H^∞ Optimal Repetitive Control: Continuous-time and Sampled-data Formulations

DISSERTATION

Presented in Partial Fulfillment of the Requirements for

the Degree Doctor of Philosophy in the Graduate

School of The Ohio State University

By

Thaddeus Eldon Peery, B.S., M.S.

* * * * *

The Ohio State University

1995

Dissertation Committee:

Professor Hitay Özbay

Professor Ümit Özgüner

Professor Krishnaswami Srinivasan

Approved by

Adviser

Electrical Engineering

Department

To all those who seak truth and honesty in all things.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my father, Mark E. Peery, for his unwavering support and friendship. Next, I would like to thank my advisor, Professor Hitay Özbay, for his gentle and insightful leadership. A special thanks goes to my first mentor, Dean Daryle Riegle, of Vincennes University. I would also like to thank my other committee members, Professors Özgüner and Srinivasan, for their time, patience, and helpful suggestions. Finally, I thank my minor area Professors, Lee Potter and Anthony Bloch, for their excellence in both teaching and advising.

In addition, I would like to thank all of the members of the "controls group" consisting of Professors Cruz, Özgüner, Özbay, Passino, Utkin, and Yurkovich, along with all of their graduate students. It was the dynamic and energetic nature of this group that convinced me to come to The Ohio State University and remains my favorite thing about the graduate program. The group members who have been with me my entire time (or almost my entire time) here deserve special mention for their kind and generous help and comradery. They are (in alphabetical order): Kevin Burgess, Stewart DeVilbiss, Julie Hurtig, Layne Lenning, Keith Redmill, Wu-Chung Su, Onur Toker, and John Watkins. Some other group members whose time here overlapped with mine to a lessor extent, but whose help and support were especially significant, include: Tony Bailey, Dan Clancy, Sergey Drakunov, Lew Fulcher, Jim

Gassman, Eric Laukonen, LaMoyne Porter, and Dave Schoenwald. Other group members who contributed to the group dynamic sufficiently to make a lasting impression include: Scott Brown, Cem Hatipoglu, Hubert Chan, Murat Dogruel, Mindy Everhart, Francisco Green, Franz Kautz, Andrew Kwong, Jeff Layne, Will Lennon, Sheila Longshore, Alfonsus Lunardhi, Amit Magan, Sashonda Morris, Vivek Moudgal, Tim Murphy, Jim Nivens, Shyamala Raghunathan, Martin Sikora, Martin Sommerville, Jeff Spooner, Cemil Ulus, Moeljono Widjaja, Jin Yang, Murat Zeren, and Mike Zimmerman. Finally, there are some other students who, while not officially part of our group, contributed significantly to the group dynamic. They include: Scott Gray, Mark Hanes, Chris Masgras, Scott McMillan, Wendy Morrow, Bill Pearson, Jim Schrock, Aneil Shah, Kevin Smith, Steve Stitt, and Konur Unyelioglu.

I would also like to thank several people from my professional past. First, I would like to thank my advisors from my M.S. at Purdue University, Professors John Chiasson and Stanislaw Żak. I would like to thank several people from my internship at the Honeywell Technology Center: Jim Krause (supervisor), Mary Jo Hoffman (internship mentor), Blaise Morton (friend/mentor), and Jen Sly (friend/office mate). I would also like to thank several people from Electrospace Systems, Incorporated: Mike Wolfe (supervisor), Les Allen (friend/program manager), the late Jeff Boughton (friend/office mate), Dorothy Metcalf (friend/secretary), Terry Beamer (department head), Bob Landis (friend/reliability engineer), and George Weber (office mate).

I wish to thank several people for specific technical assistance. I thank Onur Toker for providing the software used in Chapter IV. I thank Professor Bamieh (of the

University of Illinois) and his student Dan Stanek for providing additional details on the proofs in [5] and for valuable feedback on early drafts of [74], respectively. Finally, I would like to thank LaMoyne Porter for helpful discussions on genetic algorithms.

Last, but not least, I would like to thank my Ohio State office mates, Behrouz Khayatian and Murat Zeren, for their pleasant companionship and fellowship, and "Professor Doug" the friendly Maintenance Engineer for relentlessly providing good cheer and lively banter.

VITA

May 28, 1963	Born - Chester County, PA
January 84 - June 85	Pre-Electrical Engineering Vincennes University Vincennes, Indiana
Summer 86	Summer Intern Research and Engineering Lab General Dynamics Fort Worth Division
August 1987	B.S. in Electrical Engineering Purdue University, West Lafayette, IN
August 87 - January 89	Graduate Instructor Fellow Purdue University, West Lafayette, IN
December 1988	M.S. (Electrical) Engineering Purdue University, West Lafayette, IN
January 89 - September 91	Member Technical Staff Electrospace Systems Incorporated Richardson, Texas
September 91 - August 92	University Fellow The Ohio State University Electrical Engineering Department
September 92 - June 94	Research Associate The Ohio State University Control Research Laboratory
Summer 93	Doctoral Intern (researcher) Honeywell Technology Center Minneapolis, MN
July 94 - June 95	University Fellow The Ohio State University Electrical Engineering Department

Publications

Research Publications

T. E. Peery and H. Özbay "On H^∞ Optimal Repetitive Controllers", *Proceedings of the 32nd IEEE Conference on Decision and Control*, San Antonio, Texas, pp. 1146-1151, December 1993.

H. Özbay and T. E. Peery "On fixed order controllers for delay systems: Discrete-Time case". *Proceedings of the American Control Conference*, San Francisco, CA, pp. 1030-1031, June 1993.

Fields of Study

Major Field: Electrical Engineering

Major Area of Specialization: ..... System and Control Theory

Minor Areas of Specialization: .....Signal Processing/Communications
Applied Mathematics

TABLE OF CONTENTS

List of Tables

List of Figures

CHAPTER I

Introduction

Control systems are fundamental to making modern technology possible. Indeed, control systems touch our lives, both directly and indirectly, many times a day. Control systems effect our lives directly through: household appliances such as microwave ovens, modern washers and dryers, and VCR's; climate control systems (heating, cooling, and humidity regulation) in our homes, offices, and automobiles; and modern transportation systems including buses, airplanes, trains, and automobiles. The indirect ways that control systems touch our lives include all of the above and many more, because none of these things could have been manufactured without control systems. Which brings us to the focus of this dissertation: control of manufacturing systems and machinery. Specifically, we will be concerned with improving the performance and reliability of machines that *repeat* a given task over and over again. The branch of control theory dedicated to enhancing the performance and reliability of systems that perform the same action over and over is called "repetitive control theory". The class of such systems encompasses much more than just manufacturing machinery and includes such everyday modern conveniences as computer disk drives. The specific application we address is an electro-hydraulic material tester, see [56]. However, the techniques developed have universal applicability.

1.1 Feedback control systems

A few of the applications of control theory mentioned above perform satisfactorily using "open loop" control. Open loop control is a control scheme where a "command" is given to the system and it is simply assumed that the system "obeys" the command. When the mathematical model of the system is both highly accurate and invariant under changing operating conditions and precise operation is not required, open loop control can be effective. An example of such a system is your favorite vending machine. While there are "feedback controllers" internal to the vending machine, the external behavior is open loop. That is, the machine does not check whether the previous purchase has been removed before going ahead and sending out the requested merchandise. A "feedback controller" is a controller that senses the "output" of the system and uses this information along with any externally generated command to determine the appropriate "actuator" input. The actuator is the device by which the controller exerts influence on the system. Most typically, the actuator is an electric motor, e.g. the compressor motor that comes on to cool your refrigerator when the difference between the external command (the desired temperature setting) and the measured (sensed) output (the temperature inside the refrigerator) exceeds some fixed limit. Feedback control must be used whenever there is enough uncertainty about the system that we can't just turn the controller loose. For instance, in order for a refrigerator's temperature to be adequately regulated using open loop control, the engineer would have to know: the exact temperature of the room in which the refrigerator would be operating, when and for how long the refrigerator would be

opened, when, and how much, food would be put in, etc. The refrigerator is an extreme example, where the operating environment is completely unpredictable, but other factors can cause sufficient uncertainty to require feedback control. There can be significant uncertainty about the nominal plant model, irrespective of changes in the operating environment.

Consider modern automobiles and their computer (feedback) controlled ignition systems. You've probably noticed that there is very little reduction in overall performance, with respect to starting and running smoothly, until the electrical components, such as wires or plugs, are completely worn out. In the "good old days", when the ignition systems were essentially open loop, a single bad spark plug would make an engine run rough. This type of internal change in the system, that happens over relatively long periods of time, is one source of what is known as "model uncertainty". There are many sources of model uncertainty, most of which are specific to a given plant. The fact is, we *never* have a completely accurate mathematical description of *any* plant. That is why we call the mathematical description a model of the plant. When both the operating environment and the plant model are known with sufficient accuracy, open loop control can be used. The reason that feedback control is such a big deal, aside from the greatly improved performance in the face of uncertainties, is that feedback control can make a system go "unstable". Where, by unstable, we mean that some system property will grow without bound, or at least exceed acceptable limits. The example of feedback resulting in instability that almost everyone is familiar with is the horrible scream that ensues when a microphone is placed in

front of a loud speaker to which it is connected. This tendency to cause instability is what makes feedback control so challenging. Furthermore, since one of the main reasons that we are using feedback control is the presence of significant uncertainty, we must ensure the robust stability of the feedback control system in the face of these uncertainties.

1.2 Repetitive control

Repetitive control is a method for obtaining "good" system performance when the system is subjected to periodic external signals. What we mean by good performance depends on whether the external signal is a command (reference signal) that we want the output of the system to follow (track), or a disturbance input (variation in the operating environment) that we want the output to ignore (reject). The basic diagram illustrating the standard control problem is shown in Figure 1. The reference signal, r, is the command we want the output, y, of the plant to track (match) in face of the disturbance, d. The feedback controller, C, acts on the error signal, e, to produce the control input, u, to the plant, P, to produce the desired response, i.e. $y = r$. Clearly, signals for which the "gain" of the controller, C, is large will tend to have their effects eliminated from the error signal, e. This is precisely the idea behind repetitive control, i.e. we want to design controllers that have high gain for periodic signals with a given period of T seconds. The basic concept of repetitive control is illustrated in Figure 2. Assume that the error input, e_r, is periodic with period T. Then, the repetitive controller output, u_r, is a constantly growing scaled version of the error input, e_r. For purposes of illustration, assume that the periodic error signal,

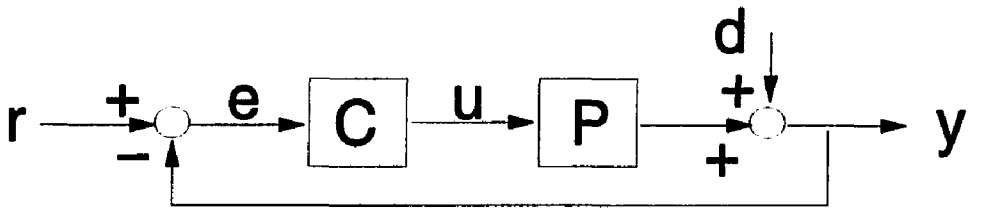


Figure 1: Basic feedback control problem.

$e_r(t)$, is zero for $t < 0$, and then periodic with period T, $e_r(t) = e_r(t+T)$, for $t > 0$. Over the first period, $0 \leq t < T$, the controller output, $u_r(t)$, is zero. Over the second period we have $u_r(t) = e_r(t-T) = e_r(t)$, and then over the third period we have $u_r(t) = e_r(t-T) + u_r(t-T) = 2e_r(t)$. Similarly, over the fourth period we have $u_r(t) = 3e_r(t)$, and so forth. This behavior can be thought of as being marginally unstable, where it is only marginally unstable because it only grows without bound in response to periodic error signals. Clearly, as $t \to \infty$, the gain on the periodic signal goes to infinity. Thus, if this controller could be used in the feedback control system of Figure 1, the error response to all periodic signals, whether references or disturbances, would be driven to zero. Unfortunately, as a practical matter, certain modifications must be made to the repetitive controller of Figure 2 in order for the feedback control system of Figure 1 to be stable, see Chapter II. Thus, while the repetitive controller in Figure 2 is ideal with respect to performance, it is pathological with respect to robust stability, even without taking plant uncertainty into account.

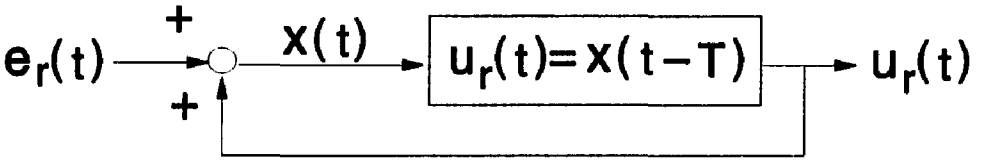


Figure 2: Basic repetitive control concept.

1.3 H^∞ optimal control

The primary advantage of H^∞ optimal control is that it provides a framework for quantifying the trade-off between performance (reference tracking and/or disturbance rejection) and robust stability in the face of plant model uncertainties. This trade-off is quantified in terms of the frequency response of the feedback control system from the external inputs, r and d, to the output, y, and the error signal, e. The two key transfer functions are: the sensitivity function, S, which is the transfer function (frequency response) from the reference input, r, to the error signal, e, and the complementary sensitivity function, $T := 1 - S$, from the reference input, r, to the output, y. The complementary sensitivity function, T, quantifies robustness with respect to plant model uncertainties. Thus, we need $|S(j\omega)|$ small in the frequency range where we require good performance, and we need $|T(j\omega)|$ small in the frequency range where we have significant plant uncertainty. Clearly this constitutes a trade-off, due to the relationship, $T := 1 - S$, between T and S. Note that the transfer function from the disturbance input, d, to the error signal, e, is $-S$. Thus, since it is the magnitude that characterizes performance, S quantifies good performance both with respect to reference tracking and disturbance rejection. Typically, we

need good performance at "low" frequencies (the performance region), and we have significant plant uncertainty at "high" frequencies (the robustness region). There is always increasing uncertainty in the plant model as frequency increases. Furthermore, high frequency plant uncertainty always places an upper bound on the achievable performance.

In a sense the combination of repetitive control and H^∞ optimal control theory is a perfect marriage. While ideal repetitive control achieves perfect performance with no robustness, H^∞ optimal control theory quantifies the trade-off between performance and robustness. In order to more fully understand the relationship between the two design techniques, we must take a closer look at what it means to have repetitive performance. The performance of repetitive control systems can be quantified in terms of the magnitude of the sensitivity function at a particular fundamental frequency, $\omega_0 := T/2\pi$, and a number of harmonic frequencies, $k\omega_0$, where k comes from a predetermined set of integers. Specifically, performance is measured by how small the sensitivity function is at these harmonics. We will refer to the finite frequency region, from $\omega = 0$ to the highest harmonic of interest, as the performance region. The plant model must be very accurate in this region. Similarly, we will refer to the semi-infinite frequency region from the point where the plant uncertainty becomes significant to $\omega = \infty$ as the robustness region. Clearly, the robustness and performance regions must be disjoint. From the small gain theorem, see e.g. [2, 49], if the system loop gain is less than one, then the system is stable. Note that the small gain theorem is satisfied whenever $|T(j\omega)|$ is "small", i.e. $|T(j\omega)| << 1$. High frequency robustness

is typically achieved in this way. Indeed, all controllers corresponding to "good" designs, have a small gain region, where the controller gain is sufficiently small to ensure that the system loop gain is less than one for all possible plant variations. Between the performance region and the small gain region, there is a frequency band, where robust stability must be ensured by other means. The narrower this transition band becomes, the more difficult it becomes to robustly stabilize the system.

1.4 Sampled-data systems

The world we all live in exists in continuous-time, i.e. there is a continuous flow from one event to the next, and there is no discrete quantization of moments in time at which events can occur. This is in stark contrast to what goes on inside a computer. In computers, things can *only* occur at discrete instants and time is measured in integer numbers of discrete steps. Thus, computer computations exist in a very different world, where everything occurs in discrete-time. Systems whose operations are carried out in discrete-time are called discrete-time systems. Obviously, the dynamic behavior of discrete-time systems is fundamentally different from the dynamic behavior of continuous-time systems. Every physical system that we may want to control exist in continuous-time, even discrete event systems, such as vending machines. Thus, whenever we attempt to use computer control we are interfacing two very different worlds. There are two interfaces that must be considered: the interface from the continuous-time world to the discrete-time world, and the interface from the discrete-time world to the continuous-time world. The interface from the continuous-time world to the discrete-time world is done by sampling the

continuous-time signals, to obtain discrete-time sequences. Just as the clock rate of the microprocessor defines the relationship of the discrete-time world of the computer to the continuous-time world in which the computer exists, the sampling rate, f_s, used to approximate a continuous-time signal by a discrete-time sequence, defines the relationship between the continuous-time signal and its discrete-time approximation. Specifically, the interface from the continuous-time world to the discrete-time world is done by sampling the continuous-time signal at discrete (or as nearly discrete as possible) time instants. Typically, the sampling is done at equally spaced time instants one sample period, τ, apart. The interface from the discrete-time world to the continuous-time world is done via a, typically constant, hold function. The constant (zero order) hold function simply remains constant at the value of the preceding element of the discrete-time sequence for one sample period, $\tau := 1/f_s$, where f_s is the sampling rate (frequency). Any system composed of a continuous-time part and a discrete-time part, is a sampled-data system. The importance of sampled-data systems in today's world of inexpensive, high speed computers cannot be over stated. In many cases, by far the most economical and reliable means of implementing control laws is by means of a computer, resulting in a sampled-data system. Historically, such systems have been designed considering the above interfaces to be ideal and using a high enough sampling rate, f_s, that the approximation is adequate, see Chapter II. In this dissertation we explore the recent trend toward directly considering the design of controllers for sampled-data systems, i.e. design of discrete-time controllers directly for continuous-time systems, without making any approximations. Specifically, we

apply sampled-data theory to H^∞ optimal repetitive control design.

1.5 Organization of the Dissertation

In Chapter II we give some mathematical rigor to the terms: "repetitive control", "H^∞ optimal control theory", and "sampled-data systems". We also define, in general terms, the problems addressed by this dissertation.

In Chapter III we pose a robust performance repetitive control problem in an H^∞ optimal formulation. This formulation is shown to lead to a "generalized" repetitive controller structure, which has implications for "classical" repetitive control design. We derive an extension to existing infinite dimensional H^∞ optimal control theory, which solves this formulation of the repetitive control problem. Finally, we present a numerical example, for an electro-hydraulic material tester, that illustrates the validity of the theoretical development.

In Chapter IV we consider nominal performance with robust stability H^∞ optimal repetitive control formulations. The resulting two-block H^∞ optimal control problem is solved using a specialized version of a considerably more general result on H^∞ optimal control of single input single output (SISO) systems. The solution of the two-block H^∞ problem is shown to have two interpretations: one as a direct design of H^∞ optimal repetitive controllers and another as a sensitivity improvement formulation. The sensitivity improvement formulation leads to a novel repetitive control structure, which can be viewed as a cascade repetitive structure. Numerical examples are done to illustrate the effectiveness of the two-block formulation for practical repetitive control design.

In Chapter V we propose a definition for "sampled-data repetitive control" and formulate a problem satisfying our definition. We derive an extension, to the existing results on sample-data systems, to a more general class of problems which includes our formulation of the sampled-data repetitive control problem. We give the solution to the sampled-data repetitive control problem in terms of the discrete-time equivalent system.

In Chapter VI we summarize the contributions of the dissertation and discuss future research directions suggested by the results of this dissertation.

In Appendix A we list the rather extensive mathematical notation required to rigorously define the research.

In Appendix B we give a detailed (and highly mathematical) derivation and discussion of the main result, Theorem 3.2.1, for the infinite dimensional H^∞ optimization problem arising from the robust performance formulation of Chapter III.

In Appendix C we discuss the issues and difficulties in obtaining exact minimal state space realizations of transfer function matrices (transfer functions for multi-input multi-output (MIMO) systems). We give a detailed outline of a proposed procedure for obtaining exact minimal realizations. We also present the details of the exact minimal realization required to complete the numerical example in Chapter III.

In Appendix D we give a detailed outline of the proof of our extension of the discrete-time equivalent system result of Bamieh and Pearson [5] for H^∞ optimal sampled-data control. This extension is required to solve the sampled-data repetitive control formulation of Chapter V.

CHAPTER II

Mathematical Definitions and Problem Statement

In this chapter we give precise mathematical definitions for the three main topics of this dissertation: repetitive control, H^∞ optimal control theory, and sampled-data systems. We also give a brief history of these research topics as it relates to the problems addressed in this dissertation. Using the precise mathematical definitions, we define, in general terms, the design problems to be addressed.

2.1 Repetitive Control Theory

Repetitive control is used in numerous industrial applications: electro-hydraulic materials testing [56, 80, 79], computer disk drives [20], motor speed control [51], trajectory control [66], non-circular machining [99, 98], friction compensation [100], and robotic manipulators [76]. The advantage of repetitive control is that it allows tracking (or rejection) of periodic signals with significant harmonic content using controllers with low order rational components. The repetitive controller structure of C_1 in Figure 3 is typical, where the time delay, T, is the period of the external input to be tracked of rejected. The repetitive part of the controller is the positive feedback loop with the delay term, e^{-sT}, where we have added a pre-filter, $q(s)$, to the ideal case shown in Figure 1. Stability analysis of this, and other, repetitive controller structures and

guidelines for the selection of $b(s)$ and $q(s)$ can be found in [39, 66, 76, 79, 97]. Let the repetitive part of the controller $C_1(s)$ be represented by

$$R(s) := \frac{q(s)e^{-sT}}{1 - q(s)e^{-sT}}, \quad \text{i.e. } C_1(s) = 1 + b(s)R(s). \tag{2.1}$$

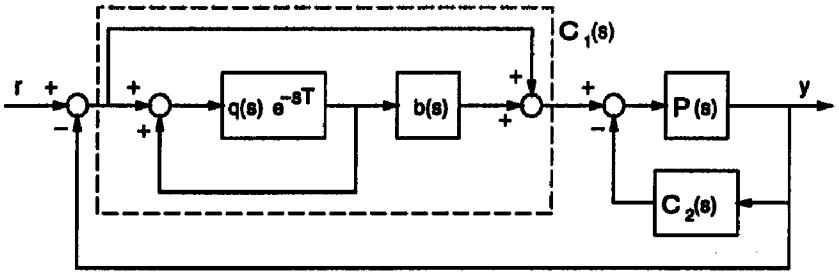


Figure 3: Typical two degree of freedom repetitive controller.

Remark 2.1.1 *For $q(s) \equiv 1$, we have $R(j\omega_0 k) = \infty$, for all integers k, where the fundamental frequency, $\omega_0 := 2\pi/T$.*

Thus, if the closed loop system could be stabilized for $q(s) \equiv 1$, we could achieve perfect tracking and rejection of arbitrary periodic signals. This can be viewed as an extension of the internal model principle, see [39].

Remark 2.1.2 *For $q(s) \equiv 1$, the closed loop system of Figure 3 cannot be stabilized for any real plant, $P(s)$, i.e. for any plant with non-zero response at infinite frequency (strictly proper plant), see [39].*

Thus, the repetitive controller parameter, $q(s)$, quantifies a trade-off between performance and (robust) stability. In terms of the usual measure of performance, the sensitivity function, we have that $|S(j\omega)| = 0$ whenever $|C_1(j\omega)| = \infty$ which occurs whenever $|R(j\omega)| = \infty$. Now, we can define repetitive performance in terms of the sensitivity function.

Definition 2.1.1 (repetitive performance) *We say that a controller provides repetitive performance, if* $|S(j\omega_0 k)| \approx 0$, *for all integers* k *(harmonics) of interest.*

Clearly, repetitive performance can be achieved without using repetitive control, at the cost of using very high order controllers. In particular, the repetitive controller, $R(s)$, could be replaced with a purely rational (finite dimensional) compensator with two poles (at $\pm j\omega_0 k$) for each harmonic of interest. This is exactly the trade-off in using repetitive controllers, a dramatic decrease in the order of the rational components, at the cost of introducing the infinite dimensional component, e^{-sT}. Similarly, it would be possible to choose the repetitive controller parameter, $q(s)$, to eliminate repetitive action and choose $b(s)$ to be a high order rational compensator providing repetitive action.

Definition 2.1.2 (repetitive action) *We say that a repetitive controller provides repetitive action, if* $q(j\omega k) \approx 1$, *for all of the harmonics of interest.*

When using an abstract design process, it is possible to come up with high order, or even infinite dimensional, $q(s)$ satisfying Definition 2.1.2. Thus, we introduce the following definition for the classical notion of low order rational transfer functions $q(s)$ that provide repetitive action.

Definition 2.1.3 (classical repetitive action) *We say that a repetitive controller provides classical repetitive action, if* $q(s)$ *is a low order rational transfer function that provides repetitive action, i.e. if* $q(s)$ *is unity-low pass.*

Any control system with two distinct controller blocks, such as $C_1(s)$ and $C_2(s)$ in Figure 3, is a two degree of freedom controller. Obviously, using a two degree of

freedom controller provides more options with respect to how to approach the design problem.

Remark 2.1.3 *Two degree of freedom controllers, such as the controller shown in Figure 3, can be used to decouple the design for* nominal *performance from the design for robust stability [102, 39].*

In repetitive control a slightly different use is made of the two degree of freedom controller formulation. The controller $C_2(s)$ is taken to be an initial (robustly) stabilizing controller, that is also designed to make it relatively easy for the component $b(s)$ of the controller $C_1(s)$ to approximately invert the stabilized plant over the performance region. For more on approximate inversion of stabilized plants see Chapter IV.

Remark 2.1.4 *It is quite natural to use a two degree of freedom controller structure for repetitive control, because the performance is provided primarily via repetitive action, which also tends to destabilize the system.*

Indeed, all repetitive controllers have a two degree of freedom structure. While the structure shown in Figure 3 is fairly typical, there are many other structures. The only thing that all repetitive control systems have in common is a two degree of freedom structure and the inclusion of a repetitive block (2.1).

Remark 2.1.5 *Each of the formulations, of the continuous-time repetitive control problem addressed in this dissertation, leads to a novel repetitive control structure, with possible implications for classical repetitive control design.*

The only other class of repetitive controllers is discrete-time repetitive controllers. The main difference of discrete-time repetitive controllers is that the delay term is no longer infinite dimensional in discrete-time. Since all real systems exist in continuous-time, it only makes sense to talk about discrete-time repetitive controllers in the context of sampled-data systems. Thus, a detailed discussion of discrete-time repetitive control is given in Section 2.3, where we discuss sampled-data systems.

