m}} = 39.6\,(\frac{\mathrm{m\,Nm}}{\mathrm{A}})$ is the torque constant. The current satisfies the equation

$$L\dot{i} + Ri = V_{\mathrm{a}} - V_{\mathrm{b}},$$

where $R = 6.8\,(\Omega)$ is the armature resistance, $L = 6.2\cdot10^{-4}\,(\mathrm{H})$ is the armature inductance, V_{a} is an input voltage applied to the motor armature, and V_{b} is the back emf (electromotive force), which is proportional to the rotor angular velocity:

$$V_{\mathrm{b}} = K_{\mathrm{b}}\dot{\theta}_{\mathrm{r}},$$

where θ_{r} is the rotor angle and $K_{\mathrm{b}} = 4.15\,(\frac{\mathrm{mV}}{\mathrm{rpm}}) = 0.0396\,(\frac{\mathrm{V}}{\mathrm{rad/sec}})$ is the back emf constant. It is readily seen that the armature inductance is very small, so it will be neglected in the analysis.

The presences of a gear-box, which connects the motor rotor with the rotating platform, and an amplifier at the motor input are accounted for via the following scalings:

$$M = \eta K_{\mathrm{g}}M_{\mathrm{r}} = \eta K_{\mathrm{g}}K_{\mathrm{m}}i =: K_{\mathrm{eq}}i,$$

where $\eta = 0.79$ is the gear-box efficiency and $K_{\mathrm{g}} = 4.33$ is the gear-box reduction ratio (so that the equivalent torque constant of the system is $K_{\mathrm{eq}} = 135.46$),

$$y = \frac{1}{K_{\mathrm{g}}}\theta_{\mathrm{r}},$$

and

$$V_{\mathrm{a}} = K_{\mathrm{a}}u,$$

where $K_{\mathrm{a}} = 24$ is the amplifier gain and u is the normalized control input ($-1 \le u \le 1$).

Summarizing, the moment at the rotating platform, M, satisfies the following equation:

$$RM = K_{\mathrm{eq}}K_{\mathrm{a}}\,u - K_{\mathrm{g}}K_{\mathrm{eq}}K_{\mathrm{b}}\,\dot{y}. \tag{A.3}$$

Dynamical equations of the whole experiment are obtained by combining (A.2) and (A.3).

[1] All constants below are taken from the motor catalog.

A.2.3 State equations and linearization

The next step is to write state equations of the system. To this end, define the state vector

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} := \begin{bmatrix} y \\ \dot{y} \\ \theta \\ \dot{\theta} \end{bmatrix}. \tag{A.4}$$

With this notation equations (A.2) and (A.3) rewrite as follows:

$$\begin{gathered}(I_\mathrm{r} + mr^2)\dot{x}_2 + mx_\mathrm{c} r\dot{x}_4 \cos x_3 - mx_\mathrm{c} r x_4^2 \sin x_3 + bx_2 = M, \\ (I_\mathrm{c} + mx_\mathrm{c}^2)\dot{x}_4 + mx_\mathrm{c} r\dot{x}_2 \cos x_3 + mgx_\mathrm{c} \sin x_3 + \zeta x_4 = 0, \\ RM = K_\mathrm{eq} K_\mathrm{a} u - K_\mathrm{g} K_\mathrm{eq} K_\mathrm{b} x_2.\end{gathered}$$

By eliminating M, we obtain:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & I_\mathrm{r} + mr^2 & 0 & mx_\mathrm{c} r \cos x_3 \\ 0 & 0 & 1 & 0 \\ 0 & mx_\mathrm{c} r \cos x_3 & 0 & I_\mathrm{c} + mx_\mathrm{c}^2 \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ mx_\mathrm{c} r x_4^2 \sin x_3 - (b + K_\mathrm{g} K_\mathrm{eq} K_\mathrm{b}/R)x_2 \\ x_4 \\ -mgx_\mathrm{c} \sin x_3 - \zeta x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ K_\mathrm{eq} K_\mathrm{a}/R \\ 0 \\ 0 \end{bmatrix} u. \tag{A.5}$$

This is a nonlinear equation that needs to be linearized around its equilibrium points.

The equilibrium points are the solution of (A.5) corresponding to the condition $\dot{x} = 0$. It is readily seen that at equilibria x_1 is arbitrary, $x_2 = x_4 = u = 0$, and x_3 satisfies $\sin x_3 = 0$, which yields either $x_3 = 0$ (the crane problem) or $x_3 = \pi$ (the inverted pendulum problem). The linearized state equations at each of these two equilibrium points are then

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & R(I_\mathrm{r} + mr^2) & 0 & Rmx_\mathrm{c} r \\ 0 & 0 & 1 & 0 \\ 0 & mx_\mathrm{c} r & 0 & I_\mathrm{c} + mx_\mathrm{c}^2 \end{bmatrix} \dot{x}_\mathrm{down} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -Rb - K_\mathrm{g} K_\mathrm{eq} K_\mathrm{b} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\zeta \end{bmatrix} x_\mathrm{down} + \begin{bmatrix} 0 \\ K_\mathrm{eq} K_\mathrm{a} \\ 0 \\ 0 \end{bmatrix} u \tag{A.6a}$$

(for $\theta = x_3 = 0$) and

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & R(I_\mathrm{r} + mr^2) & 0 & -Rmx_\mathrm{c} r \\ 0 & 0 & 1 & 0 \\ 0 & -mx_\mathrm{c} r & 0 & I_\mathrm{c} + mx_\mathrm{c}^2 \end{bmatrix} \dot{x}_\mathrm{up} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -Rb - K_\mathrm{g} K_\mathrm{eq} K_\mathrm{b} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\zeta \end{bmatrix} x_\mathrm{up} + \begin{bmatrix} 0 \\ K_\mathrm{eq} K_\mathrm{a} \\ 0 \\ 0 \end{bmatrix} u \tag{A.6b}$$

(for $\theta = x_3 = \pi$), where

$$x_\mathrm{down} := \begin{bmatrix} x_1 - x_{\mathrm{equilib},1} \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad \text{and} \quad x_\mathrm{up} := \begin{bmatrix} x_1 - x_{\mathrm{equilib},1} \\ x_2 \\ x_3 - \pi \\ x_4 \end{bmatrix}$$

are the deviations of the state vector from the corresponding equilibrium point. In deriving these linearized equations the equality

$$\frac{\mathrm{d}}{\mathrm{d}x_i}(E^{-1}F) = E^{-1}\left(\frac{\mathrm{d}}{\mathrm{d}x_i}F - \frac{\mathrm{d}}{\mathrm{d}x_i}E \cdot E^{-1}F\right)$$

and the fact that

$$\left(F := \begin{bmatrix} x_2 \\ mx_c r x_4^2 \sin x_3 - (b + K_g K_{eq} K_b/R)x_2 \\ x_4 \\ -mgx_c \sin x_3 - \zeta x_4 \end{bmatrix}\right)\Bigg|_{x=x_{equilib}} = 0$$

for all $i = 1, 2, 3, 4$ were used to handle the descriptor form of the state equation in (A.5). The known parameters of (A.6) are summarized in Table A.1. There are still three parameters—I_r, b, and ζ—which

Parameter	x_c	r	m	g	I_c	R	K_{eq}	K_b	K_a	K_g
Value (in CI)	0.3325	0.265	0.2	9.81	0.0295	6.8	135.46	0.0396	24	4.33

Table A.1: Measurable parameters of the experimental setup

cannot be measured directly. We, thus, need to identify these parameters experimentally.

A.2.4 Identification of I_r and b

The first two of the unknown parameters above, i.e., the moment of inertia of the rotating platform I_r and its viscous friction constant b, are not connected with the pendulum. It is therefore possible to determine them from a simpler experiment when the pendulum is detached from the system (see Fig. A.6). The main

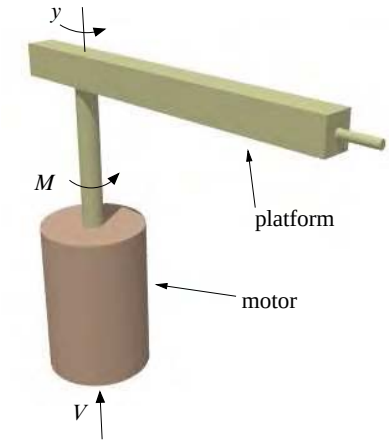


Figure A.6: Experiment for identifying I_r and b

advantage of this experiment is that the load is considerably simplified, so that the overall dynamics is easier to deal with. Indeed, the load now is a standard rotational body with unknown axial load I_r and damping b. The load dynamics are:

$$M = I_r\ddot{y} + b\dot{y}$$

Combining this equation with that of the DC motor, (A.3), the system can be presented by the block-

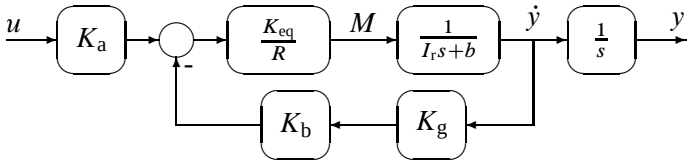


Figure A.7: Block-diagram of the DC motor with axial load

diagram in Fig. A.7 (the armature inductance L is neglected). The transfer function of this system can be easily derived. It is of the following form:

$$P_m(s) = \frac{K_{eq}K_a/(RI_r)}{s(s+(Rb+K_{eq}K_gK_b)/(RI_r))} =: \frac{\beta}{s(s+\alpha)}. \tag{A.7}$$

This is a second-order transfer function with two defining parameters, $\alpha > 0$ and $\beta > 0$. The parameters we are looking for, I_r and b, are uniquely determined by α and β. Indeed, it is readily seen that

$$I_r = \frac{K_{eq}K_a}{R\beta} \quad \text{and} \quad b = \frac{K_{eq}}{R}\left(\frac{\alpha}{\beta}K_a - K_bK_g\right). \tag{A.8}$$

We thus first identify α and β experimentally and then calculate the "physical" parameters by (A.8).