2.2 H^∞ Optimal Control Theory

H^∞ optimal control theory is a norm based optimization method for obtaining control designs. While no norm based optimization method allows the incorporation of *all* design requirements, the H^∞-norm based method allows most design requirements and the key trade-off of performance (in terms of tracking or rejection of external signals) versus robust stability. The reason that H^∞ optimization is so useful is that it allows the use of all the frequency based design techniques from classical control theory, see e.g. [35, 53, 64]. Specifically, see [33], the H^∞-norm (or simply the infinity-norm) of a transfer function, F, is defined by

$$\|F\|_\infty := \sup\{|F(s)| \ : \ Re(s) > 0\} = ess \sup_{\omega \in \mathbf{R}} \{|F(j\omega)|\}, \tag{2.2}$$

where the second equality holds by the maximum modulus theorem. Thus the infinity norm of a transfer function is just the peak Bode magnitude, or maximum gain of the system. The connection to the Bode magnitude is precisely what makes the H^∞ optimal control methodology so useful, by allowing the application of the great wealth of knowledge about control design using frequency shaping, see [35, 53, 64, 22].

The frequency shaping is done through the use of weighting functions, which are multiplied by certain transfer functions characterizing system properties, prior to evaluating the norm (or undertaking the norm-based optimization). The two most commonly used system transfer functions are the sensitivity function, $S(s)$, and the complementary sensitivity function, $T(s)$, introduced in Chapter I. The sensitivity function, $S(s)$, is weighted by the performance weight, $W_1(s)$, and the complementary sensitivity function, $T(s)$, is weighted by the robustness weight, $W_2(s)$. The H^∞ optimal control problem is to minimize the infinity-norm of a weighted transfer function (or combination of transfer functions) over all stabilizing controllers. A typical formulation, and the one used in Chapter IV, is the two-block (nominal performance with robust stability) formulation given by the minimization, over all stabilizing controllers, of the performance measure

$$\gamma := \left\| \begin{bmatrix} W_1 S \\ W_2 T \end{bmatrix} \right\|_\infty := \left\| \left(|W_1 S|^2 + |W_2 T|^2 \right)^{1/2} \right\|_\infty . \tag{2.3}$$

Clearly, wherever a given weighting function is large, the optimization process will tend to make the corresponding transfer function small. Obviously, we must make the magnitude of the robustness weight, $|W_2(j\omega)|$, large where the plant uncertainty is high, i.e. in the robustness region. Furthermore, we typically make $|W_2(j\omega)|$ small elsewhere so as not to interfere unnecessarily with obtaining other objectives. Similarly, the magnitude of the performance weight, $|W_1(j\omega)|$, is typically chosen to be large in the performance region and small elsewhere. More precise shaping of the weights, W_1 and W_2, depends on problem specific information.

From the definition, (2.3), of the two-block problem it is not at all clear how to find the optimal solution, i.e. it is not clear how to search over all stabilizing controllers. The solution to this difficulty is the Youla parameterization of all stabilizing controllers, see e.g. [33], in terms of a free parameter, Q, in the Hardy space H^∞.

Definition 2.2.1 *The Hardy space, see e.g. [41, 75, 91], H^∞ is defined to be the set of all complex valued functions, $H(s)$, of the complex variable, $s = \sigma + j\omega$, $\sigma, \omega \in \mathbb{R}$, that are bounded and analytic in the open right half plane, i.e. for $\sigma > 0$.*

The Youla parameterization is given in terms of *any* factorization, $P = N/D$ with $N, D \in H^\infty$, of the plant and *any* solution of the associated Bezout identity

$$NX + DY = 1, \quad \text{where} \quad X, Y \in H^\infty. \tag{2.4}$$

Now, we can give the Youla parameterization of all stabilizing controllers for the generic one degree of freedom feedback control system of Figure 4:

$$C = \frac{X + DQ}{Y - NQ}, \quad \text{where} \quad Q \in H^\infty. \tag{2.5}$$

Remark 2.2.1 *All one degree of freedom controllers, regardless of configuration, are stabilized by all controllers parameterized by (2.5).*

A simplified parameterization is possible when the plant is stable (or has been stabilized). Specifically, we can let $N = P$, $D = 1$, $X = 0$, and $Y = 1$, which gives

$$C = \frac{Q}{1 - PQ}, \quad \text{where} \quad Q \in H^\infty. \tag{2.6}$$

This simplified parameterization comes into play quite often in repetitive control, due to the initial stabilization step, see Chapter IV.

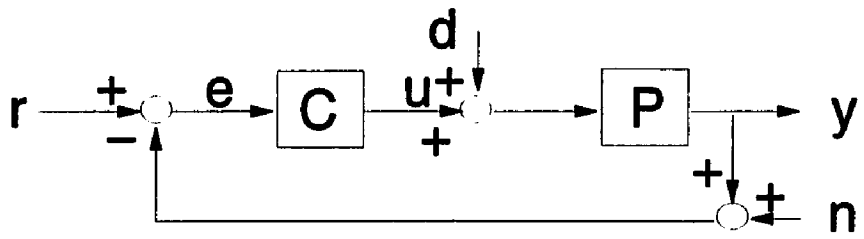


Figure 4: Generic one degree of freedom controller.

Using the Youla parameterization (2.5) we can restate the two-block formulation in terms of the free parameter, $Q \in H^\infty$. We restate the two-block problem for the generic configuration, see Figure 4. The optimal performance is given by

$$\gamma_o := \inf_{Q \in H^\infty} \left\| \begin{bmatrix} W_1 DY \\ W_2 NX \end{bmatrix} - \begin{bmatrix} W_1 DN \\ -W_2 DN \end{bmatrix} Q \right\|_\infty . \tag{2.7}$$

This form is suitable for optimization and has the structure of a model matching problem, see e.g. [33, 22]. Note that, since the infinity-norm is a supremum (maximum) operation, the H^∞ optimal control problem can be viewed as a min/max optimization problem. While this min/max property is of interest to mathematicians and has led to some related control formulations, it is not significant to the specific problems addressed in this dissertation.

An interesting, but not especially desirable, property of the H^∞ optimal formulation is that the optimal solution results in perfectly "flat" performance, i.e. the infinity norm is attained at every frequency. The drawback of this property is that it means that the optimization process "takes everything we say literally", i.e. everything must be posed in terms of equality constraints even though the actual design requirements are inequality constraints. Further complicating the situation is the fact

that, the optimization problem cannot be solved for arbitrary nonlinear weights, e.g. weights that are identically zero over certain frequency ranges. However, the actual design requirements are exactly of this form, i.e. we have *no* performance requirement in the robustness region and typically *no* robustness requirement in some (if not most) of the performance region. Through intelligent selection and adjustment of the weighting functions, it is usually possible to get around these difficulties, at least for relatively simple problems. More general frameworks are needed for complex problems, see e.g. [59]. Another framework for intelligent weight selection for H^∞ optimal control comes from μ-syntehesis [23, 72], in the "D-scales" and "D-K Iteration" techniques. These techniques, in their present form, are only applicable to finite dimensional problems. This dissertation is concerned with infinite dimensional (delay system) problems, and we must therefore depend upon intelligent selection and adjustment of our weighting functions. When we find that a particular portion of one or our weighting functions is being "taken more literally" than we would like, i.e. is determining the optimal performance level, when that point is not that important, we can adjust the shape of the weighting function by using a more complicated (higher order) weight.

Remark 2.2.2 (controller order) *For most optimization methods, including H^∞, the order of the "optimal" controller is proportional (typically greater than or equal) to the sum of the order of the plant and the total order of all weighting functions, for finite dimensional problems.*

For infinite dimensional problems, the order and complexity of the optimal controller are generally even greater, see [95] for the *SISO* (single input single output) case and Chapter III for the *SIMO* (single input multi-output, or vector) case. (The *MIMO* (multiple input multiple output) case has been solved only for a restrictive class of plants and weighting functions, see [70].) Thus, we have to be careful about increasing the order of our weighting functions, since it will lead to higher order "optimal" controllers. The combined order of the plant and the lowest order feasible weights often results in controllers that are too high order. For this reason, reduced order approximation (model order reduction) is often employed to obtain a more practical (better) controller from the "optimal" solution, see Chapter IV. When optimization techniques yield controllers that are too high order, or are otherwise impractical, the "optimal" solution can still be quite useful if it provides insight into the characteristics of "good" solutions, see Chapter III. To date there has been little progress with respect to incorporating a penalty on controller order into the "optimal" design problem. A suitable framework for digital implementations has been posed in [68] which depends on the ability to find the optimal controller of a given order. The problem of finding the optimal controller of a given order remains an open research topic.

2.3 Sampled-data Systems and Control Theory

Today most controllers are implemented on digital computers. Historically, this has been done either by means of a discrete-time approximation of a controller designed in continuous-time or by designing a discrete-time controller using a discrete-time

approximation of the plant, see e.g. [65]. Neither of these methods address inter-sample behavior. Stability analysis has been addressed in [34, 17], and robustness is addressed in [82]. Furthermore, these approximation methods require very high (five to ten times the "bandwidth") sampling rates in order to achieve reasonable approximations. Finally, these approximation methods necessarily provide suboptimal performance. These problems have given rise to an exciting new area of research called *Sampled-Data Systems.*

In sampled-data system design, the fact that the plant exists in continuous-time while the controller is implemented in discrete time is directly incorporated into the design process. There are three main approaches to this problem: jump system methods [90], game theoretic methods (see e.g. [6, 7, 93]), and methods using a *lifting* technique (see e.g. [5, 104, 4, 16, 18, 15, 14, 105, 19]). (The lifting technique is described in detail in Chapter V.) The game theoretic results are a special case of the jump system results [90]. The problem with the jump system results (and therefore the game theory results) is that they give solutions that are not practically implementable (as detailed below). The lifting based methods, on the other hand, provide a completely general framework for controller designs that can be readily implemented with existing technology. The lifting technique generates discrete time representations of continuous-time systems which preserve algebraic operations on systems, as well as signal and system norms, see e.g. [5, 104, 4]. The main difficulty is that the states of the lifted systems are function-valued rather than real-valued. For LTI systems, it has been shown that the resulting state space realization has

dimension less than or equal to the original system, see [5]. Using this fact, Bamieh and Pearson [5] show how to construct an "equivalent" finite dimensional system with a real-valued state space. (A very closely related, but quite different, finite dimensional discrete-time equivalent system was derived by Kabamba and Hara [46].) The equivalence is in the sense that internal stability and the infinity norm of the system (H^∞ control problem) are preserved.

2.3.1 Approaches to the sampled-data problem

In its most general form (see Figure 5) the sampled-data problem includes a continuous-time generalized plant model, G, (which includes any weighting functions), a continuous-time pre-filter, F, (which is often included in the plant model), a sampler, $\mathcal{S}$, (which is usually assumed to be ideal), a discrete time compensator, and a hold function, $\mathcal{H}$, that constructs a continuous control signal from the discrete output of the compensator. The weighting functions and design criteria should be posed in continuous time, since this is the environment in which the real plant exists. The pre-filter is typically included in the plant model because it is generally part of the sensor that generates the sampled data and is thus fixed at the control design stage. Furthermore, elaborate pre-filters are impractical to the extent that analog control is impractical. In almost all cases the sampling is assumed to be ideal. Real samplers are not ideal, but if their time constants are "fast" compared to the sensed signals, the approximation is good. The integral nature of real samplers is addressed in [67]. The hold function is typically taken to be fixed. Usually this fixed hold is taken to be the so called zero-order hold, which generates piecewise constant outputs. Arbitrary time varying

hold functions are considered in [90, 89, 6, 7], and arbitrary fixed hold functions are addressed in [46]. Most D/A (digital to analog) converters, that are commercially available, are essentially zero-order holds. Any physically realizable hold function must be essentially fixed, i.e. not time-varying. The only exception to this would be something of the nature of a digitally programmable filter, in which case the hold still must be from a fixed set of time invariant holds. Finally, the sample and hold functions are almost always assumed to be synchronized. While no physical system has perfectly synchronized sample and hold functions, many systems do operate with very near synchronization. Asynchronous sampled-data systems are considered in [103].

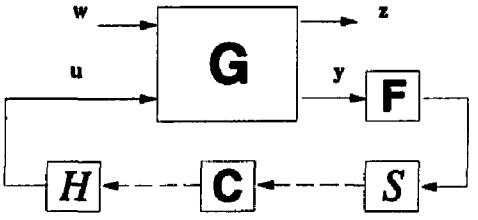


Figure 5: General form of the sampled-data problem.

From the above it appears that the most practical approach is to include the pre-filter in the plant model and assume ideal sampling and fixed zero-order hold. This is the approach of [5, 17, 16, 15, 4, 82, 25, 57, 14, 50, 19, 92, 81]. We will also follow this approach. The other approaches, while not as practically applicable, may yield useful insights into the nature of the problem. Some other approaches are:

- Arbitrary time varying hold function design is considered in [6]. In this paper Basar shows that for the state feedback case this leads to a constant

(memoryless) discrete controller and hold function that is a nonlinear function of time. In [90] it is shown that for LTI plants with output feedback the hold is of the form Ce^{At} and the discrete controller has an observer type structure.

- Simultaneous design of the pre-filter and the discrete compensator is addressed by Sun et al. in [89] for LTI plants with output feedback. Toivonen uses this approach with respect to time-varying plants in [94]. Both of these approaches result in n^{th} order (order of the generalized plant) pre-filters.

- Control updating between data samples is considered in [18]. They show that such dual-rate time-varying control can be superior to LTI control (more on this below).

The last method is the *only* means by which arbitrary time varying hold functions could be approximated. This is the main drawback of such hold functions. That is, they must be approximated, which opens up a whole new can of worms. The advantage of such hold functions, if they could be realized, is that they make arbitrary *zero* placement possible [1]. However, this can also be accomplished through the use of periodic digital controllers in conjunction with zero order holds [34]. Such controllers can even be used to stabilize decentralized fixed modes [71], if the fixed modes are unstructured. The problem with "optimal" pre-filters is that they will be very high order and therefore candidates for *digital* implementation. Furthermore, they would, in many cases, have to be internal to the sensor. Whether dual rate makes sense, depends on the particular implementation. The only systems for which it makes

sense, are those systems for which the rate at which the data can be sampled is the limiting factor. Let f_y be the sample rate, with respect to the output data. Let f_u be the control input update rate, which corresponds to the rate at which these outputs must be computed. Unless there is a fixed delay in the relationship between y and u, f_y is strictly a function of the sensor technology, whereas f_u is a function of the available computational capacity and the complexity of the relationship between y and u. To date the case of $f_y > f_u$ has not been considered. In [18] it is shown that, when $f_u > f_y$ is achievable, the performance is better than for $f_u = f_y$, for a given f_y. Finally, none of the above consider the quantization (finite word length) problem and we will not either. For a discussion of quantization issues see [27, 58].

2.3.2 Sampled-data repetitive control

Since the only advantage to repetitive control is reduced order controller order, it seems natural that a sampled-data repetitive controller should have a digital repetitive structure. The general structure of digital repetitive controllers is

$$C(z) = \frac{d(z)}{1 - q(z)z^{-L}}, \quad \text{where} \tag{2.8}$$

$q(z)$ and $d(z)$ are rational functions and the integer L is at least twice the ratio of the sampling frequency (rate), $f_s = 1/\tau$, to the fundamental frequency, $f_0 := \omega_0/2\pi$, of the the periodic input. If $q(z) \equiv 1$, then $C(z)$ would cause *discrete-time* plants to perfectly track periodic discrete-time signals with L harmonics and the same sample rate, assuming $d(z)$ renders the system stable. Such a $d(z)$ may or may not exist, for any particular problem. Note that the sampling rate, f_s, must be chosen to be an

integer multiple of the fundamental reference frequency, f_0.

Remark 2.3.1 (digital repetitive control) *Digital repetitive controllers have a purely rational (finite dimensional) structure.*

This is a significant advantage over continuous-time repetitive controllers. The only drawback of continuous-time repetitive controllers is the infinite dimensional structure requiring delays for implementation. The inclusion of the delays significantly increases the complexity of the design process as well as the difficulty of implementation. Thus, digital repetitive controllers have all the advantages, and none of the disadvantages, of repetitive control. The only difficulty lies in how to obtain digital repetitive controllers that satisfy a continuous-time "repetitive" tracking requirement. This is where the sampled-data formulation comes into play.

CHAPTER III

Robust Performance Formulation

In this chapter we pose a robust performance problem for the two degree of freedom repetitive controller of Figure 6, as an H^∞ optimal control problem. We then develop new infinite dimensional H^∞ theory to solve the resulting problem, and present a numerical example.

3.1 Problem Formulation

The sensitivity function of Figure 6, is given by

$$S := \frac{E(s)}{R(S)} := \frac{R(s) - Y(s)}{R(S)} = \frac{1 + PC_2}{1 + P\hat{C}_2}, \quad \text{where} \quad \hat{C}_2 := C_1 + C_2. \tag{3.1}$$

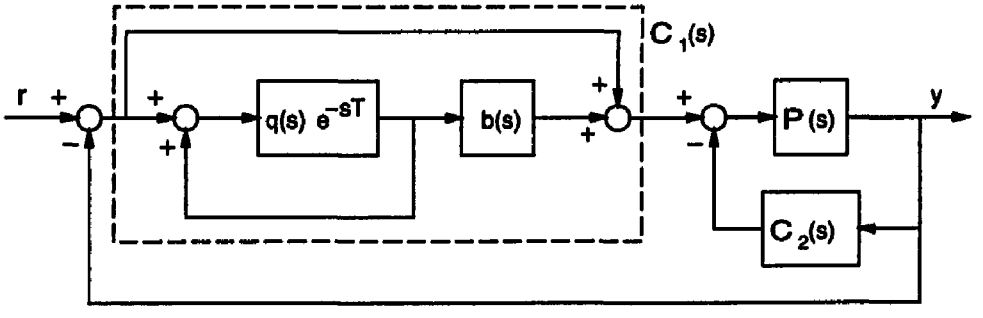


Figure 6: Initial two degree of freedom repetitive controller.

The pair $(C_1, \hat{C}_2)$ corresponds to the standard form of the two degree of freedom control problem [102]. We consider the additive uncertainty description, defined by

$$P_\Delta(s) = P(s) + W_2(s)\Delta(s), \tag{3.2}$$

where P is the nominal plant, P_Δ represents the *unknown* actual plant description, $\Delta \in H^\infty$ is the normalized (such that, $|\Delta(j\omega)| < 1$) uncertainty and W_2 is the additive uncertainty weight. Robust performance implicitly requires robust stability as a prerequisite. The two degree of freedom robust stabilization problem is to find a controller pair $(C_1, \hat{C}_2)$, such that the closed loop system is stable for all possible plants P_Δ. It can be shown, see e.g. [13, 22], that a sufficient condition for robust stability is

$$\|W_2\hat{C}_2(1 + P\hat{C}_2)^{-1}\|_\infty < 1, \tag{3.3}$$

where $\hat{C}_2$ stabilizes the nominal plant, P. In addition to robust stability we want good performance for all P_Δ (robust performance), i.e. we also want $|S_\Delta(j\omega)| \leq |W_1(j\omega)|^{-1}$, for all ω, where

$$S_\Delta := \frac{1 + (P + W_2\Delta)C_2}{1 + (P + W_2\Delta)\hat{C}_2}. \tag{3.4}$$

It can be shown that these definitions lead to the following sufficient condition for robust performance:

$$|W_1|\left(\left|\frac{1 + PC_2}{1 + P\hat{C}_2}\right| + \left|\frac{W_2C_2}{1 + P\hat{C}_2}\right|\right) + \left|\frac{W_2\hat{C}_2}{1 + P\hat{C}_2}\right| < 1, \quad \text{for all } \omega. \tag{3.5}$$

A closely related control problem in terms of the infinity-norm, see [22], is the H^∞ optimal control problem given by

$$\gamma_{opt} = \inf_{C_1, \widehat{C}_2 \text{ stabilizing } P} \left\| \begin{bmatrix} W_1 \left(\frac{1+PC_2}{1+P\widehat{C}_2} \right) \\ W_1 W_2 \left(\frac{C_2}{1+P\widehat{C}_2} \right) \\ W_2 \left(\frac{\widehat{C}_2}{1+P\widehat{C}_2} \right) \end{bmatrix} \right\|_\infty . \tag{3.6}$$

Definition 3.1.1 (robust performance problem) *The continuous-time robust performance problem, considered in this chapter, is defined by (3.6).*

This formulation does not yet have any concrete relationship to repetitive control. Indeed, it is simply an abstract formulation of a generic robust performance problem. We will impose a repetitive structure, by constraining C_1 to have the repetitive structure shown in Figure 6.

Remark 3.1.1 *A characterization of all stabilizing two degree of freedom controllers, where C_1 has the repetitive structure shown in Figure 6, and guidelines on how to choose $q(s)$ for nominal closed loop stability and performance are presented in [39]. Robustness with respect to performance is not addressed.*

In order to solve the robust performance problem, and impose a repetitive structure on C_1, we must bring in a parameterization of all stabilizing controllers. The set of all two degree of freedom controller pairs $(C_1, \widehat{C}_2)$ stabilizing a given nominal plant, P, can be parameterized using a factorization approach [102].

Remark 3.1.2 *While there is only one generic parameterization for all one degree of freedom control systems, see Remark 2.2.1, the parameterization of all stabilizing two degree of freedom controllers is configuration specific.*

We consider the closed loop system shown in Figure 6. Let $X, Y \in \mathrm{IRH}^\infty$ (the set of rational functions in H^∞) be the solutions of the Bezout identity, $NX + DY = 1$, for the nominal plant $P = N/D$, where $N, D \in \mathrm{IRH}^\infty$. Recalling that $\hat{C}_2 := C_1 + C_2$, the set of all two degree of freedom controllers stabilizing P is given by the parameterization

$$C_1 = \frac{Q_1}{Y - NQ_2} \quad \text{and} \quad \hat{C}_2 = \frac{X + DQ_2}{Y - NQ_2}, \tag{3.7}$$

where Q_1 and Q_2 are free parameters in H^∞. In terms of the parameterization of all stabilizing controller pairs, $(C_1, \hat{C}_2)$, the problem (3.6) can be expressed as

$$\gamma_{opt} = \inf_{Q_1, Q_2 \in H^\infty} \left\| \begin{bmatrix} W_1 \\ W_1 W_2 X D \\ W_2 X D \end{bmatrix} - \begin{bmatrix} W_1 N & 0 \\ W_1 W_2 D & -W_1 W_2 D^2 \\ 0 & -W_2 D^2 \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} \right\|_\infty . \tag{3.8}$$

Remark 3.1.3 *If we are only interested in robust stabilization, we can set $W_1 = 0$, which reduces (3.8) to a one block H^∞ problem involving Q_2 only, which is essentially the case when we are designing an initial stabilizing controller.*

Similarly, if we are only interested in nominal performance, we can set $W_2 = 0$, which reduces (3.8) to a one block H^∞ control problem involving Q_1 only, however it is hard to foresee any situation where we wouldn't care about robust stability. This illustrates one advantage of two degree of freedom control: the nominal performance and the robust stabilization problems can be decoupled. However, the advantage for our purposes is that two degree of freedom controllers are completely general and give the maximum possible design flexibility. For robust performance we need to solve (3.8), which couples Q_1 and Q_2. The problem (3.8) is a $MIMO$ two-block H^∞

control problem, which has a well known solution, when W_1, W_2 and P are finite dimensional.

In order to force a repetitive structure on the solution to (3.8), we constrain C_1 to have the repetitive structure of Figure 6, i.e. we set

$$1 + \frac{b(s)q(s)e^{-sT}}{1 - q(s)e^{-sT}} = \frac{Q_1(s)}{Y(s) - N(s)Q_2(s)} \tag{3.9}$$

$$\text{or } (Q_1 - Y + NQ_2) = qe^{-sT}(Q_1 - Y + NQ_2 + b(Y - NQ_2)). \tag{3.10}$$

If $q(s)$ is proper, then the lefthand side of (3.10) must have an equivalent time delay of at least T seconds. Since $Y, N \in \mathrm{IRH}^\infty$ do not contain delays, we let

$$Q_1(s) - Y(s) = e^{-sT}\hat{Q}_1(s) \text{ and} \tag{3.11}$$

$$Q_2(s) = e^{-sT}\hat{Q}_2(s), \tag{3.12}$$

where $\hat{Q}_1, \hat{Q}_2 \in H^\infty$.

Remark 3.1.4 *Given Q_1 and Q_2, there are infinitely many ways to choose corresponding b and q satisfying (3.9). However, if we constrain b and q to be rational, then there exists a unique solution, see Section 3.3. Furthermore, the resulting optimal $\hat{C}_2$ has the alternative repetitive structure shown in Figure 7, and the overall repetitive control design has the generalized repetitive structure of Figure 8.*

Under the decomposition defined by (3.11) and (3.12), the problem (3.8) becomes

$$\gamma_{opt} = \inf_{\hat{Q}_1, \hat{Q}_2 \in H^\infty} \left\| \begin{bmatrix} W_1(1 - NY) \\ W_1W_2(X - Y)D \\ W_2XD \end{bmatrix} - \begin{bmatrix} W_1N & 0 \\ W_1W_2D & -W_1W_2D^2 \\ 0 & -W_2D^2 \end{bmatrix} e^{-sT} \begin{bmatrix} \hat{Q}_1 \\ \hat{Q}_2 \end{bmatrix} \right\|_\infty . \tag{3.13}$$

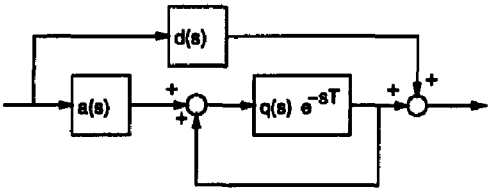


Figure 7: Alternate repetitive controller structure.

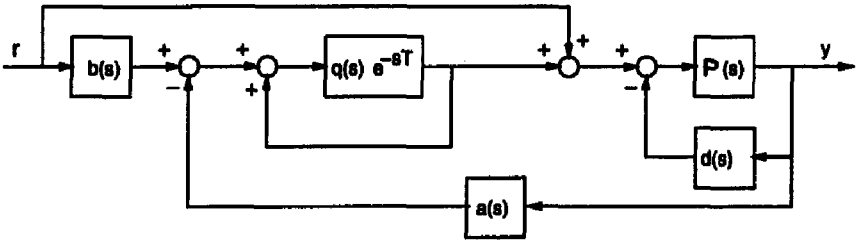


Figure 8: Resulting generalized repetitive controller structure.

Definition 3.1.2 (robust repetitive performance problem) *The robust repetitive performance problem is defined by (3.13) and the solution has the structure shown in Figure 8.*

Explicit formulas for computing the generalized repetitive controller parameters (a, b, d, and q), shown in Figure 8, are given in Section 3.3. The H^∞ optimal control problem given by (3.13) is a vector (SIMO) H^∞ problem with a scalar delay. The solution of this problem requires an extension of the previous results on infinite dimensional H^∞ optimal control, see [28, 31, 69]. In the next section we provide the required extension of infinite dimensional H^∞ optimal control theory for a class of problems including those of the form (3.13).

3.2 Solution of the Infinite Dimensional H^∞ Problem

The infinite dimensional H^∞ optimal control problem (3.13) is of the form

$$\gamma_{opt} = \inf_{\hat{Q}_1, \hat{Q}_2 \in H^\infty} \|F - mG\hat{Q}\|_\infty, \quad \text{where } \hat{Q} := [\ \hat{Q}_1 \quad \hat{Q}_2\ ]^T; \tag{3.14}$$

F and G are arbitrary 3×1 and 3×2 matrices, respectively, with rational entries in H^∞; and $m \in H^\infty$ is an arbitrary inner function. Clearly, (3.13) is a special case of (3.14), where $m(s) = e^{-sT}$. The solution procedure given here is a extension/modification of previously obtained results on the H^∞ optimal control of $SISO$ distributed plants, see [31, 69].

Remark 3.2.1 *A similar vector H^∞ optimal control problem involving time delays has been studied in [28] for an automobile engine idle speed control. They show that the problem can be posed as a singular-value/singular-vector problem for a finite rank infinite dimensional operator, but they do not give explicit formulas for the solution of this problem or present a numerical example.*

We now employ the common techniques of inner/outer factorization and outer factor absorption to obtain a "clean" formulation of the vector H^∞ problem (3.14), see e.g. [22, 33, 102]. Let $G = M_G P_G$ be an inner/outer factorization of G, where M_G is a 3×2 inner matrix and P_G is a 2×2 outer matrix. Define $\Theta := [\ M_G \quad M_G^\perp\ ]$, where $M_G^\perp$ is a 3×1 inner matrix, such that Θ is a 3×3 inner matrix.

Remark 3.2.2 *The outer factor can be absorbed, without loss of generality, by defining $\tilde{Q} := P_G\hat{Q}$, under certain genericity assumptions, see [32].*

Remark 3.2.3 *Thus, to obtain the solution of (3.14) we simply compute* $\hat{Q} = P_G^{-1}\tilde{Q}$, *where if* P_G^{-1} *does not exist, i.e. is improper, the "optimal" performance cannot be achieved, but we can come as close as we like by appropriate approximations to the inverse, see e.g. [22].*

The resulting "clean" (standard) form of the vector H^∞ problem is given by

$$\gamma_{opt} = \inf_{\tilde{Q}_1,\tilde{Q}_2 \in H^\infty} \left\| \begin{bmatrix} R_1 \\ R_2 \\ R_3 \end{bmatrix} - m \begin{bmatrix} \tilde{Q}_1 \\ \tilde{Q}_2 \\ 0 \end{bmatrix} \right\|_\infty , \tag{3.15}$$

where $R := [\ R_1 \quad R_2 \quad R_3\ ]^T := \Theta^* F$. Note that, without loss of generality, we can assume that $R_3 \in \mathbb{R}\mathrm{H}^\infty$.

In order to solve the standard form given by (3.15), we must define a two-block operator, $\mathbf{A}$, in terms of the projection operator, $\mathbf{P}_{H(M)}$, and the multiplication operator, $\mathbf{M}_W$, which are defined in Appendix A.

Definition 3.2.1 *The two-block operator,* $\mathbf{A} : H^2 \to H(M) \oplus H^2$ *(where* $H(M)$ *and* H^2 *are defined in Appendix A), is defined by*

$$\mathbf{A} := \begin{bmatrix} \mathbf{P}_{H(M)} \begin{bmatrix} \mathbf{M}_{V_1} \\ \mathbf{M}_{V_2} \end{bmatrix} \\ \mathbf{M}_{V_3} \end{bmatrix}, \tag{3.16}$$

where m_1 *and* m_2 *are the minimal Blaschke products, such that* $V_1 := m_1 R_1$ *and* $V_2 := m_2 R_2$ *are in* H^∞; $M := m \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix}$; *and* $V_3 := R_3$.

Now, we can state a lemma relating the optimal performance, γ_{opt}, of problem (3.15) to the maximum singular-value of $\mathbf{A}$.

Lemma 3.2.1 *The optimal performance, γ_{opt}, of problem (3.15) is given by $\gamma_{opt} = \|\mathbf{A}\|$, where the operator norm of $\mathbf{A}$ is defined to be the square root of the largest element of the spectrum of $\mathbf{A}$, $\sigma(\mathbf{A})$.*

Lemma 3.2.1 follows directly from the commutant lifting theorem, see e.g. [29, 30, 77]. The spectrum, $\sigma(\mathbf{A})$, consists of two parts: the discrete spectrum $\sigma_d(\mathbf{A})$ consisting of the singular-values with finite multiplicity (the usual singular-values as in the finite dimensional case) and the essential spectrum $\sigma_e(\mathbf{A})$. The essential norm of $\mathbf{A}$, $\|\mathbf{A}\|_e$, is defined to be the square root of the largest element of $\sigma_e(\mathbf{A})$.

Remark 3.2.4 *Under the genericity assumption that the norm of $\mathbf{A}$ is larger than the essential spectrum, see Assumption 1 in Appendix B, $\gamma_{opt} = \|\mathbf{A}\|$ can be uniquely determined by solving the singular-vector/singular-value problem for $\mathbf{A}$.*

Lemma 3.2.2 *Let (γ_{opt}, x^o) be a singular-value/singular-vector pair for the two block operator $\mathbf{A}$, then the optimal interpolant $\tilde{Q}^{opt}$, solving (3.15), is given by*

$$\tilde{Q}_1^{opt} = \frac{\mathbf{P}_+ R_1 m^* x^0}{x^0} \quad \text{and} \quad \tilde{Q}_2^{opt} = \frac{\mathbf{P}_+ R_2 m^* x^0}{x^0}, \tag{3.17}$$

where the projection operator, $\mathbf{P}_+$, is the "stable projection", see Appendix A.

Proof: By the commutant lifting theorem, see e.g. [29, 30, 77], we have

$$\left[\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} - M \begin{bmatrix} \tilde{Q}_1^{opt} \\ \tilde{Q}_2^{opt} \\ V_3 \end{bmatrix} \right] x^0 = \mathbf{A} x^0, \tag{3.18}$$

and from the definition of $\mathbf{A}$, we have

$$\mathbf{A}x^0 = \begin{bmatrix} \mathbf{P}_{H(M)} \begin{bmatrix} \mathbf{M}_{V_1} x^0 \\ \mathbf{M}_{V_2} x^0 \end{bmatrix} \\ \mathbf{M}_{V_3} x^0 \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{H(M)} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \\ V_3 \end{bmatrix} x^0. \tag{3.19}$$

Clearly, (3.18) will be satisfied iff

$$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} x^0 - M \begin{bmatrix} \tilde{Q}_1^{opt} \\ \tilde{Q}_2^{opt} \end{bmatrix} x^0 = \mathbf{P}_{H(M)} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} x^0 = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} x^0 - M\mathbf{P}_+ M^* \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} x^0. \quad (3.20)$$

Canceling common terms and multiplying by M^* from the left, gives

$$\begin{bmatrix} \tilde{Q}_1^{opt} \\ \tilde{Q}_2^{opt} \end{bmatrix} x^0 = \mathbf{P}_+ M^* \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} x^0 = \mathbf{P}_+ \begin{bmatrix} R_1 \\ R_2 \end{bmatrix} m^* x^0. \quad \square \quad (3.21)$$

Thus the solution of (3.15) is equivalent to solving the singular-value/singular-vector problem for **A**.

Remark 3.2.5 *Since m^* has no stable poles, we can write $\tilde{Q}_1^{opt} = \tilde{q}_1/x^0$ and $\tilde{Q}_2^{opt} = \tilde{q}_2/x^0$, where $\tilde{q}_1$ and $\tilde{q}_2$ are stable rational transfer functions.*

Theorem 3.2.1 (solution, operator singular-value/singular-vector problem) *Under certain genericity assumptions, see Assumptions 1-6 in Appendix B, there exist a finite dimensional matrix R_σ, (B.52), such that the pair (σ, x) is a singular-value/singular-vector pair for **A** iff the matrix R_σ is singular. Furthermore, the singular-vector, x, can be explicitly constructed from the interpolation constraints, Φ, defined by $R_\sigma \Phi = 0$.*

The proof, which is extremely long and very mathematical, appears in Appendix B. Appendix B also gives completely explicit equations for constructing R_σ and the singular-vector, x.

Remark 3.2.6 *The maximum singular-value, $\gamma_{opt} := \sigma_{max}$, of **A** can be found by varying the parameter σ over the finite interval $(\|\mathbf{A}\|_e, \|R\|_\infty)$, and finding the largest value of σ for which R_σ is singular.*

The essential norm can be readily determined by means of the following lemma from [30].

Lemma 3.2.3 *The essential norm, $\|\mathbf{A}\|_e$, of $\mathbf{A}$, is given by*

$$\|\mathbf{A}\|_e = max\left\{\|V_3\|_\infty, \|D\|\right\}, \tag{3.22}$$

where (A, B, C, D) is a minimal realization for $\hat{F} := \left[\begin{array}{ccc} V_1^T & V_2^T & V_3^T \end{array}\right]^T$.

Thus the finite interval over which one must search for γ_{opt} is easily determined.

3.3 Calculation of the repetitive controller parameters

In this section we show how to calculate the parameters of the generalized repetitive controller structure shown in Figure 6. From Remark 3.2.5, $\tilde{Q}^{opt}$ can be written as $\tilde{Q}^{opt} = \frac{1}{x^0}[\tilde{q}_1\ \tilde{q}_2]^T$, where $\tilde{q}_1$ and $\tilde{q}_2$ are stable rational functions. It can be shown (see Appendix B) that x^0 can be written as $x^0 = g + mh$, where g and h are rational functions. From Remark 3.2.3 and the fact that all of the entries of P_G^{-1} are rational functions, we can write $\hat{Q}^{opt} = \frac{1}{x^0}[q_1\ q_2]^T$, where q_1 and q_2 are rational functions. Next, we substitute these equations into (3.11) and (3.12), and get

$$Q_1^{opt} = Y + \frac{mq_1}{g + mh} \quad \text{and} \quad Q_2^{opt} = \frac{mq_2}{g + mh}, \quad \text{where} \quad m(s) = e^{-sT}. \tag{3.23}$$

Finally, we substitute (3.23) into (3.7), and use a little algebra, to get

$$1 + \frac{bqm}{1 - qm} = \frac{Q_1}{Y - NQ_2} = 1 + \frac{m(q_1 + q_2N)/gY}{1 - m(q_2N - hY)/gY}, \quad \text{and} \tag{3.24}$$

$$\frac{X + DQ_2}{Y - NQ_2} = \frac{X/Y + m(hX + q_2D)/gY}{1 - m(q_2N - hY)/gY}. \tag{3.25}$$

Thus, by matching coefficients, we have that the unique rational functions $q(s)$ and $b(s)$ are given by

$$q = \frac{q_2 N - hY}{gY} \quad \text{and} \quad b = \frac{q_1 + q_2 N}{q_2 N - hY} \,. \tag{3.26}$$

We set $\hat{C}_2$ equal to the transfer function of the repetitive controller shown in Figure 7, and get

$$\hat{C}_2 = \frac{X + DQ_2}{Y - NQ_2} = \frac{X/Y + m(hX + q_2 D)/gY}{1 - m(q_2 N - hY)/gY} = \frac{d + (a - d)qm}{1 - qm} \,. \tag{3.27}$$

Thus the optimal $\hat{C}_2$ is a repetitive controller of this form, with

$$a = \frac{q_2}{Y(q_2 N - hY)} \quad \text{and} \quad d = \frac{X}{Y} \,. \tag{3.28}$$

Remark 3.3.1 *The rational function $d(s)$ plays the role of the initially stabilizing controller in the traditional repetitive control design procedures.*

Indeed, we can define $X := \tilde{X} + D\tilde{r}$ and $Y := \tilde{Y} - N\tilde{r}$, where $\tilde{r} \in H^\infty$ is a free parameter. This parameterization of all (initially) stabilizing controllers, $d(s)$, can be used to obtain an initial design, $d = X/Y$, with desirable properties.

3.4 Numerical Example

In this section we consider a numerical example using a plant model for an electro-hydraulic material testing machine [56, 80, 79]. Specifically, we consider frequency normalized plants of the form

$$P(s) = \frac{K}{(s - \epsilon)(s + \alpha)(s^2 + 2\zeta s + 1)} \,, \tag{3.29}$$

where the pole at ϵ is an approximation to a pole at the origin.

Remark 3.4.1 *Frequency normalization is necessary for the problem to be numerically well conditioned, i.e. for a numerical solution to be possible using MATLAB. This is a very common difficulty, and is in no way related to the repetitive control formulation.*

In this example $K = 0.2103, \epsilon = .001,\ \alpha = 0.8653,$ and $\zeta = 0.4$. The following $N, D, X,$ and Y satisfy the Bezout identity for $P(s)$:

$$N = \frac{K}{(s+\epsilon)(s+\alpha)(s^2+2\zeta s+1)}, \qquad D = \frac{s-\epsilon}{s+\epsilon}, \tag{3.30}$$

$$X = 0.0082, \text{ and } Y = \frac{s^3+1.6673s^2+1.6956s+0.8687}{(s+\alpha)(s^2+2\zeta s+1)}. \tag{3.31}$$

The initially stabilized closed loop system is given by

$$\frac{P(s)}{1+d(s)P(s)} = NY. \tag{3.32}$$

This initial stabilization step is the first in the two step design process. The second step, which is the subject of this research, solves for the H^∞ optimal $a(s)$, $b(s)$, and $q(s)$.

Let the fundamental period of the signal of interest be $T = 13.7333$, which implies that $\omega_0 = 2\pi/T = 0.4575\ rad/sec$ is the fundamental frequency (first harmonic). Thus the second harmonic is also within the bandwidth of the system.

Remark 3.4.2 *The signal of interest is taken to have only two significant harmonics, because of numerical difficulties associated with large fundamental periods T.*

The specific source of the numerical difficulties associated with large fundamental periods, is discussed later in this section. Let the performance weight, W_1, have a spike at $\omega_1 = 2\omega_0$. Specifically, the weights are given by

$$W_1 = \frac{(s+\alpha)(s^2+2\zeta s+1)}{8s^3+14.2798s^2+7.3216s+11.3681} \quad \text{and} \quad W_2 = \frac{0.5s+0.1}{s+2}. \tag{3.33}$$

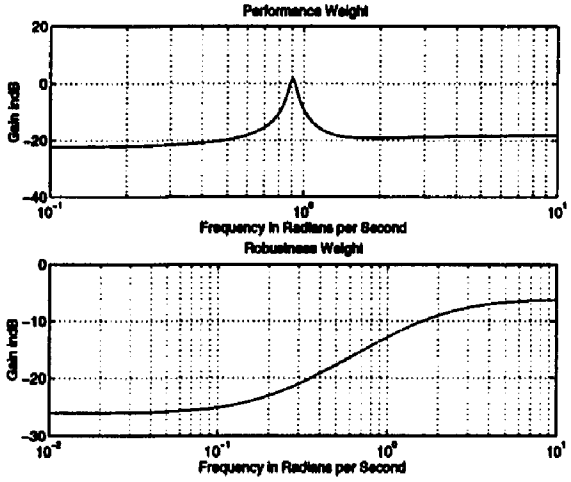


Figure 9: Bode magnitude plots of the weights, W_1 and W_2.

The numerator of W_1 is set equal to the denominator of Y, so that F will not have repeated poles, which would lead to repeated poles for R and violate the genericity assumptions of Theorem 3.2.1, see Assumption 2 in Appendix B. The Bode magnitude plots of the weights are shown in Figure 9.

Remark 3.4.3 *The idea behind choosing a rational weight with a single spike (notch), even when there are more harmonics of interest, is that for properly chosen low order rational weights, the resulting "optimal" controller will still provide repetitive action at several harmonics.*

This approach is employed quite successfully in Chapter IV, which strongly implies that, given better computational facilities, it would work well for this formulation as well.

Since D is inner, F and G defined by (3.13) and (3.14) have a common inner factor, and can be redefined as

$$F := W \begin{bmatrix} 1 - NY \\ X - Y \\ X \end{bmatrix} \quad \text{and} \quad G := W \begin{bmatrix} N & 0 \\ 1 & -D \\ 0 & -D \end{bmatrix}, \quad \text{where}$$

$$W := \begin{bmatrix} W_1 & 0 & 0 \\ 0 & W_1 W_2 & 0 \\ 0 & 0 & W_2 \end{bmatrix}. \tag{3.34}$$

With F and G as defined above, it can be shown that

$$\begin{bmatrix} R_1 \\ R_2 \end{bmatrix} = W_1{}^* W_1 N^* \begin{bmatrix} g_{11} \\ g_{21} \end{bmatrix} - X D^* \begin{bmatrix} p_{12} \\ p_{22} \end{bmatrix} - Y \begin{bmatrix} p_{11} \\ p_{21} \end{bmatrix} \tag{3.35}$$

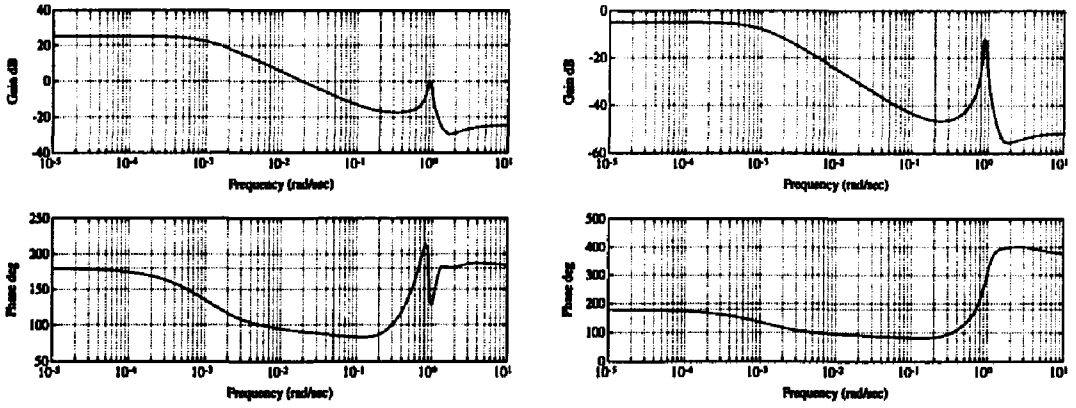
$$\text{and } R_3 = \{ \; F^* F - (R_1{}^* R_1 + R_2{}^* R_2) \; \}^{\frac{1}{2}}, \tag{3.36}$$

$$\text{where } P_G{}^{-*} =: \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \quad \text{and} \quad P_G =: \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}. \tag{3.37}$$

To solve for R we must obtain the outer factor, P_G, of the $MIMO$ transfer function (transfer function matrix) G. Inner/outer factorization of G requires a minimal state space realization. Finding an exact minimal realization for transfer function matrices is a nontrivial problem. Appendix C addresses this issue at some length and gives the details for this example. The resulting minimal realization of G is tenth order.

Remark 3.4.4 *The frequency normalization mentioned above is necessary in order for the realization to be* numerically *controllable and observable.*

The inner/outer factorization of G, using the minimal state space realization, was done using the algorithm in [33]. The resulting common denominator, DR, for R_1 and R_2 is seventeenth order and R_3 is thirteenth order. The Bode plots of the seventeenth



R_1, (17^{th} Order) R_2, (17^{th} Order)

Figure 10: Full order Bode plots of R_1 and R_2, prior to balanced reduction.

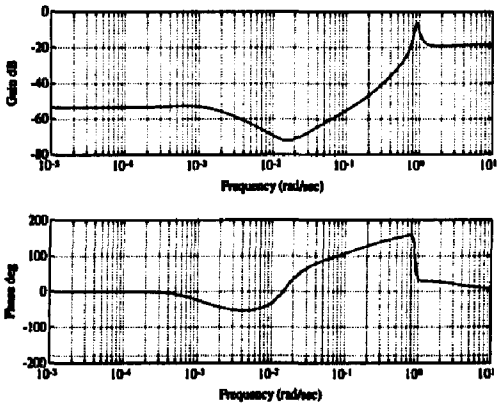
order R_1 and R_2 are shown in Figure 10 and the Bode plot of the thirteenth order R_3 is shown in Figure 11.

Remark 3.4.5 *The high order of these transfer functions makes it numerically impossible to compute the solution to (3.15) using MATLAB, since MATLAB has only sixteen decimal digits of precision and the resulting high order polynomial operations required to construct R_σ, such as root finding and convolution, are numerically ill-conditioned.*

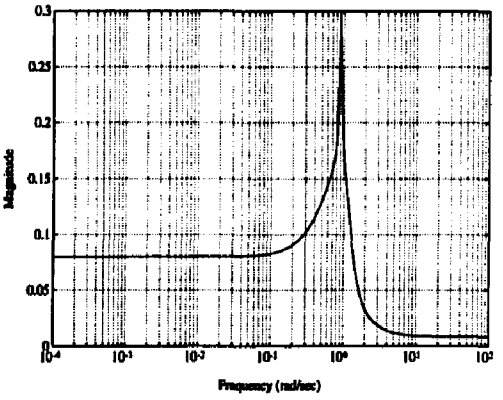
Thus, in order to obtain a numerical solution using MATLAB, the components of R must be approximated by reduced order transfer functions. The singular-value decomposition (SVD) method of [61] for balanced reduction, was used to obtain:

$$\widetilde{R_1} = -0.06206\frac{s^4 + 2.6913s^3 + 4.6800s^2 + 2.0322s + 0.7007}{s^4 + 2.9083s^3 + 0.97887s^2 + 2.3436s + .0023426}\ , \tag{3.38}$$

$$\widetilde{R_2} = 0.002747\frac{s^4 + 0.68439s^3 + 1.0788s^2 + 10.0898s - 0.4793}{s^4 + 2.9083s^3 + 0.97887s^2 + 2.3436s + .0023426}\ , \text{ and} \tag{3.39}$$



R_3, (13^{th} Order)



Approximation Error

Figure 11: Full order Bode plots of R_3 and plot of $\|R - \tilde{R}\|$.

$$\widetilde{R_3} = 0.13368 \frac{s^3 + 1.0548s^2 + 0.4114s + .03597}{s^3 + 2.8808s^2 + 1.0457s + 2.2995} \,. \tag{3.40}$$

The total approximation error, $\|R - \tilde{R}\|$, is shown in Figure 11.

Remark 3.4.6 *The error peak is at the peak of W_1, which is undesirable but unsurprising. However, even the peak error magnitude (0.3) is quite small.*

Remark 3.4.7 *Solving the singular-value/singular-vector problem using $\tilde{R}$ gives $\tilde{\gamma}_{opt}$, where $\gamma_{opt} \leq \tilde{\gamma}_{opt} + \|R - \tilde{R}\|$.*

From Remark 3.2.6 we only have to search over the finite interval $(\|\mathbf{A}\|_e, \|R\|_\infty)$ to find γ_{opt}, where from Lemma 3.2.3 we know that $\|\mathbf{A}\|_e = max\left\{\|\tilde{R}_3\|_\infty, \|D\|\right\}$, where D comes from the state space representation of $\tilde{R}$. With a little calculation, we find that $\|D\| = 0.1474$ and $\|\tilde{R}_3\|_\infty = .5691$, thus $\|\mathbf{A}\|_e = .5691$. Thus, we only need to search over the interval $(0.5691, 18.5717)$, which gives$\tilde{\gamma}_{opt} = 1.296068976743$. All of the digits shown *are significant.* The degree to which $\|\tilde{R} - m\tilde{Q}\|$ is constant for all

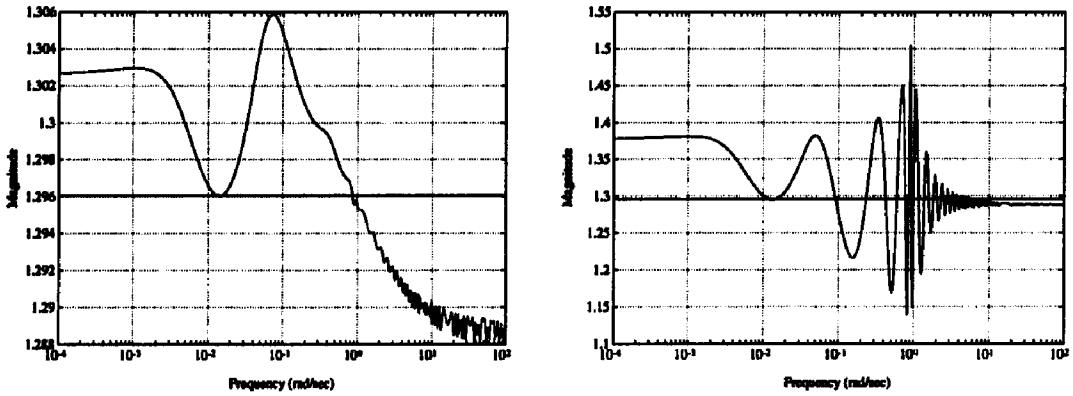
frequencies is very sensitive to the exact value of $\tilde{\gamma}_{opt}$. The corresponding value of the delay, to the same precision, is $T = 13.733333333333$. The resulting performance for the approximate system, $\|\tilde{R} - m\tilde{Q}\|$, and the performance, $\|R - m\tilde{Q}\|$, of the actual system are shown in Figure 12. Ideally, both of the above performance plots would be completely flat.