As the system described by (A.7) is unstable, an open-loop experiment is not easy to perform. A standard approach in this case is to stabilize the system and identify the closed-loop model first. The open-loop model is then easy to reconstruct. Although we do not know the parameters of (A.7), the system can obviously be stabilized by any proportional static feedback law. Thus, we first identify the parameters of

$$T(s) := \frac{P_m(s)k}{1+P_m(s)k} = \frac{\beta k}{s^2+\alpha s+\beta k},$$

where $k = 3.8$ is the controller gain. The parameters of $T(s)$ are identified from its response to the input $r(t) = 0.25 \cdot 1(t)$ by the Identification Toolbox of Matlab. Some fine-tuning of the resulting model is required to guarantee that the resulting transfer function has no zeros and that $T(0) = 1$, which, in turn, is required to guarantee that the resulting P_m has a pole at the origin. Fig. A.8, presenting the experimental

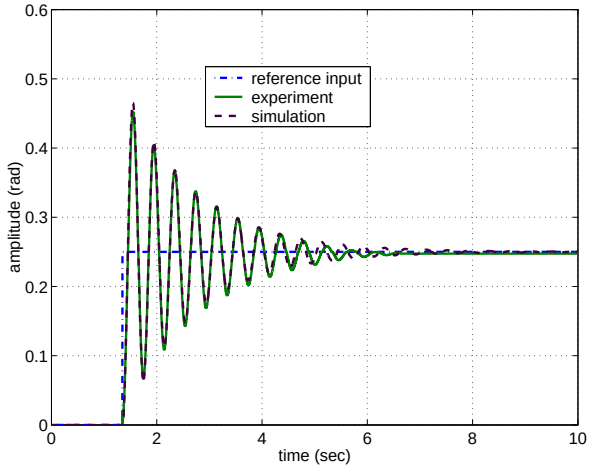


Figure A.8: Time response in the experiment for identifying I_r and b

and simulated responses of the system, demonstrates a good match between the model and the real motor. The identified DC motor transfer function is

$$P_{\mathrm{m}}(s) = \frac{57.8436}{s(s+1.4079)} = \frac{41.085}{s(0.71s+1)},$$

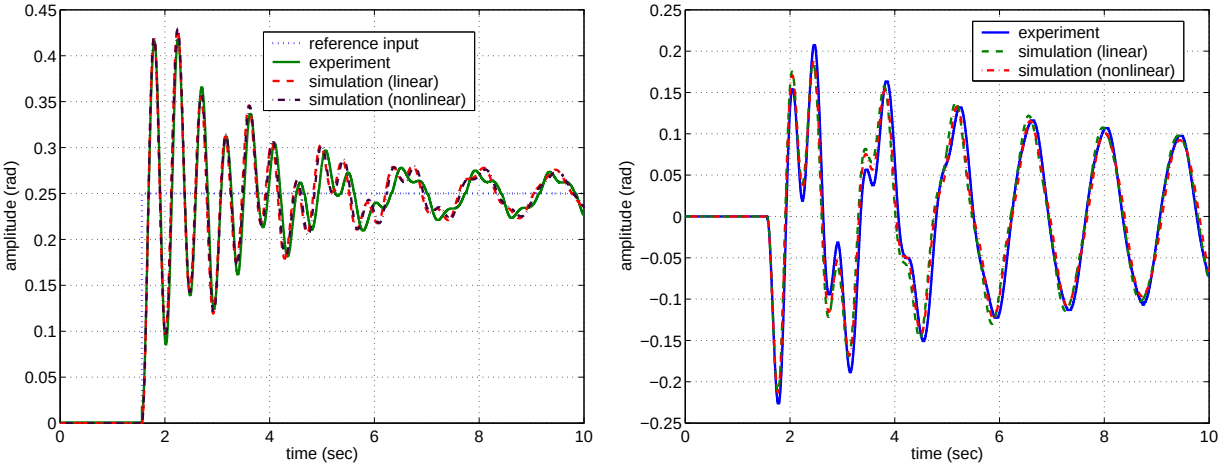
which is exactly the system in (3.1). Using formulae (A.8), the parameters I_{r} and b of the load are calculated, see Table A.2.

Parameter	I_{r}	b	ζ
Value (in CI)	0.0105	0.0104	0.003

Table A.2: Identified parameters of the experimental setup

A.2.5 Identification of ζ

The last parameter of (A.6) to be identified is the damping of the pendulum, ζ. This parameter is also identified experimentally, in the same setup as in §A.2.4 except that the pendulum is attached to the platform. Because there is only one parameter to be identified, it can be easily found by a linear search over the interval $(0, 1)$. The optimal ζ, in the least squares sense, is presented in Table A.2 and the resulting time responses are shown in Fig. A.9. One can see a reasonably good match between the experiment and



(a) Platform position (y)

(b) Pendulum position (θ)

Figure A.9: Time responses in the experiment for identifying ζ

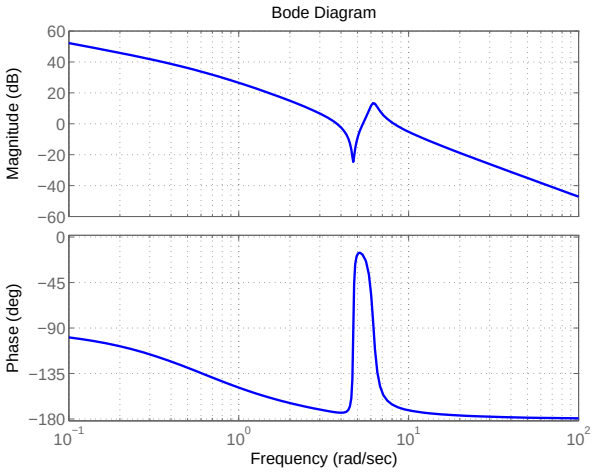
simulations (both linear and nonlinear, using model (A.5)).

A.2.6 Transfer functions of the experimental setup

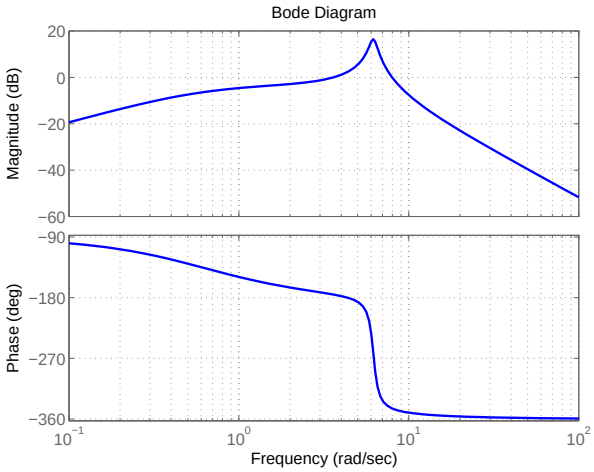
Thus, we have now all parameters of the linearized plants given by (A.6). This results in the following SIMO transfer functions:

$$P_{\mathrm{down}}(s) = \begin{bmatrix} 43.7195(s^2+0.1033s+22.3) \\ -25.8342s^2 \end{bmatrix} \frac{1}{s(s+0.6186)(s^2+0.6245s+38.35)}, \tag{A.9}$$