Remark 3.4.8 *The performance for the approximate system, $\|\tilde{R} - m\tilde{Q}\|$, is quite flat, with a maximum deviation from $\tilde{\gamma}_{opt} = 1.296068976743$ of less than one percent, which confirms the accuracy of the new H^∞ theory developed in this chapter.*

Remark 3.4.9 *The error, in terms of deviation from $\tilde{\gamma}_{opt} = 1.296068976743$, in the plot of $\|R - m\tilde{Q}\|$ is exactly what one would expect, i.e. at both high and low frequencies it is approximately equal to the frequency approximation error, see Figure 11, and the peak error occurs near the peak approximation error and is bounded by the approximation error.*

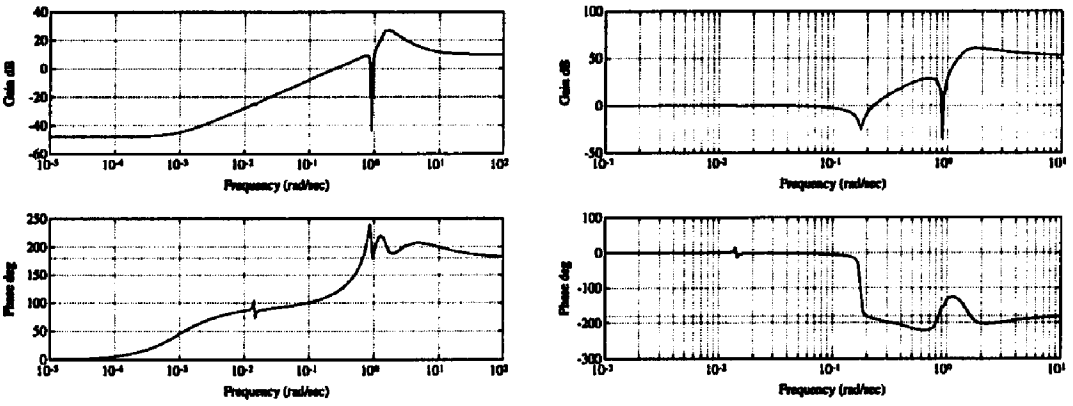
What little error (deviation) there is in the $\|R - m\tilde{Q}\|$ plot, is likely due to numerical difficulties associated with polynomial operations and/or arising from the scaling of some of the entries of R_σ by the $e^{-\gamma_i T}$ (see Appendix B for the definition of the γ_i). Indeed, when the fundamental period, T, was increased even a little bit, the numerics broke down completely and a solution could not be found, i.e. it resulted in an ill-conditioned R_σ, which is *numerically* singular for all sigma. For this example the real part of the γ_i ranges from -2.85 to $+2.85$, which leads to scale factors ranging from 10^{-17} to 10^{16}. Thus, even for this relatively modest value of T, R_σ is poorly

conditioned. This difficulty can only be resolved through increased computational precision (all calculations were done using MATLAB, which only has sixteen decimal digits of precision).



Performance, $\|\tilde{R} - m\tilde{Q}\|$ "True" Performance, $\|R - m\tilde{Q}\|$

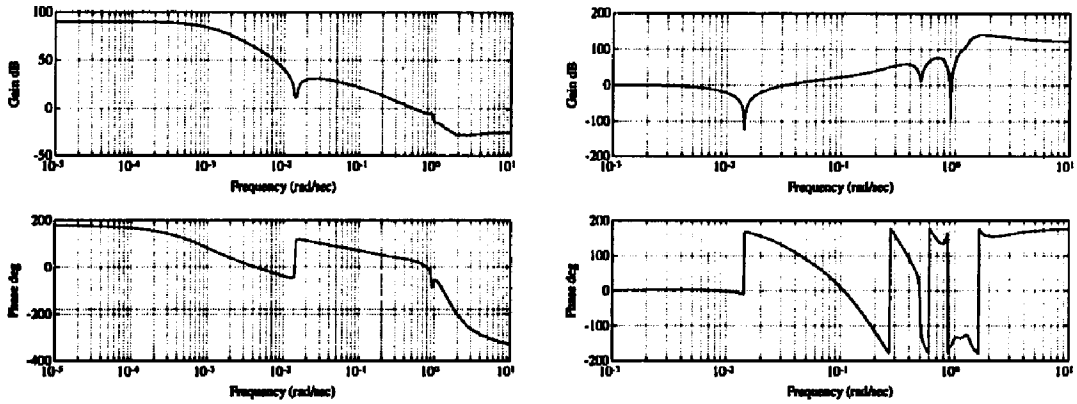
Figure 12: Performance of $\tilde{Q}$ for both the approximation $\tilde{R}$ and the original R.



$a(s)$, (19^{th} Order) $b(s)$, (19^{th} Order)

Figure 13: Bode plots of the repetitive controller parameters $a(s)$ and $b(s)$.

The Bode plots of the repetitive controller transfer functions $a(s)$ and $b(s)$ are shown in Figure 13, and $q(s)$ is shown in Figure 14. While $a(s)$ and $b(s)$ are nineteenth order and $q(s)$ is twentieth order, they could all clearly be well approximated by much



$q(s)$, (20^{th} Order) Sensitivity Function, $S(s)$

Figure 14: Repetitive controller parameter $q(s)$ and resulting performance $S(s)$.

lower order compensators.

Remark 3.4.10 *Ignoring localized behavior, $a(s)$ can be well approximated by a first order transfer function while $b(s)$ and $q(s)$ can be well approximated by second order transfer functions.*

In spite of this encouraging fact, the bottom line is that the performance of this design is awful, see Figure 14. It is believed that this is due to some combination of poor weight selection and numerical errors. However, until the numerical issues can be resolved adequately to allow ready computation for meaningful (multiple significant harmonics) problems, the difficulties and potential of this formulation cannot be evaluated.

Remark 3.4.11 *The excellent results of Chapter IV, for the two-block repetitive control formulation, are reason for considerable optimism about the potential for this repetitive control formulation, pending the resolution of the numerical difficulties.*

At least with presently available computers, e.g. the Sun Sparc Station 5, symbolic/multiprecision packages such as Maple and Mathematica are *not* the answer, because they simply run too slow on problems of this complexity.

Remark 3.4.12 *Clearly, $q(s)$ does not provide classical repetitive action, since it is very large at low frequency. Indeed, the only time that $q(s) \approx 1$ is near the single spike in the performance weight, W_1.*

While this further illustrates the shortcomings of this particular design, the fact that $q(s) \approx 1$ is near the spike in W_1 is quite encouraging with respect to the ultimate potential of the formulation.

Remark 3.4.13 *A very interesting feature of $a(s)$ is that its Bode magnitude is very small in the performance region, effectively opening the outer control loop. while it appears that this is part and parcel of the poor design results, other choices, such as a low pass $a(s)$, may lead to better classical repetitive controller designs.*

In classical repetitive controller design, see e.g. Figure 6, $a(s)/b(s)$ is taken to be identically equal to one, whereas in this design the ratio is practically zero in the performance region. Finally, we should comment on the odd hitch that appears in the Bode plots of both b and q (and hence also in the Bode plot of S) near 0.015 $rad/sec.$ This hitch occurs at one of the γ_i, which are supposed to be canceled by construction, and is believed to be the result of numerical errors.

3.5 Summary and Conclusions

In this chapter we posed an H^∞ optimal robust performance problem, see Definition 3.1.1. By constraining the controller C_1 of Figure 6, we obtain a H^∞ optimal robust repetitive performance problem, see Definition 3.1.2. Under certain genericity assumptions (see Assumptions 1-6 in Appendix B), H^∞ control problems of the more general form (3.15) can be reduced to the calculation of $\tilde{Q}^{opt}$ from (3.17) using the singular-vector, x^0, corresponding to the largest singular-value, $\sigma_{max} = \gamma_{opt}$, of the two block operator **A**, see Definition 3.2.1. The pair (σ, x) is a singular-value/singular-vector pair for **A** iff the matrix R_σ, see (B.52), is singular. To find $\gamma_{opt} = \sigma_{max}$ we need only vary σ over a finite interval, see Remark 3.2.6. The corresponding singular-vector, x^0, can be calculated from the optimal interpolation constraints, Φ^0, defined by $R_{\sigma_{max}}\Phi^0 = 0$, see Appendix B. Finally, $\hat{Q}^{opt}$ can be calculated from x^0 using (3.23).

The H^∞ optimal robust repetitive performance problem of Definition 3.1.2, yields the generalized repetitive controller structure of Figure 8. The unique rational controller parameters of this generalized structure can be calculated directly from $\hat{Q}^{opt}$, see Section 3.3. The parameter $d = X/Y$ is fixed prior to solving the H^∞ optimal robust repetitive performance problem, since it is just the ratio of the two parameters in the solution of the Bezout identity used to parameterize all stabilizing controllers. The only requirement on $d(s)$ is that it must stabilize the nominal plant. Thus, $d(s)$ plays the role of the initially stabilizing controller from classical repetitive control design. So, the overall design procedure has two steps: first find an initially stabilizing

controller $d(s)$, which determines the X and Y used in the H^∞ optimization, and then solve for $q(s)$, $a(s)$ and $b(s)$ by the H^∞ optimization method detailed above.

The numerical example illustrated several interesting effects. The construction of the matrix, R_σ, was found to be numerically poorly conditioned for high order R when using MATLAB. Use of a symbolic program, such as Maple or Mathematica, might reduce this problem, but with currently available computational facilities the resulting run times are excessive. Furthermore, R_σ becomes poorly conditioned for large fundamental periods, T, due to the scaling of some of the entries by $e^{-\gamma_i T}$ (see Appendix B for the definition of the γ_i). This problem is generic in that the γ_i are symmetric with respect to the origin. Thus, as T gets large R_σ will become increasingly poorly conditioned if any of the γ_i have a "large" real part. The only way around this problem is increased machine precision. The form of the resulting controller parameters, $a(s), b(s)$, and $q(s)$, provides insights that may be useful in "classical" repetitive controller design. In particular, the fact that the feedback gain function $a(s) \neq b(s)^{-1}$, suggests that there may be an advantage in introducing $a(s)$ as a new design parameter in classical design. In this case $a(s)$ effectively opens the outer loop at low frequency, which serves to help eliminate repetitive action. However, other choices, such as a low pass $a(s)$, may lead to better classical repetitive designs.

The extension of the existing infinite dimensional H^∞ optimal control theory, required to solve the associated vector H^∞ problem, is of independent theoretical importance. The extension of these results to vector problems of different dimensions is straight forward. The extension to non-scalar inner infinite dimensional factors is

not straight forward, but the mathematical techniques involved should remain essentially the same. For small delays (less than the system time constant) the numerical difficulties with respect to the delay term would be effectively eliminated. However, for repetitive control the delays must always be large (several times the system time constant). While numerical/computational difficulties presently prevent practical repetitive controller design using this formulation, there is considerable promise that, with the ever improving speed and computation power of computers and advances in multi-precision algorithms, it will become possible. Furthermore, in light of the results of Chapter IV, there is considerable reason to believe that the formulation will, at that time, provide usable practical designs.

CHAPTER IV

Nominal Performance With Robust Stability

In this chapter we consider the "two-block" (nominal performance with robust stability) formulation of the continuous-time H^∞ optimal repetitive control problem. The primary advantage of the two-block formulation is that when the performance weight, $W_1(s)$, and the robustness weight, $W_2(s)$, are properly chosen the numerical difficulties associated with H^∞ optimal repetitive control are significantly reduced. The drawback is that we sacrifice specific guarantees with respect to maintaining the desired performance in the face of plant variations.

4.1 Initial Stabilization and Approximate Inversion

In repetitive control design the nominal plant, $P_n(s)$, is generally assumed to have been stabilized by an initial stabilizing controller, $d(s)$. The stabilized plant, $P(s)$, is then typically approximately inverted, over the performance region, by the compensator , $b(s)$, see Figure 15. Note that this formulation corresponds to the two degree of freedom control problem, where one degree of freedom is fixed by the selection of the initial stabilizing controller, $d(s)$, see Chapter III. This formulation leads to the equivalent approximate system shown in Figure 16. In this section we prove a lemma giving sufficient conditions for stabilization of the system in Figure 15 using

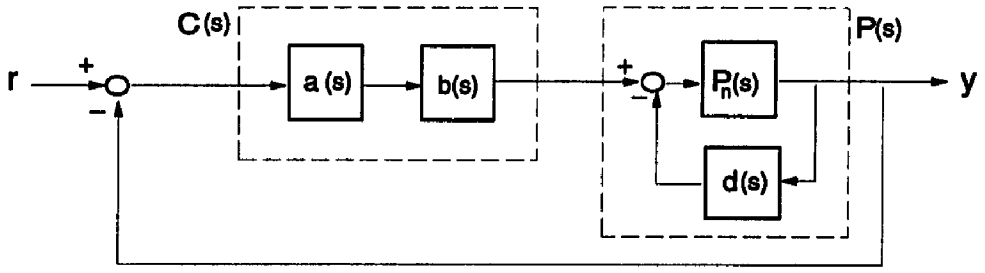


Figure 15: System structure, nominal performance with robust stability problem.

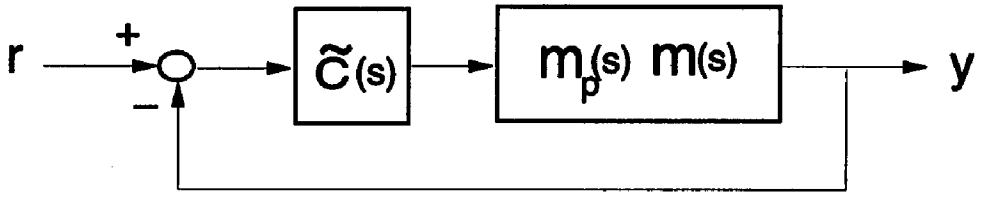


Figure 16: Simplified structure assuming perfect inversion of the stabilized plant.

controllers, $\tilde{C}(s)$, designed for the equivalent approximate system shown in Figure 16. We also give a simple formula for bounding the reduction in high frequency robustness due to the approximate nature of the plant inversion.

It is well known, see e.g. [22], that the system of Figure 15 is internally stable iff the four transfer functions: $S := (1 + CP)^{-1}$,PS, CS, and $T := PCS$ are all stable. While the controller $d(s)$ ensures that $P(s)$ is stable, it cannot ensure that it is minimum phase. Thus we introduce the factorization, $P(s) = m_p(s)P_o(s)$, where $P_o(s)$ is outer (minimum phase) and $m_p(s)$ is inner (all pass). In order to obtain controllers with a repetitive structure, see Section 4.3, we introduce inner factor, $m(s) := e^{-sT}$, which can be thought of as having been factored out of $a(s)$, i.e. $a(s) := m(s)\tilde{C}(s)$. In order to ensure the stability of the resulting design (by ensuring the stability of b, see Lemma 4.1.1), we only attempt to invert the outer factor, $P_o(s)$,

of the stabilized plant, $P(s)$. The approximate system, assuming that $b(s)P(s) = b(s)P_o(s)m_p(s) = m_p(s)$, is shown in Figure 16. It should be noted that even though we are just attempting to invert the outer part of a stable plant, the inversion is necessarily approximate irrespective of plant uncertainty. The inversion must be only approximate because all physical plants, $P_n(s)$, are strictly proper, i.e. they have zero response at infinite frequency. Furthermore, stabilization of the strictly proper plant, $P_n(s)$, by any realizable (proper) initial controller, $d(s)$, results in a strictly proper stable plant, $P(s)$. Thus, to exactly invert the plant would require an improper (unrealizable) compensator, $b(s)$. However, for all practical purposes it is sufficient that $b(s)$ approximately invert the plant over a finite frequency band (including the "performance band"), see [96] and Lemma 4.1.1 below.

Since the equivalent plant, $m_p(s)m(s)$, is stable, the parameterization of all stabilizing controllers is given by

$$\tilde{C}(s) = \frac{Q(s)}{1 - m_p(s)m(s)Q(s)}, \tag{4.1}$$

where $Q(s) \in H^\infty$ is a free parameter. Recall that classical repetitive controllers have unity feedback in the repetitive block, see Figure 3 and (2.1). Thus the repetitive structure for $a(s) := m(s)\tilde{C}(s)$, shown in Figure 17, is a modified repetitive structure for $m_p \neq 1$. Note that for $Q(s)$ unity-low pass (i.e. $Q(j\omega) \approx 1$ at "low" frequencies) and $m(s) \equiv 1$, the controller $a(s)$ has a classical repetitive structure. Indeed, the sensitivity function of the system shown in Figure 16 is $\tilde{S}(s) = 1 - m_p(s)m(s)Q(s)$ and the corresponding complementary sensitivity function is $\tilde{T}(s) = m_p(s)m(s)Q(s)$.

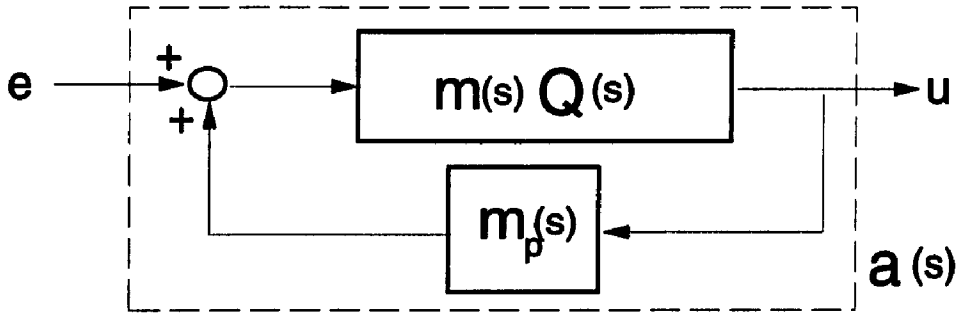


Figure 17: Modified repetitive structure of $a(s)$.

Remark 4.1.1 *The fact that the modified repetitive structure (see Figure 17) for non-minimum phase plants arises naturally in the parameterization of all stabilizing controllers, implies that it is a superior structure with respect to obtaining stable repetitive control systems.*

Whether or not the modified structure is superior with respect to obtaining good repetitive performance is not immediately obvious, but it does warrant further research.

Using the parameterization of all stabilizing controllers, we can state a lemma giving sufficient conditions for system stability under approximate inversion. Note that accurate inversion can only be obtained where the plant uncertainty is low. Thus, where plant uncertainty is high (i.e. in the robustness region) the plant cannot be accurately inverted.

Lemma 4.1.1 (Approximate Inversion) *If d stabilizes P_n, $\tilde{C}$ stabilizes $m_p m$, b is stable, and $\|(bP_o - 1)Q\|_\infty < 1$, then the system shown in Figure 15 is internally stable.*

Proof: Recall from (4.1), that the parameterization of all $\tilde{C}$ that stabilize $m_p m$, in terms of the free parameter $Q \in H^\infty$, is given by

$$\tilde{C}(s) = \frac{Q(s)}{1 - m_p(s)m(s)Q(s)}. \tag{4.2}$$

Using the fact that $a(s) := m(s)\tilde{C}(s)$, we have that the four transfer functions that characterize internal stability are:

$$S(s) = \frac{1 - m_p(s)m(s)Q(s)}{1 + (b(s)P_o(s) - 1)m_p(s)m(s)Q(s)}, \tag{4.3}$$

$$C(s)S(s) = \frac{m(s)Q(s)b(s)}{1 + (b(s)P_o(s) - 1)m_p(s)m(s)Q(s)}, \tag{4.4}$$

$$P(s)S(s) = \frac{m_p(s)P_o(s)(1 - m_p(s)m(s)Q(s))}{1 + (b(s)P_o(s) - 1)m_p(s)m(s)Q(s)}, \quad \text{and} \tag{4.5}$$

$$T(s) = \frac{m_p(s)m(s)b(s)P_o(s)Q(s)}{1 + (b(s)P_o(s) - 1)m_p(s)m(s)Q(s)}. \tag{4.6}$$

Since $\tilde{C}(s)$ stabilizes $m_p(s)m(s)$, we know that $Q(s) \in H^\infty$, so the numerator of $S(s)$ is stable. Furthermore, since $b(s)$ and $P_o(s)$ are both stable, all of the numerators are stable. Finally, the common denominator will be stable if

$$\|(bP_o - 1)m_p m Q\|_\infty = \|(bP_o - 1)Q\|_\infty < 1. \quad \square \tag{4.7}$$

Note that while the conditions given in the Lemma 4.1.1 are conservative, they are not very restrictive nor difficult to satisfy. Clearly, the approximate inversion must be highly accurate wherever $|Q(j\omega)|$ is relatively large (e.g. the performance region). Conversely, where $|Q(j\omega)|$ is small (e.g. the small gain region) the approximation can be very course. Indeed, the product $b(s)P_o(s)$ can be allowed to go to zero in this region with very little adverse effect. Recall that for "good" designs, the small gain region (frequencies where the loop gain is less than one for all possible plant variations)

is semi-infinite. Thus the only practical impact of under-inversion ($|b(j\omega)P_o(j\omega)| < 1$) in this region, is a reduction in the high frequency robustness bound. High frequency robustness is normally quantified via the complementary sensitivity function, $T(s)$, introduced above. The following lemma gives an upper bound on the reduction in robustness due to under-inversion ($|b(j\omega)P_o(j\omega)| = \beta < 1$).

Definition 4.1.1 (robustness reduction factor) *The robustness reduction factor, $\delta(T,\omega)$, is defined to be the ratio of the absolute value of the actual $T(j\omega)$ to the absolute value obtained with exact inversion ($b(s)P_o(s) \equiv 1$), i.e.*

$$\delta(T,\omega) = \frac{b(j\omega)P(j\omega)}{1 + (b(j\omega)P(j\omega) - 1)m_p(j\omega)m(j\omega)Q(j\omega)}. \tag{4.8}$$

Lemma 4.1.2 (robustness reduction bound)

If $|Q(j\omega)| < \alpha < \frac{1}{2}$ *and* $|b(j\omega)P_o(j\omega)| = \beta < 1$, *then* $\delta(T,\omega) \leq \frac{\beta}{1-(\beta+1)\alpha}$.

Proof: A worst case bound for a rational function is the maximum possible numerator divided by the minimum possible denominator. Since, $m_p(s)$ and $m(s)$ are inner, $|b(j\omega)P_o(j\omega)m_p(j\omega)m(j\omega)Q(j\omega)| = \beta|Q(j\omega)|$. Similarly, we have the inequality $|1+(b(j\omega)P_o(j\omega)-1)m_p(j\omega)m(j\omega)Q(j\omega)| \geq |1-(b(j\omega)P_o(j\omega)-1)\alpha| \geq 1-(\beta+1)\alpha$. Thus, we have

$$|T(j\omega)| = \left|\frac{m_p(j\omega)m(j\omega)b(j\omega)P_o(j\omega)Q(j\omega)}{1 + (b(j\omega)P_o(j\omega) - 1)m_p(j\omega)m(j\omega)Q(j\omega)}\right| \leq \frac{\alpha\beta}{1-(\beta+1)\alpha}. \tag{4.9}$$

Using the fact that for $b(j\omega)P_o(j\omega) \equiv 1$, we have $|T(j\omega)| = |Q(j\omega)|$, which gives the desired result. □

Note that the worst case for the bound ($\beta = 1$) corresponds to the perfect inversion case. Thus the bound is conservative. Furthermore, since the magnitude of $P_o(j\omega)$ goes to zero, as ω goes to infinity, and the magnitude of the proper rational transfer function $b(j\omega)$ is finite for all $\omega \in \mathbb{R}$, the actual robustness is significantly improved at very high frequencies. That is, the true $T(j\omega)$ goes to zero, as ω goes to infinity, while the "ideal" (perfect inversion) $T(j\omega)$ does not. Finally, a simple example shows that even the worst case bound is very small for $\alpha << 1$.

Example: Let $|Q(j\omega)| < \alpha_t = 0.1$ and $0 \leq |b(j\omega)P_o(j\omega)| = \beta < 1$ (under-inversion) for all $\omega \in [\omega_t, \infty)$, then the worst case (conservative) bound on the reduction in robustness is $\delta(T, \omega) \leq 1/0.8$ or $1.9382dB$. □

4.2 Solution of the Two-Block Problem

In [95], Toker and Özbay characterize all optimal and suboptimal solutions to the two-block (nominal performance with robust stability) H^∞ control problem for a large class of infinite dimensional systems. This class of systems includes systems of the form shown in Figure 16. In this section we present a specialized version of their result, which is specific to the optimal solution of this smaller class of problems.

The two block (nominal performance with robust stability) problem is formulated in terms of a performance weighting function, $W_1(s)$, the sensitivity function, $S(s)$, a robustness weighting function, $W_2(s)$, and the complementary sensitivity function, $T(s)$. The H^∞ optimal solution to this problem is defined to be the minimization,

over all stabilizing controllers, of the performance measure γ defined by

$$\gamma := \left\| \begin{bmatrix} W_1 S \\ W_2 T \end{bmatrix} \right\|_\infty . \tag{4.10}$$

Let γ_o denote the optimal solution (minimum value of γ), and make the following preliminary definitions

$$E_\gamma(s) := \left(\frac{W_1(-s)W_1(s)}{\gamma^2} - 1 \right) \text{ and } F_\gamma(s) := G_\gamma(s) \prod_{k=1}^{n_1} \frac{(\eta_k - s)}{(\eta_k + s)}, \text{ where } \tag{4.11}$$

the η_k, $k = 1...n_1$, are the poles of $W_1(-s)$ and the outer factor $G_\gamma(s)$ is defined by

$$G_\gamma(-s)G_\gamma(s) := \left(1 - \left(\frac{W_2(-s)W_2(s)}{\gamma^2} - 1 \right) E_\gamma(s) \right)^{-1} . \tag{4.12}$$

Theorem 4.2.1 (Toker and Özbay [95]) *Let the weights $W_1(s)$ and $W_2(s)$ be rational functions. Then the H^∞ optimal solution to the two block problem defined by (4.10), for the system defined in Figure 16, is given by*

$$\hat{C}_o(s) = \frac{E_{\gamma_o}(s)F_{\gamma_o}(s)L(s)}{1 + m_p(s)m(s)F_{\gamma_o}(s)L(s)}, \tag{4.13}$$

where $L(s) := L_2(s)/L_1(s)$, which satisfies $|L(j\omega)| \equiv 1$, is the ratio of two polynomials of degree less than or equal to $(n_1 - 1)$. The coefficients of these polynomials constitute $2n_1$ unknowns, which can be uniquely computed from the following set of $2n_1$ linearly independent equations:

$$0 = L_1(\beta_k) + m_p(\beta_k)m(\beta_k)F_{\gamma_o}(\beta_k)L_2(\beta_k), \tag{4.14}$$

$$0 = L_2(-\beta_k) + m_p(\beta_k)m(\beta_k)F_{\gamma_o}(\beta_k)L_1(-\beta_k), \tag{4.15}$$

where the zeros of E_{γ_o} (β_k, $k = 1 \ldots n_1$) are ordered such that $\beta_{n_1+i} = -\beta_i$, $i = 1 \ldots n_1$ and the first n_1 zeros are in the closed right half plane (RHP).

Remark 4.2.1 *While the solution parameter $L(s)$ satisfies $|L(j\omega)| \equiv 1$, it is not, in general, an inner function, since it is not, necessarily, stable.*

Replacing γ_o with γ, the system of equations in the coefficients of $L_1(s)$ and $L_2(s)$ defines a matrix function of γ that must be singular when $\gamma = \gamma_o$. Thus, γ_o may be found as in the robust performance problem formulation of Chapter III, i.e. by searching over a finite interval for the largest value of gamma that makes this matrix singular. Onur Toker has written a user friendly software package in MATLAB that implements this theory for the more general case developed in [95].

4.3 Sensitivity Improvement Formulation

As seen in Chapter III, an important issue in H^∞ optimal repetitive controller design is how to ensure that the resulting controllers actually provide repetitive action. Recall that repetitive performance is obtained when the magnitude of the sensitivity function is "small" at the fundamental frequency, $\omega_o = \frac{2\pi}{T}$, and at a number of harmonics, $\omega = k\omega_o$. Repetitive action occurs when repetitive performance is achieved as a direct result of the time delay term, $m(s) := e^{-sT}$. For the system shown in Figure 16, repetitive action is achieved precisely when $Q(jk\omega_o) \approx 1$, for some set of integers k. One approach, that ensures repetitive action of the resulting H^∞ optimal controller, is choosing the performance weight, W_1, to have an infinite dimensional repetitive structure. The problem with infinite dimensional weights is that in general the resulting H^∞ optimal control problem remains unsolved. In particular, Theorem 4.2.1 requires that the weighting functions $W_1(s)$ and $W_2(s)$ be rational. The

sensitivity improvement formulation incorporates an infinite dimensional repetitive performance weight, $W_1(s)$, such that the resulting two-block H^∞ problem reduces to the solution of a two-block H^∞ problem that can be solved by Theorem 4.2.1.

The sensitivity improvement problem is formulated in terms of a nominal repetitive control design. The nominal design may be a pre-existing design or it may be the first part in a two part repetitive control design. The nominal design must have the structure shown in Figure 17. Thus, it may not be possible to use an existing design for non-minimum phase plants, since this is a novel structure for such plants.

Definition 4.3.1 (nominal repetitive design) *The nominal repetitive design for the sensitivity improvement problem is given by Figure 17, with* $Q(s) = q_n(s)$.

Typically, $q_n(s) \in H^\infty$ will be a low order rational transfer function that is unity-low pass, i.e. $q_n(j\omega) \approx 1$ for $\omega \in [0, k\omega_o]$. The sensitivity and complementary sensitivity of the nominal design are given by $S_n(s) = 1 - m_p(s)m(s)q_n(s)$ and $T_n(s) = m_p(s)m(s)q_n(s)$, respectively. The sensitivity improvement problem is formulated by defining the performance weight, $W_1(s)$, in terms of the nominal classical repetitive sensitivity function, $S_n(s)$. This directly constrains the solution to provide repetitive performance.

Definition 4.3.2 (performance weight for sensitivity improvement) *The infinite dimensional repetitive performance weight for the sensitivity improvement problem is given by*

$$W_1(s) := \frac{\widehat{W}_1(s)}{S_n(s)} = \frac{\widehat{W}_1(s)}{1 - m_p(s)m(s)q_n(s)}, \tag{4.16}$$

where $\widehat{W}_1(s)$ is a rational transfer function.

Thus the resulting overall sensitivity function, $S(s)$, will tend to have the nominal sensitivity function, $S_n(s)$, as a factor. The solution of the resulting infinite dimensional two-block H^∞ problem is the subject of the following subsection.

4.3.1 Solution of the sensitivity improvement problem

The solution to the sensitivity improvement problem also has the modified repetitive structure shown in Figure 17, but this time the free parameter $q \in H^\infty$ is infinite dimensional.

Definition 4.3.3 (sensitivity improvement problem) *The solution of the sensitivity improvement problem, defined by (4.16), is given by the repetitive structure of Figure 17, with $Q(s) = q(s)$.*

Thus the overall sensitivity and complementary sensitivity functions are given by $S(s) = 1 - m_p(s)m(s)q(s)$ and $T(s) = m_p(s)m(s)q(s)$.

The sensitivity improvement problem, in its present form, cannot be solved by Theorem 4.2.1, since the performance weight,

$$W_1(s) := \frac{\widehat{W}_1(s)}{S_n(s)} = \frac{\widehat{W}_1(s)}{1 - m_p(s)m(s)q_n(s)}, \tag{4.17}$$

is infinite dimensional. To convert the problem to an equivalent problem that we can solve, we note that

$$W_1(s)S(s) = \widehat{W}_1(s)(1 - m_p(s)m(s)\hat{q}(s)) =: \widehat{W}_1(s)\hat{S}(s). \tag{4.18}$$

Therefore, we have

$$W_1(s)S(s) = \widehat{W}_1(s)(1 - m_p(s)m(s)\hat{q}(s)), \tag{4.19}$$

where $\hat{q} \in H^\infty$ is a free parameter given by

$$\hat{q}(s) =: \frac{q(s) - q_n(s)}{1 - m_p(s)m(s)q_n(s)}, \tag{4.20}$$

or $q(s) := q_n(s) + \{1 - m_p(s)m(s)q_n(s)\}\hat{q}(s)$. Hence, we have

$$W_1(s)S(s) = \widehat{W}_1(s)(1 - m_p(s)m(s)\hat{q}(s)) =: \widehat{W}_1(s)\hat{S}(s). \tag{4.21}$$

The definition of $\hat{S}(s)$ is natural since, letting $Q(s) = \hat{q}(s)$ in Figure 17, it is precisely the sensitivity function corresponding to $\hat{q}(s)$. Note that $1 - m_p(s)m(s)\hat{q}(s) \in H^\infty \Rightarrow q \in H^\infty$ for all q_n, $\hat{q} \in H^\infty$. Thus, whenever $Q(s) = \hat{q}(s)$ stabilizes the system of Figure 16, then so does $Q(s) = q(s)$, assuming that the nominal repetitive design is stable.

Remark 4.3.1 *The sensitivity improvement design is a perturbation of the original design in the sense that for $\hat{q}(s) = 0$, we recover the nominal design, i.e. we have $q(s) = q_n(s)$.*

Since $\widehat{W}_1(s)$ is a rational function, the H^∞ optimal solution to the sensitivity improvement problem can be obtained from Theorem 4.2.1 using an appropriate rational robustness weight $\widehat{W}_2(s)$. The corresponding complementary sensitivity function, $\hat{T}(s)$, associated with the free parameter $\hat{q}(s)$ is $\hat{T}(s) := m_p(s)m(s)\hat{q}(s)$. Thus the optimal solution, $\hat{C}_o(s)$, of Theorem 4.2.1 using $\widehat{W}_1(s), \widehat{W}_2(s), \hat{S}(s)$, and $\hat{T}(s)$ in place

of $W_1(s), W_2(s), S(s)$, and $T(s)$ corresponds to the optimal $\hat{q}(s)$. Letting $\hat{q}(s)$ be the optimal solution, and solving for $\hat{q}(s)$ in terms of $\hat{C}_o(s)$, we have

$$\hat{q}(s) = \frac{\hat{C}_o(s)}{1 + m_p(s)m(s)\hat{C}_o(s)} = \frac{E_\gamma(s)F_\gamma(s)L(s)}{1 + m_p(s)m(s)F_\gamma(s)L(s)\{E_\gamma(s) + 1\}}, \qquad (4.22)$$

where γ_o has been replaced by γ for convenience. Now, we can define the optimal solution to the sensitivity improvement problem.

Definition 4.3.4 (optimal solution of the sensitivity improvement problem) *The H^∞ optimal solution to the sensitivity improvement problem, defined by (4.16), is given by $q(s) := q_n(s) + (1 - m_p(s)m(s)q_n(s))\hat{q}(s)$, where $\hat{q}(s)$ is given by (4.22).*

The resulting overall sensitivity function, $S(s)$, is given by

$$S(s) = 1 - m_p(s)m(s)q_n(s) - m_p(s)m(s)\{1 - m_p(s)m(s)q_n(s)\}\hat{q}(s) \quad \text{or}$$

$$S(s) = \{1 - m_p(s)m(s)q_n(s)\}\{1 - m_p(s)m(s)\hat{q}(s)\} = S_n(s)\hat{S}(s). \qquad (4.23)$$

Thus the overall sensitivity is the product of the nominal sensitivity, $S_n(s)$, and the "sensitivity improvement" term , $\hat{S}(s)$, that has a repetitive structure, where the quotation marks are to emphasize the fact that the sensitivity is only improved at those frequencies where the sensitivity improvement term is less than one. Taking the product of two transfer functions is equivalent to cascading the two corresponding systems. Thus the solution to the sensitivity improvement problem can be thought of as the cascade of the nominal repetitive design, as defined by $q_n(s)$, and the "improvement" design, as defined by $\hat{q}(s)$.

Definition 4.3.5 (cascade repetitive control) *The repetitive controller, and controller structure, resulting from the optimal solution of the sensitivity improvement problem will be referred to as the cascade repetitive controller, and the cascade repetitive control structure, respectively.*

See Subsection 4.3.2 for more on the resulting cascade structure and Section 4.4 for more on cascade repetitive control.

The resulting overall complementary sensitivity function, $T(s)$, for the cascade repetitive design, is given by

$$T(s) = m_p(s)m(s)q_n(s) + m_p(s)m(s)\left\{1 - m_p(s)m(s)q_n(s)\right\}\widehat{q}(s) \quad \text{or}$$

$$T(s) = T_n(s) + S_n(s)\widehat{T}(s). \tag{4.24}$$

Note that the overall complementary sensitivity function is a (weighted) sum rather than a product of the individual complementary sensitivity functions. This means that instead of having an improvement, as for the sensitivity, we have a (hopefully modest) decrease in overall robustness, as measured by the overall complementary sensitivity function. Wherever $|T_n(j\omega)|$ is small, $|S_n(j\omega)| = |1 - T_n(j\omega)| \approx 1$, thus the scaling of $\widehat{T}(s)$ by $S_n(s)$ has little effect on the overall robustness. While the overall robustness (roll-off in T) is less than that of the nominal design (roll-off in T_n), the reduction can typically be made insignificant through the proper selection of $\widehat{W}_2(s)$. Note that $|T_n(j\omega)| = |m_p(j\omega)m(j\omega)q_n(j\omega)| = |q_n(j\omega)|$, and similarly $|\widehat{T}(j\omega)| = |\widehat{q}(j\omega)|$. Thus, robustness issues can be studied by directly looking at q_n and $\widehat{q}$. Note that the high frequency behavior of $\widehat{q}(j\omega)$ depends directly on the

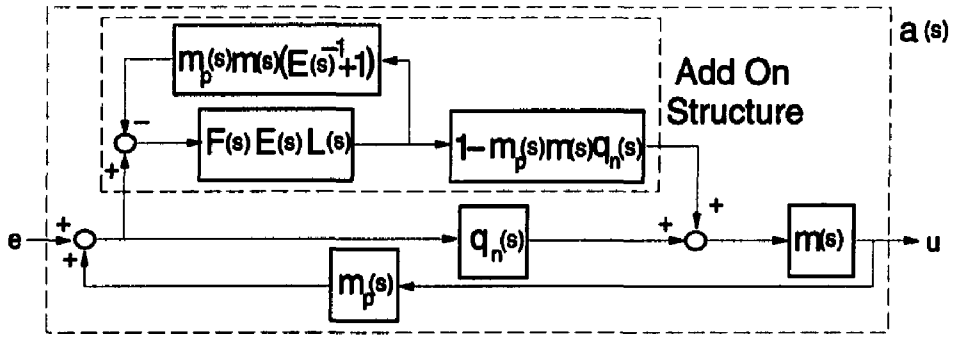


Figure 18: Structure of the H^∞ optimal cascade repetitive controller.

high frequency behavior of $\widehat{W}_2(j\omega)$. The goal is to find choices of $\widehat{W}_1(s)$ and $\widehat{W}_2(s)$ that will lead to enhanced repetitive performance in the frequency band(s) of interest, while preventing any significant reduction in the overall robustness. For more on weight selection for repetitive control see Section 4.6.

4.3.2 Resulting controller structure

The resulting H^∞ optimal controller has an add-on structure, see Figure 18. Recall that by "add-on structure" we mean that the "new" part of the controller, $\{1 - m_p(s)m(s)q_n(s)\}\,\widehat{q}(s)$, can be connected in parallel with some portion of the existing controller, in this case $q_n(s)$. The new structure generally requires a total of three delays to implement (the delay from the nominal design plus two additional delays), but it can provide significant sensitivity improvement while maintaining adequate robustness. Furthermore, through judicious weight selection, see Section 4.6, a reduced order approximation with a simplified structure having only two delays can be obtained, see Subsection 4.3.3.

To see how we arrive at the structure shown in Figure 18, recall the definition of $\hat{q}$ (4.22) in terms of $\hat{C}_o$, from Theorem 4.2.1,

$$\hat{q}(s) = \frac{\hat{C}_o(s)}{1 + m_p(s)m(s)\hat{C}_o(s)} = \frac{E_\gamma(s)F_\gamma(s)L(s)}{1 + m_p(s)m(s)F_\gamma(s)L(s)\{E_\gamma(s) + 1\}}. \tag{4.25}$$

Recalling that $q(s) := q_n(s) + \{1 - m_p(s)m(s)q_n(s)\}\hat{q}(s)$ and the definition of $\tilde{C}(s)$, we have

$$a(s) := m(s)\tilde{C}(s) = \frac{m(s)q(s)}{1 - m_p(s)m(s)q(s)}, \tag{4.26}$$

which yields the structure shown in Figure 18 for the overall combined (cascade repetitive) solution to the sensitivity improvement problem.

Remark 4.3.2 *While the "cascade repetitive" design of Figure 18 is a cascade design, in the sense that the overall sensitivity in the product of the individual sensitivities of the nominal design (q_n) and the improvement design ($\hat{q}$), it will only be a truly cascade repetitive design if $\hat{q}$ provides repetitive action.*

Recall that the sensitivity function can be factored as

$$S(s) = \{1 - m_p(s)m(s)q_n(s)\}\{1 - m_p(s)m(s)\hat{q}(s)\} = S_n(s)\hat{S}(s). \tag{4.27}$$

It is clear from this factorization that the maximum sensitivity improvement, as measured at the harmonics, is obtained precisely when $\hat{S}(s) := 1 - m_p(s)m(s)\hat{q}(s)$ provides repetitive action. That is, we want $\hat{S}(j\omega_0 k) \approx 0$ for some set of integers, k.

Remark 4.3.3 (impossibility of perfect performance) *We cannot ever achieve $\hat{S}(s) := 1 - m_p(s)m(s)\hat{q}(s) \equiv 0$, since $\hat{q} \in H^\infty$ must be stable and causal.*

In order to get $\hat{S}(j\omega_0 k) \approx 0$, we need $\hat{q}(j\omega_0 k) \approx 1/m(j\omega_0 k)m_p(j\omega_0 k)$. Thus, ideally we would like $\hat{q} \in H^\infty$ to invert both $m_p(j\omega_0 k)$ and $m(j\omega_0 k)$ for the required set of integers k. We make the following observation with respect to achievable performance (improvement) via $\hat{q}$:

- Exact inversion of $m_p(s)$ by $\hat{q}(s)$ is impossible, since all the zeros of $m_p(s)$ are in the right half plane (RHP) and $\hat{q}(s)$ must be stable.
- Exact inversion of $m(s)$ by $\hat{q}(s)$ is impossible, since $m(s) := e^{-sT}$ is a pure time delay and $\hat{q}(s)$ must be causal.
- Approximate inversion of $m(j\omega_0 k)$ is not difficult, since $m(j\omega_0 k) = e^{j2\pi k} \equiv 1$.
- Approximate inversion of $m_p(j\omega_0 k)$, on the other hand, presents a significant problem, since the phase of $m_p(j\omega_0 k)$ is different for every k.

This explains why there will be significant reductions in achievable performance for certain non-minimum phase systems, i.e. for those systems where there is significant variation in the phase of $m_p(j\omega)$ over the performance region. The selection of the performance and robustness weights, $\widehat{W}_1$ and $\widehat{W}_2$, such that $\hat{q}$ that provides repetitive action are discussed in Section 4.6.

Remark 4.3.4 *In general, $\hat{q}(s)$ is an irrational function, see (4.25), but if $\hat{q}(s)$ could be approximated by a low order rational function, then the solution of the sensitivity improvement problem would be the stable cascade of two classical repetitive controllers.*

The conditions under which $\hat{q}(s)$ can be approximated by a low order rational function are discussed in Subsection 4.3.3 and selection of appropriate weights is discussed in Section 4.6.

In order to better understand the controller structure and the issues involved in obtaining reduced order/complexity approximations, we must take a detailed look at the rational components, F_γ, E_γ, and L. Irrespective of cancellations that might occur due to the particular weight selections, there are structural cancellations that occur in the forming $F_\gamma(s)$, $E_\gamma(s)F_\gamma(s)$, and $(E_\gamma(s)^{-1}+1)$. To see where the cancellations occur, we must take a closer look at the definitions of these terms. In so doing we will also be able to calculate an upper bound on the combined structural order of the rational components of the infinite dimensional cascade repetitive controller $a(s)$.

Let $\widehat{W}_1(s) =: N_{W_1}(s)/D_{W_1}(s)$, where the polynomial $D_{W_1}(s)$ is taken to be monic. Similarly, let $\widehat{W}_2(s) =: N_{W_2}(s)/dW_2(s)$. Finally, let $\widehat{N}_{W_1}(s) := N_{W_1}(s)/\gamma$ and $\widehat{N}_{W_2}(s) := N_{W_2}(s)/\gamma$. Now, by definition, see Section 4.2, we have

$$E_\gamma(s) = \frac{\widehat{N}_{W_1}(-s)\widehat{N}_{W_1}(s) - D_{W_1}(-s)D_{W_1}(s)}{D_{W_1}(-s)D_{W_1}(s)} \quad \text{and} \tag{4.28}$$

$$\prod_{k=1}^{n_1} \frac{(\eta_k - s)}{(\eta_k + s)} = (-1)^{n_1} \frac{D_{W_1}(-s)}{D_{W_1}(s)}. \tag{4.29}$$

Similarly, for $G_\gamma(-s)G_\gamma(s)$ we have

$$G_\gamma(-s)G_\gamma(s) = \frac{D_{W_1}(-s)D_{W_1}(s)dW_2(-s)dW_2(s)}{dG_\gamma(-s)dG_\gamma(s)} =: \frac{nG_\gamma(-s)nG_\gamma(s)}{dG_\gamma(-s)dG_\gamma(s)}. \tag{4.30}$$

Thus, by inspection, we have $nG_\gamma(s) = D_{W_1}(s)dW_2(s)$. From the definition of $F_\gamma(s)$, after cancellations we have

$$F_\gamma(s) = G_\gamma(s) \prod_{k=1}^{n_1} \frac{(\eta_k - s)}{(\eta_k + s)} = (-1)^{n_1} \frac{D_{W_1}(-s)dW_2(s)}{dG_\gamma(s)}. \tag{4.31}$$

Similarly, after cancellations we have

$$F_\gamma(s)E_\gamma(s) = (-1)^{n_1}\frac{\left(\widehat{N}_{W_1}(-s)\widehat{N}_{W_1}(s) - D_{W_1}(-s)D_{W_1}(s)\right)dW_2(s)}{D_{W_1}(s)dG_\gamma(s)}. \qquad (4.32)$$

Now, we can readily determine the structural upper bound on the order of $a(s)$. From (4.30) we see that $ord(G_\gamma) = ord(W_1) + ord(W_2)$, from (4.31) we see that $ord(F_\gamma) = ord(G_\gamma)$, from (4.32) we see that $ord(F_\gamma E_\gamma) = 2ord(W_1) + ord(W_2)$, and by definition $ord(L) \leq ord(W_1) - 1$. Thus the upper bound on the structural order of the rational components of $a(s)$ is

$$ord(a) \leq 5ord\left(\widehat{W}_1\right) - 1 + ord\left(\widehat{W}_2\right) + 2ord(q_n) + 3ord(m_p). \qquad (4.33)$$

Remark 4.3.5 *The structural upper bound on the total order of the rational components of $a(s)$ is not of any practical importance, since for properly selected weighting functions the resulting controllers can be well approximated by much lower order rational parts within a simplified cascade repetitive structure, see Section 4.6.*

While the existence of right half plane (RHP) plant zeros ($m_p(s) \neq 1$) does not significantly impact the structural upper bound on the total order of the optimal controller, the relative impact is magnified in the reduced order case. Furthermore, the presence of RHP plant zeros reduces the achievable repetitive performance, often significantly.

4.3.3 Reduced order control with a simplified structure

In this subsection we explore ways to obtain rational approximations for $\widehat{q}(s)$ and the conditions under which it can be done. Recall from (4.25), that in terms of the

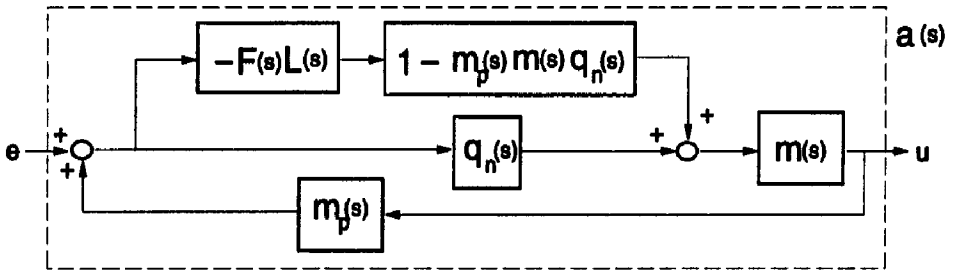


Figure 19: Simplified cascade repetitive structure ($E_\gamma(j\omega) \approx -\hat{E}_\gamma \approx -1$).

solution parameters of Theorem 4.2.1, we have

$$\hat{q}(s) = \frac{E_\gamma(s)F_\gamma(s)L(s)}{1 + m_p(s)m(s)F_\gamma(s)L(s)\{E_\gamma(s)+1\}}. \tag{4.34}$$

By inspection, the *only* way that we can obtain a nonzero rational approximation for $\hat{q}(s)$ is if we can replace $E_\gamma(s)$ with -1. Recalling that

$$E_\gamma := \left(\frac{\widehat{W}_1(-s)\widehat{W}_1(s)}{\gamma^2} - 1\right), \tag{4.35}$$

we see that $E_\gamma(s) \equiv -1$ iff $\widehat{W}_1(s) \equiv 0$. Clearly, we cannot have a performance weight $\widehat{W}_1(s) \equiv 0$ for a sensitivity improvement problem, or for any meaningful problem formulation. Thus the approximation must consist of two parts, i.e. we need $E_\gamma(j\omega) \approx -\hat{E}_\gamma$ outside of a "small" frequency band, and we also require that the constant, $\hat{E}_\gamma < 1$, be approximately equal to one.

Remark 4.3.6 *To obtain a rational approximation for $\hat{q}(s)$, and the corresponding simplified cascade repetitive structure of Figure 19, we must be able to make the double approximation $E_\gamma(j\omega) \approx -\hat{E}_\gamma \approx -1$.*

Remark 4.3.7 *Whether $E_\gamma(j\omega) \approx -\hat{E}_\gamma$, depends entirely on the weighting function, $\widehat{W}_1$, while whether $\hat{E}_\gamma \approx 1$, also depends strongly on the robustness weight, $\widehat{W}_2$, through its effect on the value of γ.*

The selection of weighting functions that achieve these objectives is discussed in Section 4.6. The goal at present is to explore the implications of making the double approximation and resulting simplifications in the cascade repetitive controller structure.

Remark 4.3.8 *Under the double approximation, $E_\gamma(j\omega) \approx -\hat{E}_\gamma \approx -1$, we have $\widehat{q}(s) \approx -F_\gamma(s)L(s) =: \check{q}(s)$, which gives $\hat{S} = 1 - m_p m\check{q} =: \check{S}$ and $\hat{T} = m_p m\check{q} =: \check{T}$.*

The resulting simplified cascade repetitive controller structure is shown in Figure 19. The desirability of this simplified structure, as compared to the original structure of Figure 18, is quite obvious. Not only has an entire infinite dimensional block (including a delay term, m) been eliminated but there has been a significant reduction in the total order of the rational components of $a(s)$.

Remark 4.3.9 *The simplified cascade repetitive controller will result in a stable closed loop system iff $\check{q}(s) := -F_\gamma(s)L(s)$ is stable, i.e. iff $\check{q} \in H^\infty$.*

The transfer function $F_\gamma(s)$ is stable by construction, see (4.31) in Subsection 4.3.2. However, there is no guarantee that $L(s)$ will be stable, see Theorem 4.2.1 and Remark 4.2.1. Thus if we wish to make the double approximation, $E_\gamma(j\omega) \approx -\hat{E}_\gamma \approx -1$, we must be certain that the corresponding $L(s)$ is stable. (Strictly speaking, we only require that $F_\gamma(s)L(s)$ is stable, which could (theoretically) happen when $L(s)$ is

unstable, since $F_\gamma(s)$ is non-minimum phase.) Conversely, since this approximation leads to instability whenever $L(s)$ is unstable, one would expect that for any robust design for which $E_\gamma(j\omega) \approx -\hat{E}_\gamma \approx -1$, that the corresponding $L(s)$ would indeed be stable. This is exactly what happens in practice, see Section 4.7.

Remark 4.3.10 *If $\hat{q}(s)$ provides repetitive action and $E_\gamma(j\omega) \approx -\hat{E}_\gamma \approx -1$, then the simplified reduced order cascade repetitive controller provides classical repetitive action, i.e. $\check{q}(j\omega) \approx 1$ in the performance region.*

This, rather intuitive, observation is illustrated by the numerical examples in Section 4.7.

Whether the sensitivity properties are maintained depends on whether the corresponding cancellations in $F_\gamma(s)$ are made. Recall from Subsection 4.3.2, that there are structural cancellations in taking the product $F_\gamma(s)E_\gamma(s)$. Thus, making the approximation $E_\gamma(j\omega) \approx -\hat{E}_\gamma \approx -1$, without making the corresponding cancellations in $F_\gamma(s)$, causes significant distortion in the frequency band where E_γ is *not* well approximated by a constant (i.e. near ω_p, see Section 4.7).