the Bode plot of which is depicted in Fig. A.10, and



(a) Bode diagram of $P_y(s)$

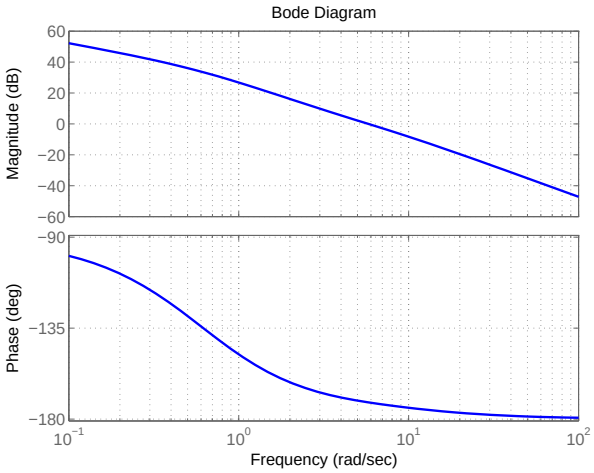


(b) Bode diagram of $P_\theta(s)$

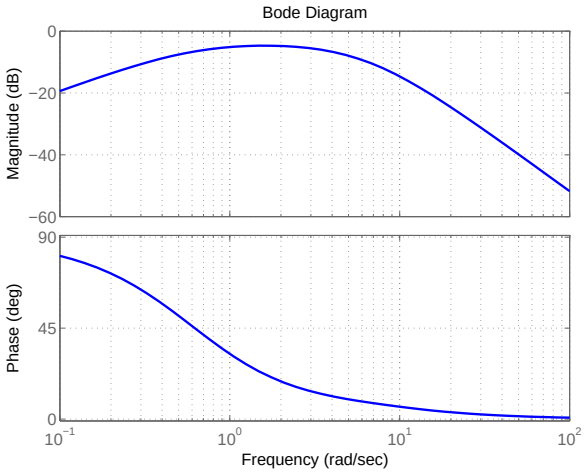
Figure A.10: Frequency response of the pendulum $P_{\text{down}}(s) = \begin{bmatrix} P_y(s) & P_\theta(s) \end{bmatrix}'$

$$P_{\text{up}}(s) = \begin{bmatrix} 43.7195(s+4.774)(s-4.67) \\ -25.8342s^2 \end{bmatrix} \frac{1}{s(s+0.6098)(s+6.562)(s-5.929)}, \tag{A.10}$$

the Bode plot of which is depicted in Fig. A.11.



(a) Bode diagram of $P_y(s)$



(b) Bode diagram of $P_\theta(s)$

Figure A.11: Frequency response of the inverted pendulum $P_{\text{up}}(s) = \begin{bmatrix} P_y(s) & P_\theta(s) \end{bmatrix}'$

State-space realizations of these transfer functions can also be found from (A.6):

$$P_{\text{down}} = \left[\begin{array}{cccc|c} 0 & 1 & 0 & 0 & 0 \\ 0 & -1.064 & 27.64 & 0.1281 & 43.72 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0.6288 & -38.63 & -0.179 & -25.83 \\ \hline 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{array}\right], \quad P_{\text{up}} = \left[\begin{array}{cccc|c} 0 & 1 & 0 & 0 & 0 \\ 0 & -1.064 & -27.64 & 0.1281 & 43.72 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0.6288 & 38.63 & -0.179 & -25.83 \\ \hline 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{array}\right].$$

Appendix B

Matlab implementation details

B.1 Simulink implementation of the resetting mechanism

In this appendix some details about the implementation of distributed-delay blocks using the reseting mechanism approach of Mondié *et al.* (2001b) are presented. This method is described in details in §3.2.2 and its central idea is to prevent unstable pole-zero cancellations in DTC blocks via a nonlinear resetting mechanism combined with a switch between two linear FIR systems.

Difficulties in implementing this scheme in Simulink stem from the fact that the only transfer function in Simulink that supports a resetting mechanism is the integrator block. Therefore, FIR blocks are built using integrator blocks as shown in §3.2.2, namely in the observer form in the SISO case (Fig. 3.8) and a more general state space form in the SIMO case (Fig. 3.9). All integrators are reset by an external signal generated by a pulse generator (resets occur when the pulse generator signal rises). Another element that needs a reset mechanism is the delay block. It has no reset options. This problem is circumvented by building a simple block, which provides a zero signal first h seconds after the reset and then switches it to the signal stored in the delay element. Moreover, since two FIR blocks work in an h seconds shift, only one delay element may be used. One FIR block is fed with the signal from the delay element, while the second—with the zero signal. After h seconds they are switched. The block scheme of this system

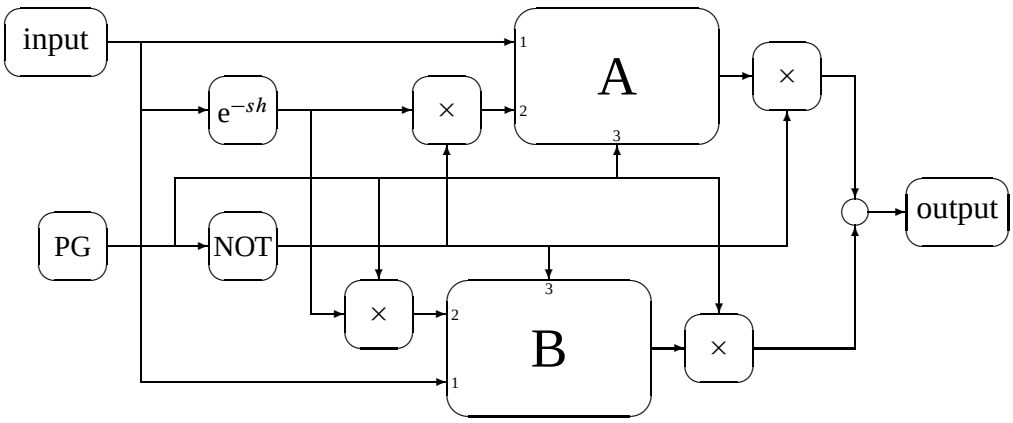


Figure B.1: Block scheme for implementing resetting FIR blocks in Simulink

is shown in Fig. B.1. One can see two blocks, A and B, which are identical FIR blocks without delays. Each of them has 3 ports. The first one is used to supply the input signal. The second port receives the signal from the delay element, which is periodically switched to zero. The third input is fed by the signal

from the pulse generator. The pulse generator (the PG block) generates a signal of the form depicted in Fig. B.2. It switches between 0 to 1 and every h seconds. The "NOT" block is used to obtain the binary

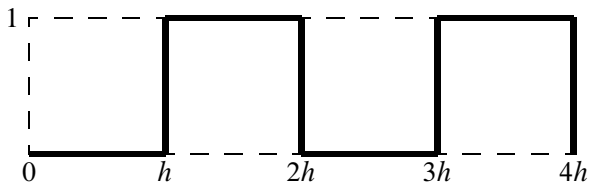


Figure B.2: Signal from the pulse generator

inverse of this pulse signal. The product blocks have two input signals—one is from the delay element (or FIR outputs) and another one is from the pulse generator (or NOT block)—then their outputs is either zero signal or the signal itself. Thus, the two pairs of the product blocks work as two switches.

Consider one cycle is details. During first h seconds, the output of the PG is zero. This is the "signal generation" period for the block A and the "history accumulation" period for the block B. During this interval, the PD generates 0 (see Fig. B.2) and then the block NOT generates 1. Both product blocks linked with the block A receive one of their inputs from the block NOT (i.e., 1), so that the block A receives a signal from the delay block and its output reaches the output block. Product blocks linked with the block B receives 0 from the pulse generator, so that the block B receives 0 from the delay block and is disconnected from the output block. At the time moment h, the signal from the pulse generator becomes 1 and the output of the block NOT vanishes. All integrators in the block A are reset and A starts to accumulate the history while the block B starts to generate the control signal. Indeed, the product blocks linked with the block A receive 0, so that the block A receives 0 from the delay block and also disconnects from the output block. At the time moment $2h$, the signal from the pulse generator becomes 0 again, and the next cycle begins.

B.2 Auxiliary Matlab functions

B.2.1 FIR block implementation by the RM method for SISO systems

```
function [numD0,denD0,numDh,denDh] = AFBPswitchSISO(Dh,D0,h,bdn)
 % Function [numD0,denD0,numDh,denDh] = AFBPswitchSISO(Dh,D0,h,bdn)
 % This fuction is used to build realization of FIR block using Resseting Model
 % Process method.
 % FIR block --> Dh-D0*exp(-hs).
 % bdn --> destination block diagram name. For example, 'MyBlockDiagram'
 vv = version;
 [numD0,denD0]=tfdata(D0,'v'); [numDh,denDh]=tfdata(Dh,'v');
 N=length(denD0); Name1='BlockPai'; Name2='Dh'; Name3='D0';
 Name4='PlantModel'; sw1='rising'; sw2='falling'; be1='R'; be2='F';
 if vv(1)=='6' & vv(3)=='1'
     pt1='simulink3';
     pt2='simulink3/Math';
     pt3='simulink3/Nonlinear';
     pt4='simulink3/Math';
 else
     pt1='simulink';
     pt2='simulink/Math Operations';
     pt3='simulink/Signal Routing';
     pt4='simulink/Logic and Bit Operations';
 end
```

```
lstr=[bdn,'/',num2str(Name1)]; add_block('built-in/SubSystem',lstr);
for j=1:2
  switch j
   case 1
    sw=sw1; be=be1; ope=1;
   otherwise
    sw=sw2; be=be2; ope=0;
   end
  Lstr=[bdn,'/',num2str(Name1),'/',num2str(Name4),num2str(be)];
  add_block('built-in/SubSystem',Lstr);
  add_block([pt1,'/Sources/In1'],[Lstr,'/inu']);
  add_block([pt1,'/Sources/In1'],[Lstr,'/inr']);
  add_block([pt1,'/Sinks/Out1'],[Lstr,'/out']);
  add_block([pt1,'/Continuous/Integrator'],[Lstr,'/Intn'],'ExternalReset',sw);
  add_block('built-in/Gain',[Lstr,'/b0'],'Gain',['num',Name3,'(',num2str(N),')']);
  add_block('built-in/Gain',[Lstr,'/bt0'],'Gain',['num',Name2,'(',num2str(N),')']);
  add_block('built-in/Gain',[Lstr,'/a0'],'Gain',['den',Name3,'(',num2str(N),')']);
  add_block([pt2,'/Sum'],[Lstr,'/sum0'],'Inputs','--+');
  add_block([pt1,'/Continuous/Transport Delay'],[Lstr,'/del',be],'DelayTime','h');
  if ope==1
   add_block([pt2,'/Product'],[Lstr,'/Product']);
   add_line(Lstr,'inr/1','Product/2');
   add_line(Lstr,'Product/1',['del',be,'/1']);
   add_line(Lstr,'inu/1','Product/1');
  else
   add_block([pt2,'/Product'],[Lstr,'/Product']);
   add_block([pt4,'/Logical Operator'],[Lstr,'/NOT'],'Operator','NOT');
   add_line(Lstr,'inr/1','NOT/1');
   add_line(Lstr,'NOT/1','Product/2');
   add_line(Lstr,'Product/1',['del',be,'/1']);
   add_line(Lstr,'inu/1','Product/1');
  end
  add_line(Lstr,'inu/1','bt0/1');
  add_line(Lstr,['del',be,'/1'],'b0/1');
  add_line(Lstr,'Intn/1','out/1');
  add_line(Lstr,'Intn/1','a0/1');
  add_line(Lstr,'inr/1','Intn/2');
  add_line(Lstr,'a0/1','sum0/1');
  add_line(Lstr,'b0/1','sum0/2');
  add_line(Lstr,'bt0/1','sum0/3');
  for i=2:N-1
    add_block([pt1,'/Continuous/Integrator'],[Lstr,'/Int',num2str(i-1)],...
     'ExternalReset',sw);
    add_block('built-in/Gain',[Lstr,'/b',num2str(i-1)],...
     'Gain',['num',Name3,'(',num2str(N-i+1),')']);
    add_block('built-in/Gain',[Lstr,'/bt',num2str(i-1)],...
     'Gain',['num',Name2,'(',num2str(N-i+1),')']);
    add_block('built-in/Gain',[Lstr,'/a',num2str(i-1)],...
     'Gain',['den',Name3,'(',num2str(N-i+1),')']);
    add_block([pt2,'/Sum'],[Lstr,'/sum',num2str(i-1)],'Inputs','--++');
    add_line(Lstr,'inr/1',['Int',num2str(i-1),'/2']);
    add_line(Lstr,['sum',num2str(i-2),'/1'],['Int',num2str(i-1),'/1']);
    add_line(Lstr,'inu/1',['bt',num2str(i-1),'/1']);
    add_line(Lstr,['del',be,'/1'],['b',num2str(i-1),'/1']);
    add_line(Lstr,'Intn/1',['a',num2str(i-1),'/1']);
    add_line(Lstr,['a',num2str(i-1),'/1'],['sum',num2str(i-1),'/1']);
    add_line(Lstr,['b',num2str(i-1),'/1'],['sum',num2str(i-1),'/2']);
    add_line(Lstr,['bt',num2str(i-1),'/1'],['sum',num2str(i-1),'/3']);
    add_line(Lstr,['Int',num2str(i-1),'/1'],['sum',num2str(i-1),'/4']);
```

```
   end
    add_line(Lstr,['sum',num2str(N-2),'/1'],'Intn/1');
  end
 add_block([pt1,'/Sources/Pulse
 Generator'],[lstr,'/PGD'],'Period','2*h','PhaseDelay','h');
 add_block([pt3,'/Switch'],[lstr,'/Switch'],'Threshold','0.5');
 add_block([pt1,'/Sources/In1'],[lstr,'/in']);
 add_block([pt1,'/Sinks/Out1'],[lstr,'/out']);
 add_block([pt4,'/Logical Operator'],[lstr,'/NOT'],'Operator','NOT');
 add_line(lstr,'in/1','PlantModelR/1');
 add_line(lstr,'in/1','PlantModelF/1');
 add_line(lstr,'PlantModelR/1','Switch/1');
 add_line(lstr,'PlantModelF/1','Switch/3');
 add_line(lstr,'PGD/1','PlantModelR/2');
 add_line(lstr,'PGD/1','PlantModelF/2');
 add_line(lstr,'PGD/1','NOT/1'); add_line(lstr,'NOT/1','Switch/2');
 add_line(lstr,'Switch/1','out/1')
```

B.2.2 FIR block implementation by the RM method for SIMO systems

```
function [Btz] = AFBPswitchSIMO(D0,h,bdn)
 % Function [numD0,denD0,numDh,denDh] = AFBPswitchSIMO(D0,h,bdn)
 % This fuction is used to build realization of FIR block using Resseting Model
 % Process method.
 % FIR block --> Dh-D0*exp(-hs).
 % bdn --> destination block diagram name. For example, 'MyBlockDiagram'
 vv = version;
 Btz=expm(-D0.a*h)*D0.b; N = size(D0.a,1);
 Name1='BlockPai'; Name2='Dh'; Name3='D0'; Name4='IntBlock';
 sw1='rising'; sw2='falling'; be1='R'; be2='F';
 if vv(1)=='6' & vv(3)=='1'
     pt1='simulink3';
     pt2='simulink3/Math';
     pt3='simulink3/Nonlinear';
     pt4='simulink3/Math';
     pt5='simulink3/Signals & Systems';
 else
     pt1='simulink';
     pt2='simulink/Math Operations';
     pt3='simulink/Signal Routing';
     pt4='simulink/Logic and Bit Operations';
 end
 lstr=[bdn,'/',num2str(Name1)]; add_block('built-in/SubSystem',lstr);
 add_block([pt1,'/Sources/In1'],[lstr,'/in']);
 add_block([pt1,'/Sinks/Out1'],[lstr,'/out']);
 add_block([pt1,'/Sources/Pulse
 Generator'],[lstr,'/PGD'],'Period','2*h','PhaseDelay','h');
 add_block([pt3,'/Switch'],[lstr,'/Switch'],'Threshold','0.5');
 add_block([pt4,'/Logical Operator'],[lstr,'/NOT'],'Operator','NOT');
 for j=1:2
   switch j
    case 1
     sw=sw1; be=be1;  ope=1;
    otherwise
     sw=sw2; be=be2;  ope=0;
    end
    Lstr=[bdn,'/',num2str(Name1),'/',num2str(Name4),num2str(be)];
    add_block('built-in/SubSystem',Lstr);
    add_block([pt1,'/Sources/In1'],[Lstr,'/inu']);
```

```
    add_block([pt1,'/Sources/In1'],[Lstr,'/inr']);
    add_block([pt1,'/Sinks/Out1'],[Lstr,'/out']);
    add_block([pt5,'/Mux'],[Lstr,'/Mux'],'Inputs',num2str(N));
    add_block([pt5,'/Demux'],[Lstr,'/Demux'],'Outputs',num2str(N));
    add_line(Lstr,'inu/1','Demux/1');
    add_line(Lstr,'Mux/1','out/1');
   for i=1:N
       add_block([pt1,'/Continuous/Integrator'],[Lstr,'/Int',num2str(i)],...
          'ExternalReset',sw);
       add_line(Lstr,['Demux/',num2str(i)],['Int',num2str(i),'/1']);
       add_line(Lstr,['Int',num2str(i),'/1'],['Mux/',num2str(i)]);
       add_line(Lstr,'inr/1',['Int',num2str(i),'/2']);
   end
   add_block([pt1,'/Continuous/Transport Delay'],[lstr,'/del',be],'DelayTime','h');
   add_block([pt2,'/Sum'],[lstr,'/sum',be],'Inputs','-++');
   add_block([pt2,'/Matrix Gain'],[lstr,'/A',be],'Gain',[Name3,'.a']);
   add_block([pt2,'/Matrix Gain'],[lstr,'/C',be],'Gain',[Name3,'.c']);
   add_block([pt2,'/Matrix Gain'],[lstr,'/B',be],'Gain',[Name3,'.b']);
   add_block([pt2,'/Matrix Gain'],[lstr,'/Bt',be],'Gain','Btz');
   add_block([pt2,'/Product'],[lstr,'/Product',be]);
   add_line(lstr,['Product',be,'/1'],['del',be,'/1']);
   add_line(lstr,'in/1',['Product',be,'/1']);
   if ope==1
    add_line(lstr,'PGD/1',['Product',be,'/2']);
   else
    add_line(lstr,'NOT/1',['Product',be,'/2']);
   end
   add_line(lstr,['del',be,'/1'],['B',be,'/1']);
   add_line(lstr,['B',be,'/1'],['sum',be,'/1']);
   add_line(lstr,['Bt',be,'/1'],['sum',be,'/2']);
   add_line(lstr,['A',be,'/1'],['sum',be,'/3']);
   add_line(lstr,['sum',be,'/1'],[num2str(Name4),num2str(be),'/1']);
   add_line(lstr,[num2str(Name4),num2str(be),'/1'],['C',be,'/1']);
   add_line(lstr,[num2str(Name4),num2str(be),'/1'],['A',be,'/1']);
  end
 add_line(lstr,'in/1','BtR/1'); add_line(lstr,'in/1','BtF/1');
 add_line(lstr,'CR/1','Switch/1'); add_line(lstr,'CF/1','Switch/3');
 add_line(lstr,'PGD/1','IntBlockR/2');
 add_line(lstr,'PGD/1','IntBlockF/2');
 add_line(lstr,'PGD/1','NOT/1'); add_line(lstr,'NOT/1','Switch/2');
 add_line(lstr,'Switch/1','out/1')
```