Remark 4.3.11 *The simplified cascade repetitive structure, of Figure 19, can be thought of as a framework for robustly cascading two classical repetitive controllers. Indeed, the two unity-low pass filters, $q_n(s)$ and $\check{q}(s)$, in this controller structure, can be designed using any method so long as they are both stable and that the combined robustness, see (4.24), is adequate.*

Recall that the cascade repetitive controller parameter, $\hat{q} \in H^\infty$, is the free parameter corresponding to the optimal solution of the two-block problem of Theorem 4.2.1

(with $W_1 = \widehat{W}_1$ and $W_2 = \widehat{W}_2$). Thus the approximation $E_\gamma(j\omega) \approx -\hat{E}_\gamma \approx -1$, also has implications for the solution of this problem irrespective of any relationship with the sensitivity improvement problem. From Theorem 4.2.1, we have

$$\hat{C}(s) = \frac{E_\gamma(s)F_\gamma(s)L(s)}{1 + m_p(s)m(s)F_\gamma(s)L(s)} \approx \frac{-F_\gamma(s)L(s)}{1 + m_p(s)m(s)F_\gamma(s)L(s)}. \tag{4.36}$$

Letting $\tilde{C}(s) = \hat{C}(s)$ in Figure 16 and recalling that $\check{q} := -F_\gamma L$, we have

$$a(s) := m(s)\tilde{C}(s) = \frac{m(s)\check{q}(s)}{1 - m_p(s)m(s)\check{q}(s)}, \tag{4.37}$$

which has a repetitive structure. Specifically, it has the modified repetitive structure of Figure 17, with $Q(s) = \check{q}(s)$. Hence, there is a certain equivalence between "good" solutions to the sensitivity improvement problem and the direct design of H^∞ optimal repetitive controllers.

4.4 Direct H^∞ Optimal Repetitive Controller Design

As we saw in the previous section, there is a certain equivalence between obtaining repetitive sensitivity improvement (cascade repetitive control) and direct H^∞ optimal repetitive control. In this section we explore this equivalence and exploit it to directly obtain H^∞ optimal repetitive controllers for any rational nominal plant $P_n(s)$. While the method is applicable to non-minimum phase plants, it is not always possible to achieve broad band repetitive action for non-minimum phase plants, see Section 4.7. The resulting controllers do not have an add-on structure, but do have a simple repetitive structure with one delay term, $m(s) := e^{-sT}$, and at most three rational transfer functions, see Figure 20.

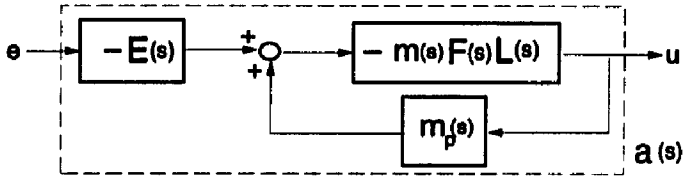


Figure 20: Controller structure for direct H^∞ optimal repetitive control.

Definition 4.4.1 (direct H^∞ optimal repetitive control problem) *The direct H^∞ optimal repetitive control problem is defined by setting $\tilde{C}(s) = \hat{C}(s)$ in Figure 15, where $\hat{C}(s)$ is as defined in Theorem 4.2.1.*

Thus the sensitivity of the direct design is $\hat{S} := 1 - m_p m\hat{q}$, which is precisely the sensitivity improvement of the cascade design, see (4.27). The direct H^∞ optimal repetitive control problem is directly related to the sensitivity improvement part of the overall sensitivity improvement (cascade repetitive) problem.

Remark 4.4.1 (equivalence to the sensitivity improvement problem) *The sensitivity function of the direct design and the sensitivity improvement term of the cascade design are both given by $\hat{S} := 1 - m_p m\hat{q}$, see (4.27). Furthermore, the complementary sensitivity function, $\hat{T} := m_p m\hat{q}$, of the direct design is the "robustness reduction" term for sensitivity improvement design, see (4.24).*

In other words, any given choice of weights and the corresponding solution to Theorem 4.2.1 has two distinct, but essentially equivalent, interpretations. Thus, we can study the achievable performance for both formulations simultaneously and without distinction.

Recalling that $\check{q}(s) := -F_\gamma(s)L(s)$ and the definition of $\hat{C}(s)$ from Theorem 4.2.1, we have

$$a(s) := m(s)\tilde{C}(s) = m(s)\hat{C}(s) = -E_\gamma(s)\frac{m(s)\check{q}(s)}{1 - m_p(s)m(s)\check{q}(s)}, \tag{4.38}$$

which has a simple repetitive structure. The unique property of this H^∞ optimal repetitive controller is that $\check{q}(s)$ is non-minimum phase. The rational filter block $-E_\gamma(s)$ is, strictly speaking, both unstable and non-minimum phase, however under the approximation $E_\gamma(j\omega) \approx -\hat{E}_\gamma \approx -1$, we recover the simple repetitive structure of Figure 17 with $Q(s) = \check{q}(s)$.

Remark 4.4.2 *In general $\check{q}$ may actually be unstable (due to the possible instability of L), however in practice, whenever $E_\gamma(j\omega) \approx -\hat{E}_\gamma \approx -1$, the corresponding L will be stable, ensuring the stability of $\check{q}$ (see Remark 4.3.9).*

Actually, $L(s)$ typically turns out to be stable even when $E_\gamma(j\omega) \approx -\hat{E}_\gamma = -\rho$, where $0 < \rho << 1$. To see why this is the case we examine the gain margin for the direct H^∞ optimal design problem, under the approximation $E_\gamma(j\omega) \approx -\hat{E}_\gamma = -\rho$.

Remark 4.4.3 *Strictly speaking, $\check{q}(s)$ is unstable iff $L(s)$ has an unstable pole that is distinct from all of the poles of $\widehat{W}_1(-s)$, i.e. all of the η_k, $k = 1 \ldots n_1$, see Theorem 4.2.1. However, in practice the poles of $L(s)$ are distinct from the η_k with probability one.*

Thus, in order to be excruciatingly correct, we state the lemma in terms of the technical conditions on the instability of $\check{q}$.

Lemma 4.4.1 (Gain margin for $E_\gamma(j\omega) \approx -\hat{E}_\gamma = -\rho,\ 0 < \rho \leq 1$) *Whenever $L(s)$ has an unstable pole distinct from all of the η_k, $k = 1 \ldots n_1$, ρ^{-1} is the positive gain margin of the direct design, i.e. the system is unstable for $b(s)P(s) = m_p(s)/\rho$.*

Proof: The free parameter $\hat{q} \in H^\infty$ (4.22) parameterizes all stabilizing controllers, i.e. the closed loop system is stable iff $\hat{q}(s) \in H^\infty$. Considering Figure 16, it is clear that if $b(s)P(s) = \delta m_p(s)$, for some $\delta > 0$, then we must absorb δ into $\tilde{C}$. Setting $E_\gamma(s) = -\rho$ and absorbing δ into $\hat{C}(s)$, we have

$$\hat{C}(s) = \frac{-\rho\delta F_\gamma(s)L(s)}{1 + m_p(s)m(s)F_\gamma(s)L(s)}. \tag{4.39}$$

Solving for $\hat{q}(s)$ from

$$\hat{q}(s) = \frac{\hat{C}(s)}{1 + m_p(s)m(s)\hat{C}(s)}, \tag{4.40}$$

we have

$$\hat{q}(s) = \frac{-\rho\delta F_\gamma(s)L(s)}{1 + m_p(s)m(s)F_\gamma(s)L(s)\,(1 - \rho\delta)}. \tag{4.41}$$

Clearly, for $\delta = 1/\rho$ we have $\hat{q}(s) = -F_\gamma(s)L(s)$, which is not in H^∞ under the conditions stated. □

Thus we see that for any design with any robustness at all, whenever $E_\gamma(j\omega) \approx -\hat{E}_\gamma \approx -1$ the corresponding $L(s)$ will indeed be stable.

Remark 4.4.4 *When the sensitivity improvement problem is taken to be a two part design procedure (i.e. $q_n(s)$ is not part of a pre-existing design), the nominal design can be taken to be the solution of the direct H^∞ optimal repetitive control design problem under the approximation $E_\gamma(j\omega) \approx -\hat{E}_\gamma \approx -1$, by letting $q_n(s) = \check{q}(s)$.*

4.5 Two Step Repetitive Controller Design Procedure

The design methodology used in this chapter is a multi-step methodology which includes, in general, the design of an initial stabilizing controller, $d(s)$, an approximate inversion filter (compensator), $b(s)$, and the design of a robustly stable H^∞ optimal repetitive controller. In the sensitivity improvement problem there may be an additional step to design the initial repetitive controller, unless the nominal design already physically exists. In this section we: take a closer look at the methodology, look at the interaction between the design of the various elements, and break the methodology down into two primary steps that are largely independent.

The design procedure consists of the following two steps:

1. Design the initial stabilizing controller, $d(s)$, and the approximate inversion compensator, $b(s)$.

2. Design the H^∞ optimal controller for the equivalent plant, $m_p(s)m(s)$, using Theorem 4.2.1 (for either a direct or a cascade design).

The first step includes all of the finite dimensional filter (compensator) design, while the second step includes the infinite dimensional design that obtains the repetitive performance. It should be noted that there is nothing unusual about the controller approximately inverting the plant over the performance region. Indeed, approximate inversion in the performance region is inherent to optimal control and is the basis of the "Dynamic Inversion" design methodology [10, 26], which extends the concepts and techniques of H^∞ and μ-syntesis to systems with nontrivial nonlinearities.

Furthermore, the two steps are essentially independent, with the only connection being the requirement that $\|(bP-1)Q\|_\infty < 1$ (see Lemma 4.1.1) to ensure stability under approximate inversion, where $Q(s) = \hat{q}(s)$ for the direct design and $Q(s) = q(s) := q_n(s) + \{1 - m_p(s)m(s)q_n(s)\}\,\hat{q}(s)$ for the cascade design. The finite dimensional problems involved in Step 1 are not the subject of this chapter, but a few comments are in order about Step 1 and the interaction with Step 2.

It may be desirable to complete step 2 first so that one knows how accurately the plant must be approximately inverted, and over what region this accuracy is required. On the other hand, the relative difficulty of achieving accurate inversion over a given region should be taken into consideration when designing $Q(s)$. The design of the initial stabilizing controller, $d(s)$, and the approximate inversion compensator, $b(s)$, is highly coupled, i.e. the choice of $d(s)$ has a dramatic impact on the ease with which the resulting stable plant $P(s)$ can be approximately inverted over the performance region. Obviously, $d(s)$ and $b(s)$ should be chosen to minimize their combined order, subject to the constraint of obtaining adequate inversion over the performance region.

Remark 4.5.1 *If at all possible, the initial stabilizing controller, $d(s)$, should be chosen so as not to introduce right half plane (RHP) zeros in $P(s)$, i.e. $d(s)$ should be stable. The problem of finding stable controllers is known as the strong stabilization problem, see [22].*

Recall, see [22], that strong stabilization is stabilization by means of a stable controller. It should be emphasized that this principle holds true whenever one is using initial stabilization to be followed by *any* control technique, since non-minimum phase

plants are universally troublesome with respect to control design.

4.6 Weight Selection

In general one selects the performance weight, $\widehat{W}_1(s)$, to be large at frequencies where good performance is required (typically "low" frequencies) and the robustness weight, $\widehat{W}_2(s)$, to be large at frequencies where the plant uncertainty is large (typically "high" frequencies). Note that when one is considering the sensitivity improvement problem, the "performance" weight may be better described as the "performance improvement" weight. In this section we address the special issues involved in selecting rational weighting functions that result in controllers that provide repetitive performance.

In order to obtain repetitive action, it is important that the robustness weight be essentially flat over the performance region. To understand why this is important recall that for repetitive action to occur we must have nearly identical phase and gain characteristics at a number of discrete frequencies uniformly distributed over the performance region. In particular, we require $\check{q}(j\omega_o k) \approx 1,\ k = 1 \ldots n$ for both the direct and cascade design problems, see Remark 4.4.1 in Section 4.4. The choice of the performance weight is a little more subtle. While the choice of a constant performance weight would be desirable with respect to having perfect phase properties over the performance region, the resulting H^∞ problem is ill-posed. The next closest thing, with respect to constant phase, is to place a very sharp spike at one of the harmonics ($k\omega_o$). This "key" harmonic should be chosen to be an especially important "high" frequency harmonic. The generic weight structures for obtaining repetitive

performance are

$$\widehat{W}_1(s) = \frac{s^2 + 2\zeta_n\omega_p s + \omega_p^2}{s^2 + 2\zeta_d\omega_p s + \omega_p^2} \text{ and } \widehat{W}_2(s) = K_2\frac{s + a_2}{s + b_2}. \tag{4.42}$$

In order for the resulting design to provide repetitive behavior, it is necessary that the effects, including phase, of the single spike in $\widehat{W}_1(s)$ be localized to just the single harmonic. This is accomplished by selecting ζ_n sufficiently small. Note that given $\zeta_n > \zeta_d$ sufficiently small, we have $\widehat{W}_1(j\omega) \approx 1$, outside of a small frequency band centered around $\omega = \omega_p$. Thus, we have

$$E_\gamma(j\omega) := \left(\frac{\widehat{W}_1(-j\omega)\widehat{W}_1(j\omega)}{\gamma^2} - 1\right) \approx -\hat{E}_\gamma, \tag{4.43}$$

which implies that $E_\gamma(j\omega) \approx -\hat{E}_\gamma$ is automatically satisfied whenever $\widehat{W}_1$ is selected to obtain repetitive performance. The spike height, which is proportional to ζ_n/ζ_d, must be very large before the resulting controller will provide repetitive action, see Section 4.7. Thus, ζ_d must be selected to be sufficiently small compared to ζ_n so that adequate spike height is obtained. Turning our attention to the robustness weight, a_2 determines the boundary (design trade-off) between the performance region and the uncertainty region and $b_2 > a_2$ is chosen such that sufficient high frequency robustness is obtained. This leaves K_2 as the only remaining free parameter. Thus, K_2 is selected such that $\hat{E}_\gamma \approx 1$. This must be done indirectly through the impact of K_2 on the design measure γ.

Remark 4.6.1 *For any $\widehat{W}_1(s)$ defined by (4.42), we have $\hat{E}_\gamma = 1 - \gamma^{-2}$.*

Substituting the generic form of $\widehat{W}_1(s)$ into the definition of $E_\gamma(s)$, we see that

$$E_\gamma(s) = \frac{s^4(\gamma^{-2} - 1) + \cdots + \omega_p^4(\gamma^{-2} - 1)}{s^4 + \cdots + \omega_p^4}. \tag{4.44}$$

Thus, $E_\gamma(0) = E_\gamma(\infty) = (\gamma^{-2} - 1) = -\hat{E}_\gamma$, so that $E_\gamma(j\omega) \approx -\hat{E}_\gamma \approx -1$, precisely when $\gamma^{-2} \approx 0$, i.e. when $\gamma >> 1$. Consequently, it follows that K_2 should be chosen such that γ is large.

Remark 4.6.2 *When $E_\gamma(j\omega) \approx -\hat{E}_\gamma \approx -1$, we get $F_\gamma(s) \approx \gamma\{\widehat{W}_2(s)\}^{-1}$. Furthermore, a reduced order F_γ of this form must be used whenever $E_\gamma(s)$ is replaced by -1, to prevent significant reduction in performance at the "key" harmonic ω_p.*

The general form of the reduced order approximation for F_γ follows directly from substituting $E_\gamma \approx -1$ into the definition of F_γ, see Theorem 4.2.1. In particular, with $W_2(s)$ replaced by $\widehat{W}_2(s)$, we have

$$G_\gamma(-s)G_\gamma(s) := \left(1 - \left(\frac{\widehat{W}_2(-s)\widehat{W}_2(s)}{\gamma^2} - 1\right) E_\gamma\right)^{-1} \approx \frac{\gamma^2}{\widehat{W}_2(-s)\widehat{W}_2(s)}. \qquad (4.45)$$

Substituting this expression into the definition of F_γ, we have

$$F_\gamma(s) := G_\gamma(s) \prod_{k=1}^{n_1} \frac{(\eta_k - s)}{(\eta_k + s)} \approx \frac{\gamma}{\widehat{W}_2(s)} \prod_{k=1}^{n_1} \frac{(\eta_k - s)}{(\eta_k + s)} \approx \gamma\{\widehat{W}_2(s)\}^{-1}, \qquad (4.46)$$

since the inner function defined by the η_k has highly localized phase properties. In practice there are, typically minor, perturbations to the zero of $\widehat{W}_2(s)$, as it becomes a pole of $F_\gamma(s)$, and corresponding perturbations to the gain constant γ/K_2, see Section 4.7.

Remark 4.6.3 *The resulting reduced order repetitive controller parameter, $\check{q}$, is a second order, stable, non-minimum phase, rational transfer function providing classical repetitive action, i.e. $\check{q} \approx 1$ in the performance region.*

An interesting feature of $\check{q}$ is that the resulting performance is "centered" around ω_p. That is, $\check{S}(j\omega) := 1 - m_p(j\omega)m(j\omega)\check{q}(j\omega)$ is smallest (provides the best performance) at $\omega = \omega_p$ and increases (provides decreased performance), almost symmetrically, at the higher and lower harmonics (at least for "nearby" harmonics). This symmetry property is a natural consequence of the symmetry in the definition of $\widehat{W}_1$. Similarly, the symmetry of the performance breaks down as we move away from ω_p, since the effects of the robustness weight become more and more asymmetrical.

Remark 4.6.4 *While $m_p(s)$ significantly impacts the achievable broad band repetitive performance, see Remark 4.3.3, it does not significantly impact the outer factor, F_γ, of the repetitive controller parameter, $\check{q} := -F_\gamma L$, see Section 4.7.*

The direct explanation for this phenomenon is that m_p has little effect on γ. There are two parts to the explanation of why m_p has little effect on γ. First, $\widehat{W}_2$ is selected such that $\widehat{E}_\gamma \approx 1 \Rightarrow \gamma >> 1$, which tends to dominate any impact of m_p on γ. Second, we are really only demanding good performance at the single harmonic, ω_p, and $m_p(j\omega_p)$ is just a simple phase shift that is easily inverted by the inner part (L) of the non-minimum phase repetitive controller parameter, $\check{q} := -F_\gamma L$. While L can easily invert m_p at a single frequency, it cannot invert m_p completely since $\check{q} := -F_\gamma L$ must be stable. Thus, even though m_p has little effect on γ (or the performance at ω_p) it does reduce, often significantly, the performance at the other harmonics.

Remark 4.6.5 *The extension of the above to higher order weighting functions is obvious, and could potentially achieve better broad band repetitive action, particularly*

for the non-minimum phase case. The costs of higher order weights are higher order controllers and increased numerical difficulties.

The primary complication, with respect to weight selection, when using higher order weights, is choosing the appropriate relative spike heights to obtain the desired broad band repetitive performance. For all practical purposes, the required relative spike heights would have to be determined by trial and error. Numerical difficulties come from the fact that calculating the H^∞ optimal solution using Theorem 4.2.1 involves polynomial operations (such as root finding) that tend to become ill-conditioned as order increases.

Remark 4.6.6 *The numerical difficulties (such as those encountered in Chapter III) associated with the delay term $m(s) := e^{-sT}$ are greatly reduced due to the fact that all of the β_k (see Theorem 4.2.1) are nearly on the imaginary axis, thus keeping the $m(\beta_k)$ from getting excessively large.*

The fact that the β_k (zeros of E_γ) are very close to the imaginary axis is a direct result of the generic form chosen for $\widehat{W}_1$, see (4.42). This is most fortunate because if any of the β_k had significant real parts then it would be impossible to calculate the solution to the two-block H^∞ problem of Theorem 4.2.1 without resorting to multiprecision, as the sixteen decimal digits of precision available in MATLAB would not be sufficient.

Remark 4.6.7 *In an approximate sense, the above design procedure reduces to selecting the key harmonic, ω_p, and the approximate form of F_γ, via $\widehat{W}_2$, and calculating the corresponding optimal inner function L.*

This view of the design process raises the possibility of choosing $\widehat{W}_2$ to be proportional to the inverse of a higher order unity-low pass filter. However, even a higher order $\widehat{W}_2$ would tend to exacerbate the numerical difficulties due to $\widehat{W}_1$ having poles (and zeros) near the imaginary axis. Furthermore, excellent sensitivity improvement can be obtained using a second order (single spike) $\widehat{W}_1$ and a first order $\widehat{W}_2$, see Section 4.7.

Remark 4.6.8 *The reduced order solution, to either the sensitivity improvement or the direct H^∞ problems, is parameterized by $\check{q} := -F_\gamma L$, where the outer factor, F_γ, has the same order as $\widehat{W}_2$ and the inner factor, L, has order one less than the order of $\widehat{W}_1$.*

In other words, while for a single spike $\widehat{W}_1$, L is only first order, L would be third order for a two spike formulation. In this chapter we only consider the simple, minimum order, weights defined by (4.42).

4.7 Numerical Examples

In this section we will look at some numerical examples illustrating the above design techniques. We will focus our attention on the second step (the repetitive design), since the first step (approximate inversion) is a relatively simple and fairly well understood rational filter design problem. For a given robustness requirement we will look at the specific weighting functions that provide classical repetitive action, i.e. $\widehat{E}_\gamma \approx 1$, see Remark 4.3.10 in Subsection 4.3.3.

Consider the solution of the two block problem of Theorem 4.2.1. The corresponding $\hat{q}(s)$ gives the solution to the sensitivity improvement (cascade repetitive)

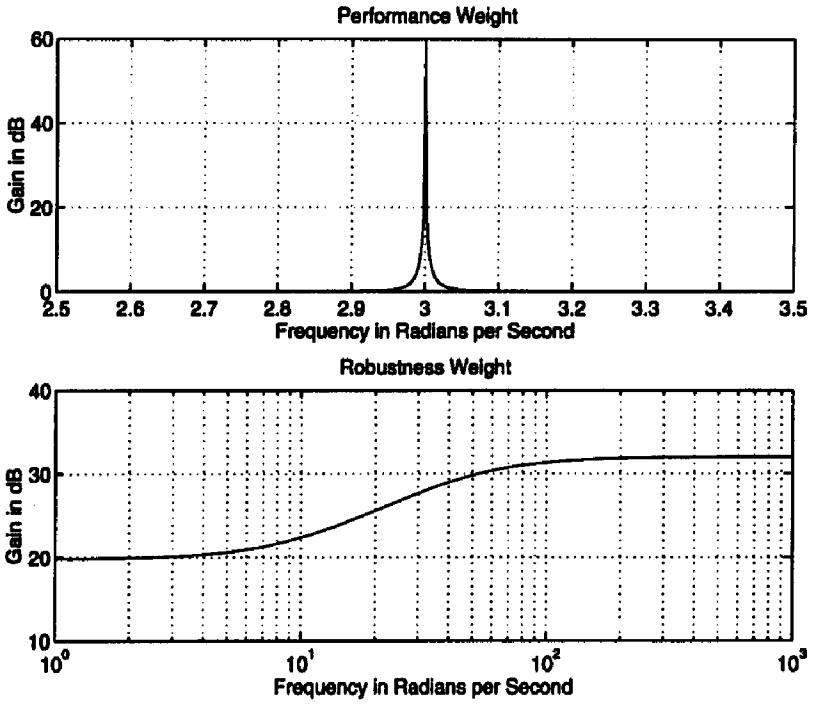


Figure 21: Bode magnitude plots of $\widehat{W}_1$ and $\widehat{W}_2$.

problem, see Definition 4.3.4 in Subsection 4.3.1, and $\hat{C}(s)$ is the solution to the direct H^∞ optimal repetitive control problem, see Definition 4.4.1. The two problems are completely equivalent, see Remark 4.4.1, in that the sensitivity improvement term, $\hat{S} := 1 - m_p m\hat{q}$, is precisely the sensitivity function for the direct problem. Thus the achievable performance of the two problems may be investigated simultaneously, with W_1 and W_2 replaced by $\widehat{W}_1$ and $\widehat{W}_2$ in Theorem 4.2.1.

Let the fundamental frequency of the periodic signal, to be tracked and/or rejected, be $\omega_o = .5\frac{rad}{sec} \Rightarrow m(s) = e^{-4\pi s}$. Let the "key" harmonic be the sixth harmonic. Finally, let the high frequency robustness requirement be that we must have $10dB$ of robustness by around $100\frac{rad}{sec}$. We now select the parameters of the generic

weighting functions, defined by (4.42), to satisfy these design requirements. Since the key harmonic is the sixth harmonic, we have $\omega_p = 6\omega_o = 3\frac{rad}{sec}$. We find that $\zeta_n = .005$ adequately isolates the phase to the sixth harmonic. Setting $\zeta_d = .000005$ gives a spike height at ω_p of $20\log(\zeta_n/\zeta_d) = 60dB$, which is sufficient to obtain controllers with repetitive action. This completes the selection of the performance weight $\widehat{W}_1$, see Figure 21 . Now for the robustness weight, $\widehat{W}_2$, we find that selecting $a_2 = 10.5$ and $b_2 = 43$ gives a high frequency to low frequency gain ratio of about $12dB$, see Figure 21, and reaches the high frequency level by $100\frac{rad}{sec}$. Letting $K_2 = 40$ gives $|\widehat{W}_2(j\omega_p)| \approx 10.13366$, which approximately determines γ. The reason that $|\widehat{W}_2(j\omega_p)|$ approximately determines γ, is that $|\widehat{W}_1(j\omega_p)| = 1000$, which implies that $|S(j\omega_p)| \approx 0$, which in turn implies that $|T(j\omega_p)| \approx 1$. Thus, at $\omega = \omega_p$, the performance is, almost solely, determined by $|\widehat{W}_2(j\omega_p)|$. Since this point is the "most challenging" point, the performance achievable at this point determines the overall optimal performance, γ. The resulting weighting functions are

$$\widehat{W}_1(s) = \frac{s^2 + .03s + 9}{s^2 + .00003s + 9} \quad \text{and} \quad \widehat{W}_2(s) = 40\frac{s + 10.5}{s + 43}. \tag{4.47}$$

For simplicity we will consider plants with only one right half plane (RHP) zero. Let z_p be the lone RHP zero of $P(s)$, giving

$$m_p(s) = \frac{s - z_p}{s + z_p}, \quad \text{where } 0 < z_p \in \mathbb{R}. \tag{4.48}$$

The "optimal performance" measure, γ, turns out to be essentially independent of z_p, where the quotation marks are to emphasize that the optimality is with respect to the solution of Theorem 4.2.1 rather than the underlying repetitive control problems,

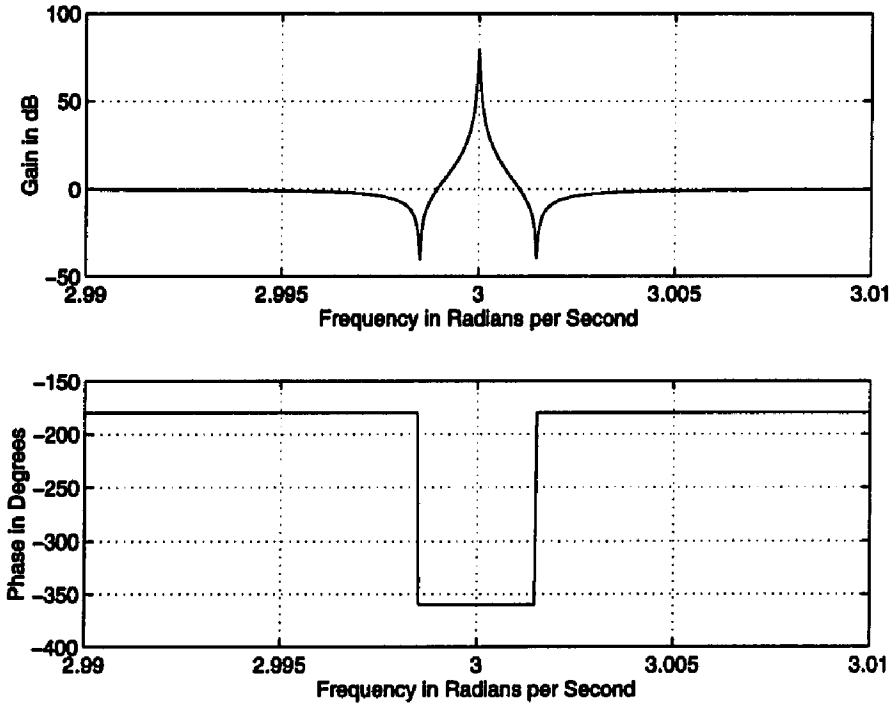


Figure 22: Localized non-constant behavior of $E_\gamma(j\omega)$ near $\omega_p = 3\frac{rad}{sec}$.

see Remark 4.6.4. In particular, the effects of m_p on γ are not seen in the first seven significant digits. Thus the seven significant digit approximation to γ given by $\gamma \approx |\widehat{W}_2(j\omega_p)| \approx 10.13366$, is not effected. The RHP plant zero has so little effect on γ, because γ is essentially determined at a single point, and is very large, see Remark 4.6.4. Thus, we have $\hat{E}_\gamma = 1 - \gamma^{-2} \approx 0.99 \approx 1$, see Remark 4.6.1. A detailed blow-up of the highly localized non-constant behavior of $E_\gamma(j\omega)$ is shown in Figure 22.