B.2.3 FIR block implementation by the LDA method

```
function [Kvec,hvec] = AFBPsna(D0,h,tau,n,bdn)
 % Function [Kvec,hvec] = AFBPsna(D0,h,tau,n,bdn)
 % This fuction is used to build realization of FIR block using Stable
 % Numerical Approximation.
 % FIR block --> Dh-D0*exp(-hs).
 % h/n --> descritisation step
 % bdn --> destination block diagram name. For example, 'MyBlockDiagram'
 D0=ss(D0); A=D0.a; B=D0.b; C=D0.c; m=size(C,1);
 Cn=C*(eye(length(A))+tau*A); Kvec(1:m,1)=-tau*C*B+h/n/2*Cn*B;
 Kvec(1:m,n+1)=tau*C*expm(-A*h)*B+h/n/2*Cn*expm(-A*h)*B;
 for i=1:n-1;
  Kvec(1:m,i+1)=h/n*Cn*expm(-A*h*i/n)*B;
 end
 vv=version; if vv(1)=='6' & vv(3)=='1'
  pt1='simulink3';
```

```
 pt2='simulink3/Math';
else
 pt1='simulink';
 pt2='simulink/Math Operations';
end
Name1='BlockPai'; str1=[bdn,'/',Name1];
add_block('built-in/SubSystem',str1);
add_block([pt1,'/Sources/In1'],[str1,'/in']);
add_block([pt1,'/Continuous/Transport
Delay'],[str1,'/del0'],'DelayTime','h/n');
add_block('built-in/Gain', [str1,'/Gain0'], 'Gain',
['Kvec(:,',num2str(n+1),')']);
add_block([pt2,'/Sum'],[str1,'/sum0']);
add_block([pt1,'/Continuous/Transfer
Fcn'],[str1,'/LPF'],'Numerator','[1]','Denominator','[tau 1]');
add_line(str1,'in/1','LPF/1'); add_line(str1,'LPF/1','del0/1');
add_line(str1,'LPF/1','Gain0/1'); add_line(str1,'Gain0/1','sum0/1');
for i=1:n-1;
 add_block([pt1,'/Continuous/Transport Delay'],...
  [str1,'/del',num2str(i)],'DelayTime',['h/n']);
 add_block('built-in/Gain', [str1,'/Gain',num2str(i)],...
  'Gain', ['Kvec(:,',num2str(n-i+1),')']);
 add_block([pt2,'/Sum'],[str1,'/sum',num2str(i)]);
 add_line(str1,['del',num2str(i-1),'/1'],['del',num2str(i),'/1']);
 add_line(str1,['del',num2str(i-1),'/1'],['Gain',num2str(i),'/1']);
 add_line(str1,['Gain',num2str(i),'/1'],['sum',num2str(i-1),'/2']);
 add_line(str1,['sum',num2str(i-1),'/1'],['sum',num2str(i),'/1']);
end add_block('built-in/Gain', [str1,'/Gain',num2str(n)], 'Gain',
['Kvec(:,1)']); add_block([pt1,'/Sinks/Out1'],[str1,'/out']);
add_line(str1,['del',num2str(n-1),'/1'],['Gain',num2str(n),'/1']);
add_line(str1,['Gain',num2str(n),'/1'],['sum',num2str(n-1),'/2']);
add_line(str1,['sum',num2str(n-1),'/1'],'out/1')
```

References

Adam, E., H. Latchman, and O. Crisalle (2000). "Robustness of the Smith predictor with respect to uncertainty in the time-delay parameter," in *Proc. American Control Conf.*, Chicago, IL, pp. 1452–1457.

Åström, K. J. (1977). "Frequency domain properties of Otto Smith regulators," *Int. J. Control*, **26**, no. 2, pp. 307–314.

Åström, K. J. and T. Hagglund (1995). *PID Controllers: Theory, Design, and Tuning*, Instrument Society of America, Research Triangle Park, NC, 2nd edn.

Åström, K. J., C. C. Hang, and B. C. Lim (1994). "A new Smith predictor for controlling a process with an integrator and long dead-time," *IEEE Trans. Automat. Control*, **39**, pp. 343–345.

Barman, J. F., F. M. Callier, and C. A. Desoer (1973). "L^2-stability and L^2-instability of linear time-invariant distributed feedback systems perturbed by a small delay in the loop," *IEEE Trans. Automat. Control*, **18**, no. 5, pp. 479–484.

Başar, T. and P. Bernhard (1991). *H^∞-Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach*, Birkhäuser, Boston.

Bryson, A. E., Jr. (1999). *Dynamic Optimization*, Addison-Wesley, Menlo Park, CA.

Cooke, K. L. and Z. Grossman (1982). "Discrete delay, distributed delays and stability swithes," *J. Math. Anal. Appl.*, **86**, pp. 592–627.

Curtain, R. F., G. Weiss, and M. Weiss (1996). "Coprime factorization for regular linear systems," *Automatica*, **32**, no. 11, pp. 1519–1531.

Dorf, R. C. and R. H. Bishop (2001). *Modern Control System*, Prentice-Hall, 9th edn.

Dym, H., T. Georgiou, and M. C. Smith (1995). "Explicit formulas for optimally robust controllers for delay systems," *IEEE Trans. Automat. Control*, **40**, no. 4, pp. 656–669.

Foias, C., H. Özbay, and A. Tannenbaum (1996). *Robust Control of Infinite Dimensional Systems: Frequency Domain Methods*, vol. 209 of *Lecture Notes in Control and Information Sciences*, Springer-Verlag, London.

Franklin, G. F., J. D. Powell, and A. Emami-Naeini (2002). *Feedback Control of Dynamic Systems*, Prentice-Hall, Upper Saddle River, NJ, 4th edn.

Furukawa, T. and E. Shimemura (1983). "Predictive control for systems with time delay," *Int. J. Control*, **37**, no. 2, pp. 399–412.

Furutani, E. and M. Araki (1998). "Robust stability of state-predictive and Smith control systems for plants with a pure delay," *Int. J. Robust and Nonlinear Control*, **8**, pp. 907–919.

Georgiou, T. T. and M. C. Smith (1992). "Robust stabilization in the gap metric: Controller design for distributed plants," *IEEE Trans. Automat. Control*, **37**, no. 8, pp. 1133–1143.

Georgiou, T. T. and M. C. Smith (1993). "Graphs, causality and stabilizability: linear, shift-invariant systems on $\mathcal{L}_2[0,\infty)$," *Math. Control, Signals and Systems*, **6**, pp. 195–223.

Golub, G. H. and C. F. Van Loan (1996). *Matrix Computations*, The Johns Hopkins University Press, Baltimore, 3rd edn.

Gu, K., V. L. Kharitonov, and J. Chen (2003). *Stability of Time-Delay Systems*, Birkhäuser, Boston.

Hespanha, J. P., P. Naghshtabrizi, and Y. Xu (2007). "A survey of recent results in networked control systems," *Proc. IEEE*, **95**, no. 1, pp. 138–162.

Horowitz, I. M. (1983). "Some properties of delayed controls (Smith regulators)," *Int. J. Control*, **38**, no. 5, pp. 977–990.

Kharitonov, V. L., S.-I. Niculescu, J. Moreno, and W. Michiels (2005). "Static output feedback stabilization: Necessary conditions for multiple delay controllers," *IEEE Trans. Automat. Control*, **50**, no. 1, pp. 82–86.

Kidron, O. and O. Yaniv (1995). "Robust control of uncertain resonant systems," *European J. Control*, **1**, no. 2, pp. 104–112.

Kleinman, D. L. (1969). "Optimal control of linear systems with time-delay and observation noise," *IEEE Trans. Automat. Control*, **14**, no. 5, pp. 524–527.

Lee, B.-K., H.-W. Chen, and B.-S. Chen (2004). "Power control of cellular radio systems via robust Smith prediction filter," *IEEE Trans. on Wireless Communication*, **3**, no. 5, pp. 1822–1832.

Lewis, R. (1979). "Control–delayed system properies via an ordinary model," *Int. J. Control*, **30**, no. 3, pp. 477–490.

Lupfer, D. E. and M. W. Oglesby (1962). "The application of dead-time compensation to a chemical reactor for automatic control of production rate," *ISA Trans.*, **1**, no. 1, pp. 72–80.

Manitius, A. Z. and A. W. Olbrot (1979). "Finite spectrum assignment problem for systems with delay," *IEEE Trans. Automat. Control*, **24**, pp. 541–553.

Marshall, J. E., H. Górecki, A. Korytowski, and K. Walton (1992). *Time-Delay Systems: Stability and Performance Criteria with Applications*, Ellis Horwood, London.

Mascolo, S. (2000). "Smith's principle for congestion control in high-speed data networks," *IEEE Trans. Automat. Control*, **45**, no. 2, pp. 358–364.

Mataušek, M. R. and A. D. Micić (1999). "On the modified Smith predictor for controlling a process with an integrator and long dead-time," *IEEE Trans. Automat. Control*, **44**, pp. 1603–1606.

McFarlane, D. and K. Glover (1990). *Robust Controller Design Using Normalized Coprime Factor Plant Descriptions*, vol. 138 of *Lecture Notes in Control and Information Sciences*, Springer-Verlag, Berlin.

Meinsma, G., L. Mirkin, and Q.-C. Zhong (2002). "Control of systems with I/O delay via reduction to a one-block problem," *IEEE Trans. Automat. Control*, **47**, no. 11, pp. 1890–1895.

Meinsma, G. and H. Zwart (2000). "On H^∞ control for dead-time systems," *IEEE Trans. Automat. Control*, **45**, no. 2, pp. 272–285.

Meirovitch, L. (1975). *Elements of vibration analisys*, McGraw-Hill.

Miall, R. C., D. J. Weir, D. M. Wolpert, and J. F. Stein (1993). "Is the cerebellum a Smith predictor?" *J. Motor Behavior*, **25**, no. 3, pp. 203–216.

Michiels, W., K. Engelborghs, P. Vansevenant, and D. Roose (2002). "Continuous pole placement for delay equations," *Automatica*, **38**, no. 5, pp. 747–761.

Michiels, W. and S.-I. Niculescu (2003). "On the delay sensitivity of Smith predictors," *Int. J. Systems Sciences*, **34**, no. 8–9, pp. 543–552.

Michiels, W., S.-I. Niculescu, and L. Moreau (2003). "Improving the static output feedback stabilizability of linear systems by introducing delays and time-varying gains: A comparison," in *Proc. 4th IFAC Workshop on Time Delay Systems*, INRIA, Rocquencourt, France.

Mirkin, L. (2003). "On the extraction of dead-time controllers and estimators from delay-free parametrizations," *IEEE Trans. Automat. Control*, **48**, no. 4, pp. 543–553.

Mirkin, L. (2004). "On the approximation of distributed-delay control laws," *Syst. Control Lett.*, **55**, no. 5, pp. 331–342.

Mirkin, L. (2006). "On the dead-time compensation from L^1 perspectives," *IEEE Trans. Automat. Control*, **51**, no. 6, pp. 1069–1073.

Mirkin, L. and Z. J. Palmor (2005). "Control issues in systems with loop delays," in *Handbook of Networked and Embedded Control Systems* (D. Hristu-Varsakelis and W. S. Levine, eds.), Birkhäuser, pp. 627–648.

Mirkin, L. and N. Raskin (2003). "Every stabilizing dead-time controller has an observer-predictor-based structure," *Automatica*, **39**, no. 10, pp. 1747–1754.

Mirkin, L. and G. Tadmor (2002). "H^∞ control of system with I/O delay: A review of some problem-oriented methods," *IMA J. Math. Control & Information*, **19**, pp. 185–199.

Mirkin, L. and Q.-C. Zhong (2003). "2DOF controller parametrization for systems with a single I/O delay," *IEEE Trans. Automat. Control*, **48**, no. 11, pp. 1999–2004.

Mondié, S., M. Dambrine, and O. Santos (2001a). "Approximation of control laws with distributed delays: A necessary condition for stability," in *Proc. SSSC'01*, Prague, Czech Republic.

Mondié, S., R. Lozano, and J. Collado (2001b). "Resetting process-model control for unstable systems with delay," in *Proc. 40th IEEE Conf. Decision and Control*, Orlando, FL, pp. 2247–2252.

Mondié, S. and W. Michiels (2003). "A safe implementation for finite spectrum assignment of unstable input-delayed systems," in *Proc. 42nd IEEE Conf. Decision and Control*, Maui, HW. (to appear).

Morari, M. and E. Zafiriou (1989). *Robust Process Control*, Prentice-Hall, Englewood Cliffs, NJ.

Nordin, M. and P.-O. Gutman (1995). "Digital QFT design for the benchmark problem," *European J. Control*, **1**, no. 2, pp. 97–103.

Normey-Rico, J. E. and E. F. Camacho (2002). "A unified approach to design dead-time compensators for stable and integrative processes with dead-time," *IEEE Trans. Automat. Control*, **47**, no. 2, pp. 299–305.

Palmor, Z. J. (1980). "Stability properties of Smith dead-time compensator controller," *Int. J. Control*, **32**, pp. 937–949.

Palmor, Z. J. (1996). "Time-delay compensation — Smith predictor and its modifications," in *The Control Handbook* (W. S. Levine, ed.), CRC Press, Boca Raton, pp. 224–237.

Palmor, Z. J. and M. Blau (1994). "An auto-tuner for Smith dead time compensator," *Int. J. Control*, **60**, no. 1, pp. 117–135.

Palmor, Z. J. and D. W. Powers (1985). "Improved dead-time compensator controllers," *AIChE Journal*, **31**, no. 2, pp. 215–221.

Palmor, Z. J. and R. Shinnar (1981). "Design of advanced process controllers," *AIChE J.*, **27**, no. 5, pp. 793–805.

Raskin, N. (2000). *Control of Flexible Structures Described by the Wave Equations*, Ph.D. thesis, Faculty of Mechanical Eng., Technion– IIT.

Reboldi, G. P., P. D. Home, G. Calabrese, P. G. Fabietti, P. Brunetti, and M. Massi Benedetti (1991). "Time delay compensation for closed-loop insulin delivery systems: a simulation study," *Int. J. Artif. Organs*, **14**, no. 6, pp. 350–358.

Richard, J.-P. (2003). "Time-delay systems: an overview of some recent advances and open problems," *Automatica*, **39**, no. 10, pp. 1667–1694.

Sidi, M. (1997). "Gain-bandwidth limitations of feedback systems with non-minimum phase plants," *Int. J. Control*, **67**, no. 5, pp. 731–745.

Skogestad, S. and I. Postlethwaite (1996). *Multivariable Feedback Control: Analysis and Design*, John Wiley & Sons, Chichester.

Smith, O. J. M. (1957). "Closer control of loops with dead time," *Chem. Eng. Progress*, **53**, no. 5, pp. 217–219.

Smith, O. J. M. (1959). "A controller to overcome dead time," *ISA J.*, **6**, no. 2, pp. 23–33.

Su, W., L. Qiu, and J. Chen (2003). "Fundamental performance limitations in tracking sinusoidal signals," *IEEE Trans. Automat. Control*, **48**, no. 8, pp. 1371–1380.

Tadmor, G. (1997a). "Robust control in the gap: A state space solution in the presence of a single input delay," *IEEE Trans. Automat. Control*, **42**, no. 9, pp. 1330–1335.

Tadmor, G. (1997b). "Weighted sensitivity minimization in systems with a single input delay: A state space solution," *SIAM J. Control Optim.*, **35**, no. 5, pp. 1445–1469.

Tadmor, G. (2000). "The standard H^∞ problem in systems with a single input delay," *IEEE Trans. Automat. Control*, **45**, no. 3, pp. 382–397.

van Assche, V., M. Dambrine, J.-F. Lafay, and J.-P. Richard (1999). "Some problems arising in the implementation of distributed-delay control laws," in *Proc. 38th IEEE Conf. Decision and Control*, Phoenix, AZ, pp. 4668–4672.

Veronesi, M. (2003). “Performance improvement of Smith predictor through automatic computation of dead time,” Tech. Rep. 35, Yokogava Technical Report English Ed.