While γ is not significantly impacted by the RHP plant zero, z_p, the broad band repetitive performance is dramatically impacted for certain zero locations. Since γ is effectively independent of m_p, so are E_γ and F_γ. Thus the effect of m_p is manifest

Table 1: The RHP zero of L, z_L, and sign of $L(s)$ vs. the RHP plant zero, z_p.

z_p	0	0.15	0.60	1.50	3.00	6.00	15.0	60.0
z_L	0.3121	0.1613	31.915	7.9708	3.6968	1.9116	0.9315	0.4646
$L(s)$	$\frac{z_L-s}{z_L+s}$	$\frac{z_L-s}{z_L+s}$	$\frac{s-z_L}{s+z_L}$	$\frac{s-z_L}{s+z_L}$	$\frac{s-z_L}{s+z_L}$	$\frac{s-z_L}{s+z_L}$	$\frac{s-z_L}{s+z_L}$	$\frac{s-z_L}{s+z_L}$

solely through the inner controller parameter L. Recall that since $\widehat{W}_1$ is second order, L is first order. In particular the first order inner controller parameter L is given by

$$L(s) = \pm\frac{s - z_L}{s + z_L}, \quad \text{where } 0 < z_L \in \mathbb{R} \tag{4.49}$$

and z_L is the lone RHP zero of L. Both the sign of $L(s)$ and the RHP zero, z_L, vary as functions of the RHP plant zero, z_p. The variations of $L(s)$ for various values of z_p are shown in Table 1. Note that the sign is "$-$" for $z_p < \omega_0 = 0.5$ and "$+$" for $z_p > \omega_0 = 0.5$. The controller parameter L is not the only thing dramatically impacted by RHP plant zeros. The number of harmonics at which "good" performance is obtained is a very strong function of the RHP plant zero location, z_p.

Recall that the sensitivity improvement term, $\widehat{S} = 1 - m_p m\widehat{q}$, of the cascade design is precisely the sensitivity function of the direct design, see Remark 4.4.1. Thus the performance of both designs can be investigated simultaneously. The performance, as measured by attenuation in dB, is shown in Table 2 at the first fifteen harmonics for the full order/complexity case, see Figure 18 for cascade and Figure 20 for direct H^∞ optimal repetitive control. Note that for minimum-phase plants ($z_p = 0$), we obtain good broad band repetitive control. Indeed, there is strong attenuation at the fifth

Table 2: Full order/complexity case: Performance, measured in harmonic attenuation ($-20\log|\widehat{S}(j\omega_0 k)|$, $k = 1 \ldots 15$) in dB, and the corresponding RHP controller zero, z_L, for various RHP plant zeros, z_p.

z_p	Harmonics															z_L
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	-0	5	10	15	22	74	24	19	16	14	12	11	10	9	9	.312
.15	-1	5	10	15	22	74	24	19	16	14	12	11	10	9	9	.161
.6	-4	1	5	10	17	74	18	13	10	8	7	6	5	4	3	31.9
1.5	-5	-4	-0	4	11	74	11	6	4	2	0	-1	-1	-2	-2	7.97
3	-6	-4	-2	2	8	74	9	4	1	-0	-1	-2	-3	-3	-4	3.70
6	-6	-4	-1	3	9	74	11	5	2	1	1	-1	-2	-3	-3	1.91
15	-5	-2	2	7	14	74	15	10	7	5	3	2	1	1	-0	.932
60	-3	2	7	12	19	74	21	15	12	10	9	8	7	6	5	.465

through eighth harmonics and significant attenuation at the third through thirteenth harmonics. The performance, like the sign of L, is not significantly effected for $z_p = .15 < .5 = \omega_0$. However, as the RHP plant zero increases past the first harmonic, the "performance region" starts to shrink, reaching a worst case at $z_p = 3 = \omega_p$ (the sixth harmonic). The worst case performance region is very narrow, with significant attenuation at only the sixth harmonic. Clearly, such performance can hardly be called repetitive. As z_p is increased further we slowly begin to recover repetitive action. Finally, at $z_p = 60 = 20\omega_p$ we have recovered most of the minimum phase performance. However, while at $z_p = 0.15 = \omega_p/20$ we have no significant performance degradation, at $z_p = 60 = 20\omega_p$ there is noticeable performance degradation.

The Bode magnitude plot of $\widehat{S}$ is shown in Figure 23. Note that while all of the spikes (notches) appear to line up pretty well with the harmonics, the downward

spikes in the plot are considerably deeper than that predicted by the actual harmonic performance shown in Table 2. The only exception to this is the sixth harmonic, at which the spike is perfectly centered. The "focus" achieved at the sixth harmonic is a direct result of the spike in the performance weighting function, $\widehat{W}_1$. This phenomenon also illustrates a fundamental question in repetitive control: how to get the spikes to line up well at all of the important harmonics? There is no simple or uniformly applicable answer to this question, i.e. it cannot really be done. There are some approaches which have had some success in improving the situation which consist of breaking up the delay term, $m(s)$, into components whose product is $m(s)$, see e.g. [84]. The idea behind these methods is to try to improve the phase matching at the harmonics. The critical role of phase matching in achieving good repetitive performance (lining up the spikes at the harmonics), is exactly what makes repetitive control so difficult for non-minimum phase plants.

The problem with non-minimum phase plants is best illustrated by looking at the worst case example, $z_p = 3(= \omega_p)$, see Table 2. It is quite clear from the data in the table that there has been a dramatic decrease in actual performance. It can be seen from looking at the plot of $\hat{S}(j\omega) = 1 - m_p(j\omega)m(j\omega)\hat{q}(j\omega)$, for $z_p = 3$, see Figure 24, that this decrease is due to an increase in the error in the *location* of the spikes (again with exception of the sixth harmonic), rather than a change in the depth of the notches. In fact, this time the deviations in the spike locations from the harmonics can be easily seen with the naked eye. The spectrum is centered at the sixth harmonic, $\omega_p = 3rad/sec$, and all of the other spikes are closer to the sixth

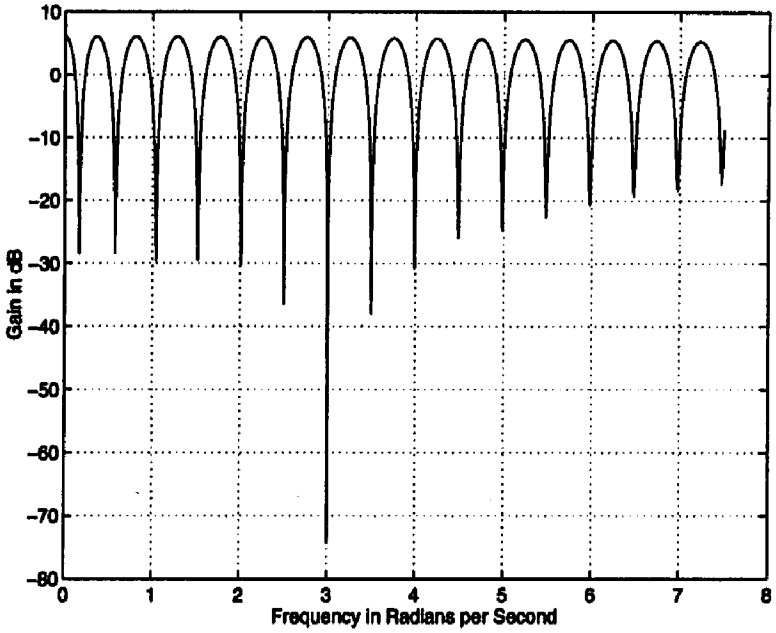


Figure 23: Bode magnitude plot of $\widehat{S}(j\omega) = 1 - m(j\omega)\widehat{q}(j\omega)$, i.e. $m_p \equiv 1$ $(z_p = 0)$.

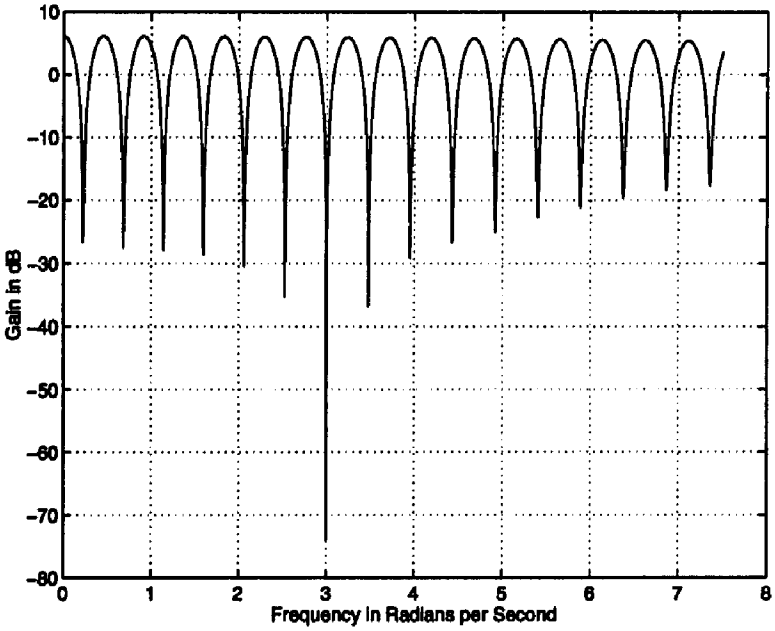


Figure 24: Worst case $(z_p = 3)$ Bode magnitude plot of $\widehat{S}$.

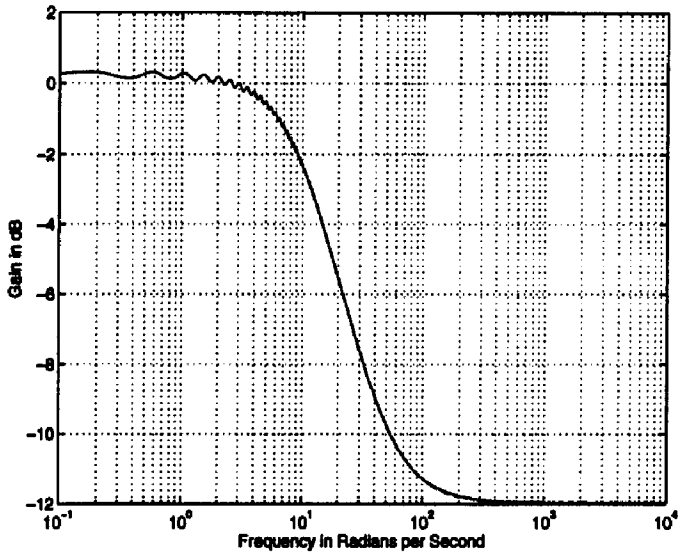


Figure 25: Bode plot of $|\widehat{T}(j\omega)| = |\widehat{q}(j\omega)|$.

harmonic than they should be. Note that there is also an extra downward spike prior to the first harmonic that appears in both of the performance plots. This extra spike is of little practical interest and causes no harm.

Robustness must be addressed separately for cascade and direct designs. While the complementary sensitivity for the direct design is given by $\widehat{T} := m_p m \widehat{q}$, the overall complementary sensitivity of the cascade design is given by $T = T_n + S_n \widehat{T}$, see (4.24). Thus, the overall robustness of the two designs will be very similar if $q_n(s)$ is "small" before $\widehat{q}(s)$ gets "small". However, at low frequencies, where $|\widehat{T}(j\omega)| \approx 1$, T exhibits wild oscillations due to the scaling of $\widehat{T}$ by the nominal repetitive sensitivity function $S_n := 1 - m_p m q_n$. The robustness of the direct design, for the minimum phase case, is shown in Figure 25. The robustness is exactly what one would expect from looking at the Bode magnitude plot of the robustness weight, $\widehat{W}_2$, see Figure 21, i.e. roughly

a scaled version of $\widehat{W}_2^{-1}$. Note that $|\hat{T}(j\omega)| := |m_p(j\omega)m(j\omega)\hat{q}(j\omega)| = |\hat{q}(j\omega)|$, thus this plot is valid for all z_p.

In order to more carefully examine the overall robustness of the sensitivity improvement (cascade repetitive) design, we must specify a particular nominal design, $q_n(s)$. Consider the common fifth order low pass filter formula

$$q_n(s) = \frac{1}{(\frac{s}{\omega_c}+1)(\frac{s^2}{\omega_c^2}+2\cos .2\pi\frac{s}{\omega_c}+1)(\frac{s^2}{\omega_c^2}+2\cos .4\pi\frac{s}{\omega_c}+1)}. \tag{4.50}$$

To make $q_n(s)$ "compatible" with the current design formulation, i.e. so that the overall design will meet our robustness requirement, let $\omega_c = 10\pi$. The corresponding Bode plot is shown in Figure 26, while the bode magnitude plot of the overall complementary sensitivity function for the cascade design is shown in Figure 27. Notice that the $-10dB$ point has shifted up from about $80\frac{rad}{sec}$ for the nominal design to about $120\frac{rad}{sec}$ for the cascade design. This trading of high frequency robustness for improved performance is characteristic of the cascade design process. However, in this case we see a dramatic improvement in performance (at least when $m_p(s)$ doesn't cause too much trouble) with only a slight decrease in robustness as measured by the $10dB$ attenuation frequency.

We now turn our attention to the problem of obtaining reduced order/complexity approximations to the controller parameters, $E_\gamma(s)$, $F_\gamma(s)$, and $L(s)$. As we saw above, $E_\gamma(s)$ can be well approximated by -1, except in a highly localized area around $\omega_p = 3\frac{rad}{sec}$. $E_\gamma(s)$ is anything but constant in this highly localized region, see Figure 22. This might lead one to (correctly) conclude that the performance at the sixth harmonic would be severely effected by setting $E_\gamma(s) = -1$. The resulting

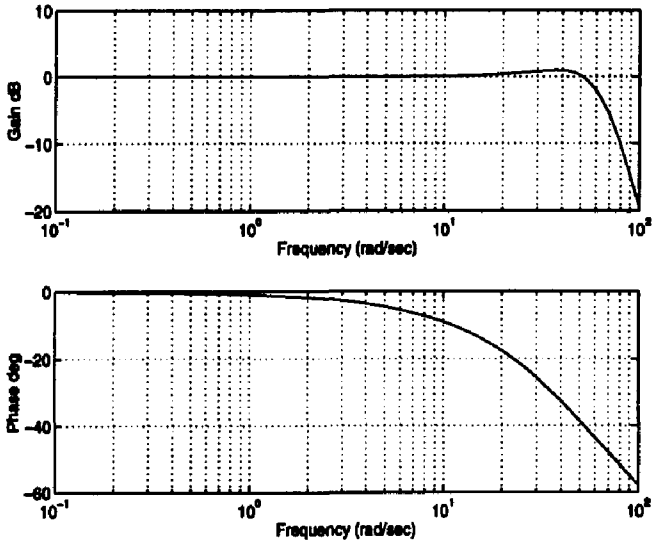


Figure 26: Bode plot of the fifth order nominal repetitive design parameter, $q_n(s)$.

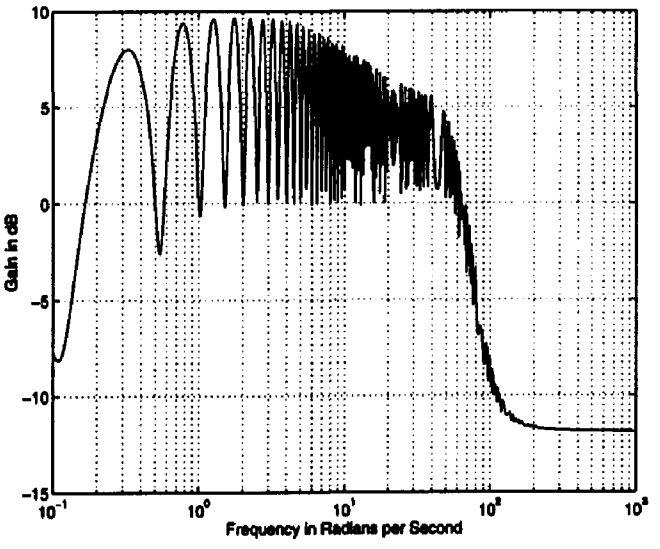


Figure 27: Bode magnitude plot of $T = T_n + S_n\hat{T}$ for the cascade design.

Table 3: Full order F_γ with constant $E_\gamma \approx -1$: Performance, measured in harmonic attenuation $(-20\log|\hat{S}(j\omega_0 k)|,\ k = 1\ldots15)$ in dB, and the corresponding RHP controller zero, z_L, for various RHP plant zeros, z_p.

z_p	Harmonics															z_L
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	-0	5	10	15	22	-7	24	19	16	14	12	11	10	9	9	.312
.15	-1	5	10	15	22	-7	24	19	16	14	12	11	10	9	9	.161
.6	-4	1	5	10	17	-7	19	13	10	8	7	6	5	4	3	31.9
1.5	-5	-3	-0	4	10	-6	12	7	4	2	0	-1	-1	-2	-2	7.97
3	-6	-4	-2	2	8	-6	10	4	1	-0	-1	-2	-3	-3	-4	3.70
6	-6	-4	-1	3	9	-6	11	5	3	1	-1	-1	-2	-3	-3	1.91
15	-5	-2	2	7	14	-6	15	10	7	5	3	2	1	1	0	.932
60	-3	2	7	12	19	-7	21	15	12	10	9	8	7	6	5	.465

performance, for $E_\gamma(s) = -1$, is shown in Table 3. Recall, see Remark 4.3.8, that for $E_\gamma(s) = -1$, we have $\hat{q} = \check{q} \Rightarrow \hat{S} = \check{S}$ and the simplified cascade repetitive structure of Figure 19, where $\check{q} := -F_\gamma L$. By comparing the data in this table with the data for the full order case shown in Table 2 it is clear that there is little effect at the other harmonics, but that the performance of the sixth harmonic has been destroyed. The good news is that the performance of the sixth harmonic can actually be restored, and then some, by making the appropriate cancellations in $F_\gamma(s)$.

Recall that there are structural cancellations that occur in forming the product $E_\gamma(s)F_\gamma(s)$, see Subsection 4.3.2. Thus, it is not surprising that making a reduced order approximation to $E_\gamma(s)$, without making the corresponding reductions in $F_\gamma(s)$, would lead to localized errors. What is somewhat surprising is that when the reductions in $F_\gamma(s)$ are made, the performance is actually improved at the sixth harmonic

(and is preserved elsewhere), see Table 4. Recall that $F_\gamma \approx \gamma \widehat{W}_2^{-1}$, see Remark 4.6.2 in Subsection 4.3.3. While this formula could be used directly to obtain a reduced order approximation for F_γ, a direct approximation of the full order F_γ was used instead. The direct approximation was obtained by extracting the pole and zero of $F_\gamma(s)$ that were not near ω_p, and keeping $F_\gamma(0)$, i.e. the DC gain, constant. In particular the reduced order approximation used in Table 4 is

$$F_\gamma(s) \approx 0.2545 \frac{s+43}{s+10.552}. \tag{4.51}$$

Since the value of γ, to seven significant digits, is the same for all z_p, see Remark 4.6.4, the approximation (4.51) is very similar to what one would obtain from the approximation formula. Specifically, the formula gives

$$F_\gamma(s) \approx \gamma \{\widehat{W}_2(s)\}^{-1} \approx 0.2533 \frac{s+43}{s+10.5}. \tag{4.52}$$

The two approximations give essentially identical performance everywhere except at the sixth harmonic. At the sixth harmonic, $\omega_p = 3\frac{rad}{sec}$, the approximation (4.52), using the formula, gives only $58dB$ of attenuation. Clearly, the approximation used constitutes the better design. However, it seems likely that the approximation (4.52), using the formula, is more accurate, where by more accurate we mean that it corresponds more closely to the "optimal" solution of Theorem 4.2.1. That is, since $\gamma \approx |\widehat{W}_2(j\omega_p)| \approx 10.13366$ and $20 \log |\widehat{W}_1(j\omega_p)| = 60dB$, the resulting attenuation at ω_p should be just a little less than $60dB$. Thus, it appears that the solution we obtained from the Theorem 4.2.1 suffers from minor numerical errors and is "suboptimal" in the sense that the performance is not really flat, but actually has a rather

Table 4: Reduced order F_γ with constant $E_\gamma \approx -1$: Performance, measured in harmonic attenuation ($-20\log|\widehat{S}(j\omega_0 k)|$, $k = 1 \ldots 15$) in dB, and the corresponding RHP controller zero, z_L, for various RHP plant zeros, z_p.

z_p	Harmonics															z_L
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	-0	5	10	15	22	103	24	19	16	14	12	11	10	9	9	.312
.15	-1	5	10	15	22	103	24	19	16	14	12	11	10	9	9	.161
.6	-4	1	5	10	17	103	19	13	10	8	7	6	5	4	3	31.9
1.5	-5	-3	-0	4	11	103	12	7	4	2	0	-1	-1	-2	-2	7.97
3	-6	-4	-2	2	8	103	10	4	1	-0	-1	-2	-3	-3	-4	3.70
6	-6	-4	-1	3	9	103	11	5	2	1	1	-1	-2	-3	-3	1.91
15	-5	-2	2	7	14	103	15	10	7	5	3	2	1	1	0	.932
60	-3	2	7	12	19	103	21	15	12	10	9	8	7	6	5	.465

significant highly localized spike at ω_p. However, this is a "happy" suboptimality in that the performance is actually *better* than it "should" have been.

Thus the final designs, using the constant approximation $E_\gamma \approx -1$ and the reduced (first) order approximation (4.51) for F_γ, have a second order $\check{q}$ as the only rational component (except for m_p, if the plant in non-minimum phase). The resulting cascade repetitive control design has the simplified structure shown in Figure 19. The direct H^∞ optimal repetitive control design has the simple repetitive structure shown in Figure 20, where the "$-E(s)$" block has been eliminated by the approximation $E_\gamma \approx -1$. In Table 5 we give the performance for the minimum phase case for both the direct and cascade designs. The direct design has only a single delay element, $m(s)$, and a single second order rational element, $\check{q}(s)$. The cascade design uses the nominal repetitive design given by $q_n(s)$ as defined by (4.50). Thus the

Table 5: Final reduced order/complexity designs: Performance, measured in harmonic attenuation ($-20\log|\widehat{S}(j\omega_0 k)|,\ k = 1\ldots 15$) in dB, minimum phase case, $z_p = 0$.

	Harmonics														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Direct	-0	5	10	15	22	103	24	19	16	14	12	11	10	9	9
Cascade	42	41	43	45	50	130	49	43	39	36	33	32	30	29	27

overall cascade repetitive controller has two delay elements, $m(s)$, and three rational elements, $q_n(s)$ twice and $\check{q}(s)$, giving a total order of twelve. However, if the add-on nature of the controller structure is not required, e.g. if $q_n(s)$ is not a pre-existing design, then the roles of $q_n(s)$ and $\check{q}(s)$ can be *reversed*, resulting in a total controller order of only nine.

The performance of the cascade design is truly outstanding. However, while the direct design provides fairly good broad band repetitive action (i.e. good attenuation at several harmonics), it does not provide good performance at the fundamental frequency, ω_0. Thus, this particular direct design would be unacceptable for any practical application. There are three different ways in which the performance of the direct design at ω_0 could be improved. First, one could simply move the "key" harmonic, ω_p, closer to ω_0, e.g. $\omega_p = 3\omega_0 = 1.5\frac{rad}{sec}$. Second, a higher order robustness weight could be used, see Remark 4.6.7. Finally, a higher order performance weight could be used with one spike at, or near, the fundamental frequency and another at a higher frequency.

4.8 Conclusions

In this chapter we have seen that the repetitive performance with robust stability control design problem can be reduced to the solution of two, lightly coupled, design problems: a finite dimensional rational design problem, which stabilizes and approximately inverts the plant, and an infinite dimensional two-block H^∞ optimal control problem which provides repetitive performance with robust stability. The solution of the resulting infinite dimensional two-block H^∞ optimal control problem was shown to have two interpretations: one as the solution to a sensitivity improvement formulation, which results in a simple cascade repetitive structure, and the other as a direct H^∞ optimal repetitive control formulation.