Wang, Z. Q., P. Lundström, and S. Skogestad (1994). “Representation of uncertain time delays in the H^∞ framework,” *Int. J. Control*, **59**, no. 3, pp. 627–638.

Watanabe, K. and M. Ito (1981). “A process-model control for linear systems with delay,” *IEEE Trans. Automat. Control*, **26**, no. 6, pp. 1261–1269.

Willems, J. C. (1971). *The Analysis of Feedback Systems*, The MIT Press, Cambridge, MA.

Yaniv, O. (1999). *Quantitative Feedback Design of Linear and Nonlinear Control Systems*, Kluwer Academic Publishers, Boston, MA.

Yaniv, O. and P.-O. Gutman (2002). “Crossover frequency limitations in MIMO nonminimum phase feedback systems,” *IEEE Trans. Automat. Control*, **47**, no. 9, pp. 1560–1564.

Zhou, K., J. C. Doyle, and K. Glover (1995). *Robust and Optimal Control*, Prentice-Hall, Englewood Cliffs, NJ.

Zwart, H. and J. Bontsema (1997). “An application driven guide through infinite-dimensional systems theory,” in *Plenary Lectures and Mini-Courses, ECC’97* (G. Bastin and M. Gevers, eds.), pp. 289–328.

תקציר

מערכות שכוללות השהיה במשוב נקראות מערכות עם זמן מת. השיטה הפופולארית לבקרת מערכות עם זמן מת היא סכימת בקרה בעלת קונפיגורציה המכונה מפצה זמנים מתים (מז"מ). רעיון המז"מ הינו לחלץ את ההשהיה מחוץ למשוב ע"י תוספת חוג פנימי אירציונלי לבקר. סכימה המז"מ הוצע לראשונה ע"י אוטו סמית לפני כחמישים שנים (מז"מ של סמית).מאז, המז"מ של סמית וסכמות בקרה רבות שמבוססות עליו הפכו לערכת הכלים הסטנדרטית לבקרת מערכות עם זמן מת. יתר על כן, הוכח לאחרונה שמבנה המז"מ הינו מבנה מהותי לבקרים אופטימאליים לפי מדדי הביצוע H^2, H^∞ ו־L^1, וזה יכול להיחשב כהצדקה אנליטית לקונפיגורצית המז"מ.

ידוע כי סכימת המז"מ יכולה להיות רגישה מאוד לאי וודאות בזמן מת ובעייתית בשלבי היישום. יתר על כן, המנגנונים שעומדים מאחורי הבעיות האלה עדיין אינם מובנים היטב. עבודה זו עוסקת במחקר הבעיות האלה. במיוחד נחקרו שני ההיבטים הבאים של בקרה רובסטית באמצעות מז"מים:

1. הוראה כי בעזרת נימוקים סטנדרדיים של קריטריון ניוקוויסט אפשר לחשוף את הסיבות לרגישותם הגבוהה של מז"מים לאי וודאויות בזמן המת. אפשר כי תופעת הרגישות הגבוהה הזו נגרמת עקב ריבוי תדירויות מעבר (crossover proliferation), ריבוי המתרחש ביתר קלות בגלל שימוש במז"מ. ראשית, נחקר עודף זמן מת עבור תהליך המכיל אינטגרטור והשהיה, המהווה "בעיית בוחן" (benchmark problem) לביצועי מערכות בקרה עם זמנים מתים. במהלך המחקר הופעלו גם הליך של תכנון דו־שלבי קלאסי וגם שיטת ייצוב חוג בשיטת ה־H^∞ האנליטית, כדי להראות שהגדלת רוחב הסרט של החוג הסגור מובילה בצורה בלתי נמנעת לריבוי תדירויות מעבר. הגדלת רוחב הסרט מובילה להורדה דרמטית של עודף הזמן המת שניתן להשיג. כתוצר לוואי של האנליזה המוצעת, הוצג כי אי רציפותו של עודף הזמן המת הינו פונקציה של פרמטרי הבקר. תופעה זו יכולה להטיל ספק ברובסטיות הנומרית של מספר שיטות ידועות לחישוב עודף הזמן המת של מז"מים. לאחר מכן, מורחבת האנליזה המוצעת לתהליכים כלליים יותר באמצעות נימוקי ייצוב חוג קלאסי כמו מעגלי M ו־N וקשר הגבר־פאזה של בודה. לבסוף, הוצגו הוראות תכנון לבקרים המבוססים על מז"מים.

2. שנית, נחקרו השיטות הקיימות ליישום נומרי של בקרים מבוססי מז"מ. הנושא המרכזי כאן הינו קירוב השהיה מפולגת כחלק של הבקר. הדגש העיקרי של המחקר הזה מוטל על שיטה המכונה קירוב בעזרת השהיה מלוכדת, שבה אלמנט ההשהיה המפולגת מקורב באמצעות קטעים מתאימים של השהיה, בדומה לקירוב אינטגרל באמצעות הסכום של רימן. מתברר כי, שימוש בשיטת המגירה הזאת יכול להיות לא בר־יישום באמצעות חומרה, אפילו אם היא בעלת עצמה חישובית רבה בזמן אמת. בעבודה הוצגה הסיבה לכך ־ גידול מהותי של הגבר בלוק המז"מ נגרם ע"י קטבים יציבים מהירים שלו ולכן נוצר מצב שהחוג הפנימי של הבקר אינו מוגדר היטב. הוראה כי הבעיות ניתנות להקלה ע"י שינויים בחוג הפנימי. הרעיון הוא לחלץ את החלק הרציונלי, היציב, מהמז"מ ולצרף ולסגור אותו עם הבקר המרכזי. מבנה זה מפחית את ההגבר של חלק המז"מ ומשפר את הרובסטיות של החוג הפנימי של הבקר. בנוסף לכך, ניתן לראות כי, מקור הקטבים היציבים, המהירים של בלוק המז"מ שמתקבלים בייצוב החוג בשיטת ה־H^∞ הוא פונקצית משקל. הוצג פתרון אפשרי, באמצעות שימוש בפונקציות משקל נון־פרופר, אשר משפר עוד יותר את הרובסטיות של החוג הפנימי של הבקר. הפתרונות שהוצגו נבדקו בניסויי מעבדה.

רשימת טבלאות

3.1 ν עבור LDA של Π עבור h שונות 50
3.2 ערכים עצמיים של המטריצות ההמילטוניות שמתקבלים 53
3.3 ν עבור LDA של Π עבור h שונות (העברת החוג) 56
3.4 ν עבור LDA של Π עבור h שונות (משקלי האי־פרופר) 62

א'.1 פרמטרים הניתנים למדידה 75
א'.2 פרמטרים שזוהו 77

3.15 קירוב בעזרת חלוקת השהיה (LDA) של $\Pi_{\text{irr}}(s)$ 55
3.16 איברי החוג הפנימי לאחר העברת החוג 55
3.17 תגובת התדר של $\Pi_{\text{irr}}\tilde{C}_{\text{s}}$ 56
3.18 תגובת המדרגה עבור $h = 0.1$ 56
3.19 תגובת המדרגה עבור $h = 0.15$ 57
3.20 תגובת המדרגה עבור $h = 0.2$ 57
3.21 מג"ש של קטבי $\hat{P}$ 59
3.22 תכנון עם פונקציות משקל הסטריקטלי־פרופר (W_2) ואי־פרופר ($W_{\bar{\text{p}},2}$) עבור $h = 0.15$ 60
3.23 תכנון עם פונקציות משקל הסטריקטלי־פרופר (W_3) ואי־פרופר ($W_{\bar{\text{p}},3}$) עבור $h = 0.2$ 61
3.24 מג"ש של קטבי $\hat{P}$ עם פונקציות משקל האי־פרופר 61
3.25 תכנון עם פונקציות משקל הפרופר ($W_{\text{p},3}$) ואי־פרופר ($W_{\bar{\text{p}},3}$) עבור $h = 0.2$ 62
3.26 מג"ש של קטבי $\hat{P}$ עם פונקציות משקל הפרופר $W_{\text{p},3}$ ($h = 0.2$) 63
3.27 תגובת המדרגה עבור $h = 0.15$ (תכנון עבור $W_{\bar{\text{p}},2}$) 63
3.28 תגובת המדרגה עבור $h = 0.2$ (תכנון עבור $W_{\text{p},3}$) 64
3.29 תגובת המדרגה של המטוטלת ($h = 0.1$) 65
3.30 תגובת המדרגה של מטוטלת ההפוכה ($h = 0.2$) 66

א'.1 מערכת הניסוי 69
א'.2 מערכת הניסוי () 70
א'.3 חלק עליון של המערכת לאחר פריקה 70
א'.4 איור של מערכת הניסוי 71
א'.5 איור של v_{c} 72
א'.6 ניסוי לזיהוי של I_{r} ו־b 75
א'.7 דיאגראמת בלוקים של מנוע זרם ישיר עם עומס צירי 76
א'.8 תגובת הזמן בניסוי לזיהוי של I_{r} ו־b 76
א'.9 תגובת הזמן בניסוי לזיהוי של ζ 77
א'.10 תגובת התדר עבור מטוטלת $P_{\text{down}}(s) = \begin{bmatrix} P_y(s) & P_\theta(s) \end{bmatrix}'$ 78
א'.11 תגובת התדר עבור מטוטלת הפוכה $P_{\text{up}}(s) = \begin{bmatrix} P_y(s) & P_\theta(s) \end{bmatrix}'$ 78

ב'.1 דיאגראמת בלוקים ליישום של בלוקי ה־FIR המאתחלים 79
ב'.2 סיגנל מיוצר פולסים 80

רשימת איורים

1.1 אופרטור השהיה 5
1.2 מערכות עם השהיה בחוג עבור דוגמאות 1.1 ו־1.2 6
1.3 פונקצית תמסורת של פחית מחוממת והקירוב שלה (דוגמא 1.3) 7
1.4 מוט אחיד חופשי־חופשי (דוגמא 1.4) 8
1.5 מבנה של מערכת בקרה עבור תהליך עם השהיה בחוג 8
1.6 תגובת התדר של $L_{\mathrm{r}}(s)=\frac{5}{s+1}\mathrm{e}^{-sh}$ עבור $h=0,0.1,1$ (דוגמא 1.5) 9
1.7 תהליך עם השהיה וקירוב פדה שלו (דוגמא 1.7) 13
1.8 סכימת בקרת חוזה־משערך 14
1.9 מבנה של מערכת בקרה עם מז"מ סמית 15
1.10 מבנה שקול של מז"מ סמית 15
1.11 מבנה מז"מ כללי 16

2.1 תהליך עם מספר תדירויות מעבר: $L(s)$ (קו רציף) ו־$L(s)\mathrm{e}^{-0.5s}$ (קו מקוקו) 22
2.2 מבנה מז"מ כללי 24
2.3 עודפי היציבות מול הגבר הבקר המנורמל $hk_pC(0)$ (תכנון דו־שלבי) 24
2.4 תיאור ניקולס של $L(s)$ עבור התפשטות תדרי המעבר הראשונה 25
2.5 תיאור ניקולס של $L(s)$ עבור התפשטויות תדרי המעבר השונות (תכנון דו־שלבי) 26
2.6 מבנה של מערכת בקרה עבור ייצוב החוג בשיטת ה־H^∞ 27
2.7 עודפי היציבות מול הגבר הבקר המנורמל $hk_pC(0)$ (ייצוב החוג בשיטת ה־H^∞) 27
2.8 תיאור ניקולס של $L(s)$ עבור התפשטויות תדרי המעבר השונות (ייצוב החוג בשיטת ה־H^∞) 28
2.9 מעגלי M ו־N כרשת בתיאור ניקולס 29
2.10 $C(s)$ עבור תכנונים בחלק 2.2 30
2.11 $\varkappa$ מול $\tilde{\omega}_{\mathrm{c}}$ 34
2.12 ביצועים אופטימאליים כפונקציה של השהיה h בחוג 39

3.1 תיאור ניקולס של $P_{\mathrm{m}}(s)\mathrm{e}^{-sh}$ עבור השהיות שונות 42
3.2 תכנון בשיטת ייצוב החוג הקלאסי עבור $P_{\mathrm{m}}(s)\mathrm{e}^{-sh}$ 43
3.3 ייצוב החוג בשיטת ה־H^∞ עבור $P_{\mathrm{m}}(s)\mathrm{e}^{-sh}$ 45
3.4 מבנה של מערכת בקרה עבור ייצוב החוג בשיטת ה־H^∞ 45
3.5 פונקצית תמסורת של חוג עבור בקר עם מז"מ והקירוב הרציונאלי שלו 46
3.6 קירוב בעזרת השהיה מלוכדת (LDA) של $\Pi(s)$ 48
3.7 מבנה של מנגנון מאתחל (RM) למימוש של $\Pi(s)$ 48
3.8 מימוש של בלוק Π ע"י צורת המשערך 49
3.9 מימוש של בלוק Π ע"י צורת מרחב המצב 49
3.10 תיאור ניקולס של L עבור Π וקירוביו עבור ν שונים 51
3.11 תיאור בודה של $\tilde{C}$ ו־Π עבור השהיות שונות 52
3.12 תגובת התדר של $\tilde{C}\Pi$ עבור הבקרית שמתקבלים 52
3.13 תיאור בודה של איברים של $\Pi=\tilde{P}-\hat{P}\mathrm{e}^{-sh}$ בתדירויות נמוכות 53
3.14 שלבים של העברת החוג 54

3.3.1 יישום מן המגירה 50
3.3.2 מנגנונים של אי דיוק 51
3.3.3 איזון של חוג הבקר: העברת החוג 53
3.3.4 דינאמיקה יציבה מהירה של מז"מ ה־H^∞ 57
3.3.5 שימוש בפונקציות משקל אי־פרופר 59
3.4 בקרה של מטוטלת מעבדתית 63
3.4.1 מטוטלת 64
3.4.2 מטוטלת הפוכה 64

4 מסקנות והצעות להמשך המחקר 67

א' ניסוי מעבדתי 69
1.א' תיאור הניסוי 69
2.א' מידול המערכת 71
2.1.א' משוואות תנועה של החלק המכני 71
2.2.א' משוואות תנועה של החלק החשמלי 73
2.3.א' משוואות המצב ולינאריזציה 74
2.4.א' זיהוי של I_r ו־b 75
2.5.א' זיהוי של ζ 77
2.6.א' פונקצית תמסורת של המערכת 77

ב' פרטי יישום ב־**Matlab** 79
1.ב' יישום של מנגנון מאתחל ב־Simulink 79
2.ב' פונקציות עזר של Matlab 80
2.1.ב' יישום של בלוקי ה־FIR בשיטת RM עבור מערכות SISO 80
2.2.ב' יישום של בלוקי ה־FIR בשיטת RM עבור מערכות SIMO 82
2.3.ב' יישום של בלוקי ה־FIR בשיטת LDA 83

תוכן עניינים

תקציר באנגלית 1

רשימת קיצורים וסמלים 3

1 מבוא 5
1.1 מערכות עם זמן מת 5
1.1.1 השפעות של השהיה בחוג 8
1.1.2 שיטות לבקרת מערכות עם זמן מת 11
1.2 פיצוי זמנים מתים 14
1.2.1 מז"מ של סמית 14
1.2.2 שינוים וחלופות 16
1.2.3 חסרונות של מז"מים 17
1.3 מטרות המחקר 19
1.4 סדר העבודה 19

2 עודף זמן מת של מז"מים 21
2.1 הקדמה: עודף זמן מת 21
2.2 בקרה של אינטגרטור והשהיה 23
2.2.1 תכנון דו־שלבי עם הבקר הראשי הסטטי 23
2.2.2 ייצוב החוג בשיטת ה־H^∞ 26
2.3 הכללות 28
2.3.1 תכנון של מז"מ סמיט עם הגבר גבוה 28
2.3.2 מעגלי M ו־N 28
2.3.3 יחס הגבר־פאזה של בודה 30
2.4 הוראות תכנון 31
2.5 פיתוחים טכניים 32
2.5.1 מז"מ סמיט משופר 32
2.5.2 ייצוב חוג בשיטת ה־H^∞ 36

3 יישום של בקרים הכוללים בלוקי ה־**FIR** 41
3.1 מערכת סרוו עבור מנוע זרם ישיר והשהיה 41
3.1.1 ייצוב חוג קלסי 42
3.1.2 תכנון מז"מ בעזרת ייצוב חוג בשיטת ה־H^∞ 44
3.2 יישום אנלוגי של בקרי מז"מים 46
3.2.1 קירוב של אלמנטים של השהיה מפולגת בעזרת השהיה מלוכדת 47
3.2.2 יישום של השהיה מפולגת בעזרת מנגנון מאתחל 47
3.2.3 השוואות 49
3.3 אנליזה של שיטת LDA 50

המחקר נעשה בהדרכת פרופ/ח' לאוניד מירקין
פקולטה להנדסת מכונות

הכרת תודה

אני אסיר תודה לפרופ/ח' לאוניד מירקין על תמיכתו בי במשך כל המחקר ובתקופת השתלמותי.
תודה לצוות המעבדה לבקרת מבנים גמישים, איליה שמיס וויקטור רויזן, על התמיכה הטכנית בשלב הניסיוני של המחקר.
אני מודה לטכניון על התמיכה הכספית הנדיבה בהשתלמותי.

בקרה רובסטית באמצעות מפצי זמנים מתים

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת תואר
מגיסטר למדעים בהנדסת מכונות

רומן גודין

הוגש לסנט הטכניון — מכון טכנולוגי לישראל

כסלו תשס"ח חיפה נובמבר 2007