The general structure of the two degree of freedom controller formulation is shown in Figure 15. The initial stabilizing controller, $d(s)$, represents the first degree of freedom and "performance" controller, $C(s) = b(s)a(s)$, represents the second degree of freedom. The compensator, $b(s)$, approximately inverts the plant, while the repetitive controller block, $a(s)$, provides the repetitive action in a robustly stable manner. The design of $d(s)$ and $b(s)$ constitutes the finite dimensional (or non-repetitive) portion of the design and is not the subject of this chapter. However, the interaction with the repetitive design is quantified in Lemma 4.1.1 and related design issues are discussed in Section 4.5. The repetitive controller, $a(s)$, is designed using the equivalent system of Figure 16. The two figures are related by $a(s) =: m(s)\tilde{C}(s)$ and $m_p(s) = b(s)P(s) =: b(s)P_o(s)m_p(s)$, where $m(s) := e^{-sT}$ is the pure delay associated with repetitive control and $P_o(s)m_p(s)$ is an inner-outer factorization of the stabilized

plant, $P(s)$. Note that the equivalent system of Figure 16 assumes perfect inversion of the outer part, $P_o(s)$ of the plant. The effects of imperfect inversion are addressed in Lemma 4.1.1 and Lemma 4.1.2.

The sensitivity improvement problem is given by Definition 4.3.3. The associated simple cascade repetitive controller structure is shown in Figure 19. The choice of weighting functions that yield this simple cascade repetitive structure is discussed in Section 4.6. The direct H^∞ optimal repetitive control problem is given by Definition 4.4.1. The associated repetitive controller structure is shown in Figure 20. Note that the same choice of weighting functions that yield the simple cascade structure of Figure 19 leads to the elimination of the "$-E(s)$" block (under the approximation $E_\gamma \approx -1$) in the direct H^∞ repetitive controller of Figure 20. The solutions of the two problems are equivalent in that the improvement in sensitivity achieved in the cascade design is exactly the same as the sensitivity of the direct design, see Remark 4.4.1. The solution of both problems is obtained from Theorem 4.2.1, which is a specialized version of a much more general result due to Toker and Özbay [95]. For properly selected weighting functions, the solution of both problems is in terms of the unity-low pass classical repetitive controller parameter, $\check{q} := -F_\gamma(s)L(s)$, where L is as defined in Theorem 4.2.1 and F_γ is a reduced order approximation, see Remark 4.6.2. Recall that under the approximation $E_\gamma \approx -1$, see Remark 4.3.8, the sensitivity (improvement, for the cascade design) is given by $\check{S}(s) := 1 - m_p(s)m(s)\check{q}(s)$, which has a classical repetitive structure. While Theorem 4.2.1 does not ensure the stability of $L(s)$, the above choice of weights does, see Remark 4.4.2. Thus, $\check{q} \in H^\infty$, which

ensures stability of the closed loop system, see Remark 4.3.9.

A very interesting feature of the repetitive controller parameter, $\check{q}$, is that it is always non-minimum phase. This is a novel feature of repetitive controllers resulting from the two-block H^∞ formulation of this chapter. The natural way that the non-minimum phase structure of $\check{q}$ arises from the H^∞ optimal formulation, suggests that non-minimum phase repetitive controller parameters may, in some sense, be "more optimal" for achieving robustly stable repetitive controllers. This possibility has obvious implications for "classical" repetitive controller design. Furthermore, the resulting repetitive controller structure has another novel feature (whenever the plant is non-minimum phase) via the inclusion of $m_p(s)$ in the feedback path of the "repetitive part", see Figure 20. Once again, the natural way in which this arises has possible implications for "classical" repetitive controller design. Finally, the simple cascade repetitive controller structure is a novel structure that can be thought of as a means of robustly cascading two repetitive controller designs irrespective of how they were obtained, see Remark 4.3.11.

The properties of designs using these procedures are explored in Section 4.7. The sensitivity (improvement, for the cascade case), $\hat{S}$ (= $\check{S}$ under $E_\gamma \approx -1$), of the two-block H^∞ solution is explored for first order $m_p(s)$. The performance is investigated for various locations of the single right half plane (RHP) plant zero. As expected, see Remark 4.3.3, the existence of RHP zeros can significantly reduce achievable performance, and the performance is most severely effected when the phase of $m_p(j\omega)$ varies the most over the performance region. In particular, for a simple real RHP

zero this occurs when the zero, z_p, equals the "key" harmonic (center of the performance region), ω_p, see Table 4. The performance of the direct design (improvement of the cascade design) and the overall cascade performance, using a fifth order nominal repetitive design (4.50), are shown in Table 5. While the direct design exhibits unacceptable performance at the fundamental frequency (first harmonic), ω_0, this is because the weights were really selected for the sensitivity improvement (cascade repetitive) design. The performance of the cascade repetitive design is excellent and indicates the power of the repetitive structure and the utility of the two-block H^∞ approach. Perhaps the most significant point about the direct design is that it proves that H^∞ optimal repetitive control formulations using rational weights will yield controllers providing "classical repetitive action" (see Remark 4.3.10), for properly selected weighting functions. This provides an indirect proof of concept for the robust performance formulation of Chapter III.

CHAPTER V

Sampled-data Repetitive Control

In this chapter we propose a natural definition for sampled-data repetitive control as satisfying a continuous-time tracking requirement and having a discrete-time repetitive structure. We formulate an H^∞ optimal sampled-data problem, which imposes the continuous-time tracking requirement, and outline a procedure for obtaining controllers with a discrete-time repetitive structure.

5.1 Approach to the Sampled-data Problem

We will consider sampled-data control design problems of the form shown in Figure 28, where any required sampling pre-filters have been included in the generalized plant, G. Figure 28 has the structure of the "standard problem", see e.g. [24], where the performance/robustness of the system is characterized by the transfer function T_{zw}.

Remark 5.1.1 *The extension of standard problem to sampled-data control, see e.g. [5, 46], can be thought of as control design using time-varying continuous-time controllers of the special form $\mathcal{H}_\tau C \mathcal{S}_\tau$, where C is a discrete-time controller.*

We assume ideal sampling, fixed zero-order hold, synchronized sample and hold functions operating at the same rate, and zero computational delay. This common

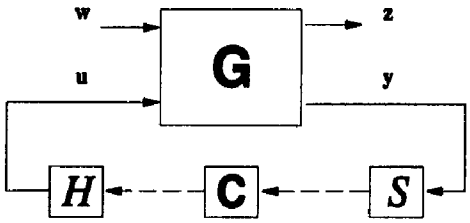


Figure 28: Basic form of the sampled-data problem.

approach, see e.g. [105, 82, 25, 19], allows the use of the lifting technique based approach of [5, 104, 4].

Remark 5.1.2 *While the computational delay is never actually zero for any real system, the approximation may be reasonable for low order (e.g. repetitive) controllers.*

The other main approach, a delay of one sample period, is addressed in [46]. Their approach is also based on the lifting technique, but has several other significant differences in addition to the inclusion of a computational delay. The ultimate goal of the lifting technique is to find an "equivalent" finite dimensional discrete-time system that is norm preserving. Then, the equivalent discrete-time system can be used for norm based discrete-time controller synthesis using the corresponding norm. Obviously, in this dissertation we are concerned with equivalence in terms of the infinity-norm.

5.2 The Lifting Technique

The purpose of the lifting technique is to obtain a discrete-time system that represents a sampled continuous-time system without *any* loss of information. The basic idea is quite simple and is defined in terms of action on signals. Define a lifting operator,

W_τ, that piecewise lifts continuous-time signals to discrete-time sequences of time limited functions. The action of W_τ is illustrated in Figure 29 and is defined by

$$W_\tau : L_e^p[0,\infty) \to l_{L^p[0,\tau]}; \quad \hat{f}_i = W_\tau f, \quad \hat{f}_i(t) = f(\tau i + t), \quad 0 \le t < \tau, \tag{5.1}$$

where the function spaces $L_e^p[0,\infty)$ and $l_{L^p[0,\tau]}$ are defined in Appendix A. The lifting operator can be thought of as a "generalized sampler", which instead of sampling just a single value of a function, samples the entire history of the function between sampling instants. The lifting, W_τ, converts time unlimited functions into infinite sequences of time limited functions. The operator W_τ is invertible, where the inverse operation is defined by

$$g = W_\tau^{-1}\hat{g}, \quad g(t) = \hat{g}_i(t - \tau i), \quad \tau i \le t < \tau(i+1). \tag{5.2}$$

When W_τ is restricted to $L^p[0,\infty) \subset L_e^p[0,\infty)$, it is an *isometry*, i.e. $W_\tau^{-1}W_\tau$ is the identity operator, and is therefore norm preserving.

Remark 5.2.1 *The inverse lifting operator, W_τ^{-1}, defined by (5.2) is noncausal. Indeed, if it were defined to be causal, then $W_\tau^{-1}W_\tau$ would be the unit delay operator instead of the identity operator.*

The purpose in defining the inverse lifting operator in this way is to use it in turn to define a lifting of systems. The induced system lifting can be thought of as a "similarity transform" of the original system under the lifting operator. The lifted system, $\hat{G}$, is given by

$$\hat{G} = W_\tau G W_\tau^{-1}. \tag{5.3}$$

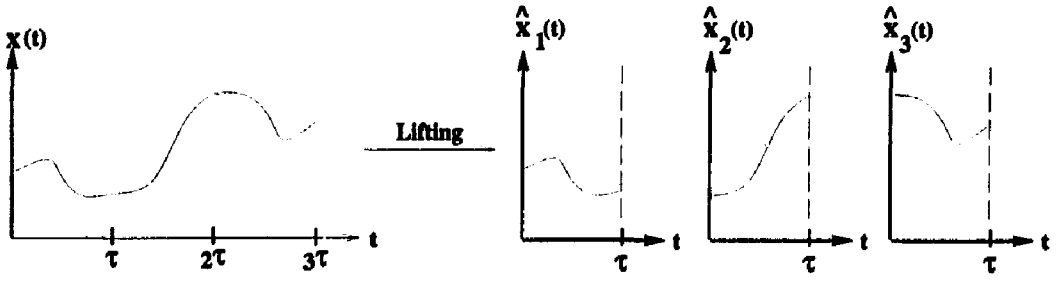


Figure 29: Graphical illustration of the lifting process.

Remark 5.2.2 *In [46] Kabamba and Hara avoid the use of the inverse lifting operator and obtain a much more complicated characterization of a discrete-time equivalent (with respect to the infinity norm) system.*

Since W_τ is an isometry *only* when it is restricted to $L^p[0,\infty)$, strictly speaking the equivalent system is valid only for L^p-stable G. However, by restricting the domain of G (through feedback control) the output can be restricted to $L^p[0,\infty)$. That is, if C stabilizes G in the hybrid system of Figure 28, then W_τ is an isometry and the lifted system is well defined. For the problem to be well posed we must have that the stability of $\mathcal{F}(\hat{\mathcal{H}}_\tau \hat{G} \hat{\mathcal{S}}_\tau, C)$ implies the stability of $\mathcal{F}(G, \mathcal{H}_\tau C \mathcal{S}_\tau)$, where the operator $\mathcal{F}$ is defined by Figure 28. By Theorem 4 of [34], the stability of the two systems $\mathcal{F}(\hat{\mathcal{H}}_\tau \hat{G} \hat{\mathcal{S}}_\tau, C)$ and $\mathcal{F}(G, \mathcal{H}_\tau C \mathcal{S}_\tau)$ is equivalent for all non-pathological sampling periods, τ, for all finite dimensional linear time-invariant (LTI) generalized plants, G.

Definition 5.2.1 (pathological sampling periods) *Pathological sampling periods are those sampling periods for which the continuous-time system has eigenvalues in*

C_+ that are multiples of the sampling frequency or whose imaginary parts differ by a multiple of the sampling frequency.

Thus the control problem for the lifted system is well defined and equivalent to the original control problem for all finite dimensional *LTI* plants under non-pathological sampling.

The "lifted" sample and hold operators, $\hat{\mathcal{H}}_\tau$ and $\hat{\mathcal{S}}_\tau$, are equivalent to $\mathcal{H}_\tau$ and $\mathcal{S}_\tau$ with the appropriate changes in the range space of $\mathcal{H}_\tau$ and the domain of $\mathcal{S}_\tau$. The lifted sample and hold operators are defined by

$$\hat{\mathcal{H}}_\tau : \mathbb{R}^u \to L^p[0,\tau]; \quad \hat{u}_k = \hat{\mathcal{H}}_\tau \tilde{u}_k \quad \Leftrightarrow \quad \hat{u}_k(t) = \tilde{u}_k, \quad 0 \le t < \tau \quad \text{and} \tag{5.4}$$

$$\hat{\mathcal{S}}_\tau : L^p[0,\tau] \to \mathbb{R}^y; \quad \tilde{y}_k = \hat{\mathcal{S}}_\tau \hat{y}_k \quad \Leftrightarrow \quad \tilde{y}_k = \hat{y}_k(0). \tag{5.5}$$

Remark 5.2.3 *The superscripts are used to keep track of the space in which the discrete-time variables live. For example, $\tilde{y}$ denotes a real valued discrete-time variable, while $\hat{y}$ denotes a function valued discrete-time variable.*

We can now absorb $\hat{\mathcal{H}}_\tau$ and $\hat{\mathcal{S}}_\tau$ into the plant to get the system shown in Figure 30. The new plant, $\tilde{G}$ is given by

$$\tilde{G} = \begin{bmatrix} \hat{G}_{11} & \hat{G}_{12}\hat{\mathcal{H}}_\tau \\ \hat{\mathcal{S}}_\tau \hat{G}_{21} & \hat{\mathcal{S}}_\tau \hat{G}_{22} \hat{\mathcal{H}}_\tau \end{bmatrix}, \quad \text{where} \quad \begin{bmatrix} \hat{z} \\ \tilde{y} \end{bmatrix} = \tilde{G} \begin{bmatrix} \hat{w} \\ \tilde{u} \end{bmatrix}. \tag{5.6}$$

The mappings of the blocks of $\tilde{G}$ are as follows:

$$\tilde{G}_{11} : L^p[0,\tau] \to L^p[0,\tau], \quad \tilde{G}_{12} : l_{\mathbb{R}^u} \to L^p[0,\tau], \tag{5.7}$$

$$\tilde{G}_{21} : L^p[0,\tau] \to l_{\mathbb{R}^y}, \quad \text{and} \quad \tilde{G}_{22} : l_{\mathbb{R}^u} \to l_{\mathbb{R}^y}. \tag{5.8}$$

Remark 5.2.4 *Since $\hat{w} = W_\tau w$ and $\hat{z} = W_\tau z$, we have*

$$\mathcal{F}(\tilde{G}, C) = W_\tau \mathcal{F}(G, \mathcal{H}_\tau C \mathcal{S}_\tau) W_\tau^{-1} \Rightarrow \|\mathcal{F}(\tilde{G}, C)\| = \|\mathcal{F}(G, \mathcal{H}_\tau C \mathcal{S}_\tau)\|. \qquad (5.9)$$

Thus the control problem for this discrete-time system is equivalent to the control problem for the sampled-data system of Figure 28.

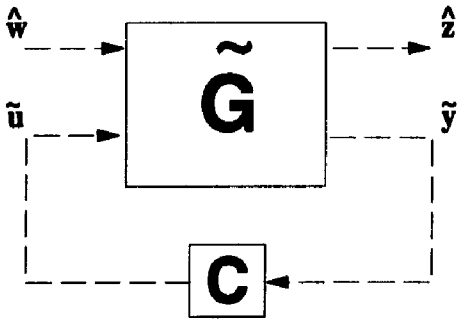


Figure 30: The discrete-time lifted problem.

5.3 The Discrete-time Equivalent for the H^∞ Problem

For the details of the above development see [5]. The development in the preceding section is completely general in that it is valid for all norms. When the norm is the infinity-norm, the equivalent problem $\tilde{G}$ is in turn equivalent to a finite dimensional problem $\acute{G}$, see [5]. The equivalence is with respect to internal stability and $\|\mathcal{F}(G, \mathcal{H}_\tau C \mathcal{S}_\tau)\|_\infty < 1 \Leftrightarrow \|\mathcal{F}(\acute{G}, C)\|_\infty < 1$. The finite dimensional equivalent system is given by:

$$\acute{G} := \left(\acute{A}, \begin{bmatrix} \acute{B}_1 \\ \acute{B}_2 \end{bmatrix}, \begin{bmatrix} \acute{C}_1 & \acute{C}_2 \end{bmatrix}, \begin{bmatrix} 0 & \acute{D}_{12} \\ 0 & 0 \end{bmatrix} \right), \qquad (5.10)$$

where the real finite dimensional matrices are defined in terms of finite rank operators as follows: $\acute{C}_2 := C_2$,

$$\acute{A} := e^{A\tau} + \hat{B}_1 \hat{D}_{11}^* \left[I - \hat{D}_{11} \hat{D}_{11}^* \right]^{-1} \hat{C}_1, \tag{5.11}$$

$$\acute{B}_2 := \hat{B}_1 \hat{D}_{11}^* \left[I - \hat{D}_{11} \hat{D}_{11}^* \right]^{-1} \hat{D}_{12} + \Psi(\tau) B_2, \tag{5.12}$$

$$\acute{B}_1 := T_B^* \begin{bmatrix} \Sigma_B^{1/2} \\ 0 \end{bmatrix}, \text{ and } \begin{bmatrix} \acute{C}_1 & \acute{D}_{12} \end{bmatrix} := \begin{bmatrix} \Sigma_{CD}^{1/2} & 0 \end{bmatrix} T_{CD}; \tag{5.13}$$

where $\Psi(t) := \int_0^t e^{As} ds = \int_0^t e^{A(t-s)} ds$,

$$T_B^* \begin{bmatrix} \Sigma_B & 0 \\ 0 & 0 \end{bmatrix} T_B := \hat{B}_1 \left[I - \hat{D}_{11}^* \hat{D}_{11} \right]^{-1} \hat{B}_1^*, \text{ and} \tag{5.14}$$

$$T_{CD}^* \begin{bmatrix} \Sigma_{CD} & 0 \\ 0 & 0 \end{bmatrix} T_{CD} := \begin{bmatrix} \hat{C}_1^* \\ \hat{D}_{12}^* \end{bmatrix} \left[I - \hat{D}_{11} \hat{D}_{11}^* \right]^{-1} \begin{bmatrix} \hat{C}_1 & \hat{D}_{12} \end{bmatrix}. \tag{5.15}$$

The symbols with hats over them are finite rank infinite dimensional Volterra integral operators, see e.g. [63], defined over the interval $[0, \tau]$ and their kernels are given by:

$$b_1(s) := e^{A(\tau - s)} B_1, \quad c_1(s) := C_1 e^{As}, \quad d_{12}(s) := C_1 \Psi(s) B_2 + D_{12} \text{ and}$$

$$d_{11}(t,s) := C_1 e^{A(t-s)} 1(t-s) B_1 + D_{11} \delta(t-s). \tag{5.16}$$

Remark 5.3.1 *The action of an operator, $\hat{H}$, with kernel $h(t,s)$ or $h(s)$ is given by $\hat{H} f(t) = \int_0^\tau h(t,s) f(s) ds$ or $\hat{H} f(t) = \int_0^\tau h(s) f(s) ds$, respectively.*

Ignoring the dimension of the vectors (matrices) in the domains and ranges, the elements are mapped as follows: $\hat{B}_1 : L^2[0,\tau] \to \mathbb{R}$, $\hat{C}_1 : \mathbb{R} \to \mathbb{R}$, $\hat{D}_{11} : L^2[0,\tau] \to L^2[0,\tau,]$ and $\hat{D}_{12} : \mathbb{R} \to \mathbb{R}$.

Remark 5.3.2 *The above characterization is in terms infinite dimensional, finite rank, operators. In order for the formulation to be usable, the operators must be explicitly evaluated.*

This was done in [5] for the special case where D_{11} and D_{12} are identically equal to zero and the pairs (A, B_1) and (A, C_1) are controllable and observable respectively.

5.4 Sampled-Data Problem Formulation

In this section we define a notion of sampled-data repetitive control and formulate a *SISO* H^∞ optimal sampled-data repetitive control problem. In order to make use of the above development, we must pose the problem in the form of Figure 28. For maximum generality we consider two degree of freedom controllers. We formulate a tracking problem, since there is no advantage to the two degree of freedom controller for the disturbance rejection problem [102]. We pose a nominal performance problem with a robust stability requirement. The formulation also provides an unspecified degree of robustness with respect to the tracking performance. The resulting *MIMO* H^∞ problem has a very similar structure to the robust performance problem posed in Chapter III for continuous time H^∞ optimal repetitive control design.

5.4.1 Sampled-Data Repetitive Control

We propose the following formal definition for sampled-data repetitive controllers, which corresponds to the purposes behind both repetitive control and sampled-data system theory.

Definition 5.4.1 (sampled-data repetitive control) *A sampled-data repetitive controller is a digital controller having a digital repetitive structure that satisfies a continuous time performance (tracking/disturbance rejection) requirement.*

This definition is reasonable in that repetitive control is concerned with handling a large number of harmonics with relatively low order (easily implemented) controllers and sampled-data controllers are digital controllers designed directly to satisfy continuous-time requirements. Previous works related to sampled-data repetitive control do not satisfy Definition 5.4.1. In [40] Hara, et al. consider a digital repetitive control problem where the reference signal is taken to be digital. The drawback of this formulation is that a continuous system cannot precisely track a discrete signal, even if the desired performance actually is of this form. They attempt to suppress the inevitable intersample ripples through the use of a periodic hold function. In [55] the authors take another indirect approach to sampled-data repetitive control that does not satisfy Definition 5.4.1. They approximate the continuous-time requirements at a sampling rate that is an integer multiple of the actual sampling rate, which does not satisfy the first part of Definition 5.4.1. Furthermore, no mention is made of attempting to recover a digital repetitive structure. Recall that the general structure of digital repetitive controllers is

$$C(z) = \frac{d(z)}{1 - q(z)z^{-L}}, \tag{5.17}$$

where $q(z)$ and $d(z)$ are rational functions and the integer L is at least twice the ratio of the sampling frequency, $f_s = 1/\tau$, to the fundamental frequency, f_0, of the the periodic input.

Remark 5.4.1 *The sampling rate, f_s, must be chosen to be an integer multiple of the fundamental frequency, f_0, of the reference/disturbance signal.*

If this condition is not satisfied, then it is not possible for a controller, with a "low order" digital repetitive structure, to satisfy the continuous-time repetitive performance requirement.

Remark 5.4.2 (order of digital repetitive controllers) *Strictly speaking, the order of digital repetitive controllers is* always *at least twice the number of harmonics of interest, due to the term z^{-L}. However, as a practical matter, it is the order of the two rational functions, $q(z)$ and $d(z)$, that determine the computation time (ease of implementation).*

Indeed, if $q(z) = d(z) \equiv 1$, then only one addition operation is required to implement the controller defined by (5.17).

5.4.2 H^∞ Optimal Tracking Problem with Robust Stability

The block diagram for the tracking performance problem with robust stability is shown in Figure 31. The signals are labeled in accordance with the general structure shown in Figure 28. The performance weight, W_1, represents the frequency distribution of the signal to be tracked. The robustness weight, W_2, represents the additive uncertainty in the plant P. The sampling pre-filters, F_1 and F_2, are included to ensure that the transfer function from any exogenous input to any sampled output is low pass, which is necessary for the sampling process to be well defined. The H^∞

tracking problem is to minimize the worst case tracking error, z_1, due to two-norm bounded input signals w_1.

Remark 5.4.3 *In the pure tracking problem a fictitious weight would have to be placed on the control signal, u, to prevent the resulting "optimal" controller from being improper.*

In our formulation the stability robustness requirement places a weight on the controller output,u, and takes care of this problem. The robust stability problem is to ensure that the two-norm of the robustness output, $z_2(t)$, satisfies $\|z_2\|_2 < 1$, for all inputs, $w_2(t)$, such that $\|w_2\|_2 < 1$. Thus, $(T_{zw})_{11}$ represents the tracking problem and $(T_{zw})_{22}$ represents the robust stability problem. The off diagonal terms of T_{zw} represent an unspecified robust tracking requirement. Thus, solving for $\|T_{zw}\|_\infty < 1$ minimizes the tracking error while guaranteeing robust stability and an unspecified degree of robustness with respect to the tracking requirement.

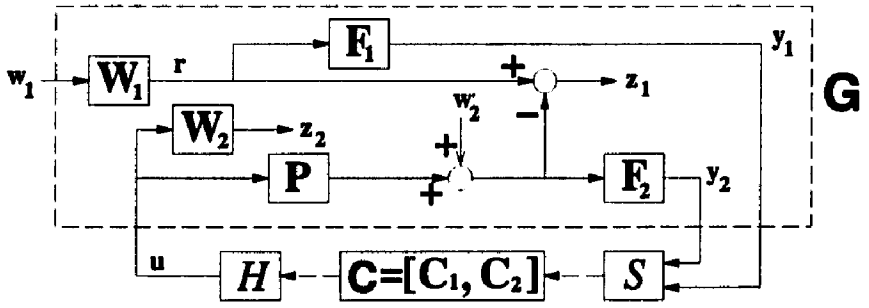


Figure 31: Block diagram of the problem definition.

By inspection of Figure 31, it can be seen that it has the same form as Figure 28 with the generalized plant, G, given by

$$G = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}, \quad \text{where} \quad \begin{bmatrix} z \\ y \end{bmatrix} = G \begin{bmatrix} w \\ u \end{bmatrix} \tag{5.18}$$

and the entries of G are given by

$$G_{11} = \begin{bmatrix} W_1 & 1 \\ 0 & 0 \end{bmatrix}, \; G_{12} = \begin{bmatrix} P \\ W_2 \end{bmatrix}, \; G_{21} = \begin{bmatrix} W_1F_1 & 0 \\ 0 & F_2 \end{bmatrix}, \; G_{22} = \begin{bmatrix} 0 \\ F_2P \end{bmatrix}. \quad (5.19)$$

Clearly, the presence of a low pass F_2 is necessary. The necessity of F_1 depends on the form of W_1. If W_1 is taken to be biproper, then F_1 must be low pass also. The continuous-time performance/robustness requirements are characterized by the closed loop transfer function T_{zw}. For the problem defined by Figure 31, we have

$$T_{zw} = \frac{1}{1 - P\mathcal{H}C_2\mathcal{S}F_2} \begin{bmatrix} W_1(1 - P\mathcal{H}(C_2\mathcal{S}F_2 - C_1\mathcal{S}F_1)) & 1 \\ W_1W_2(\mathcal{H}C_1\mathcal{S}F_1) & W_2(\mathcal{H}C_2\mathcal{S}F_2) \end{bmatrix}. \quad (5.20)$$

Remark 5.4.4 *The diagonal terms are analogous to two of the three terms in the continuous-time robust performance problem formulation given in Chapter III, see Definition 3.1.1.*

Indeed, ignoring the samplers, holds, and pre-filters and replacing C_2 by $\hat{C}_2$ and $C_2 - C_1$ by C_2 the diagonal terms become precisely the first and third entries in the vector H^∞ problem of Chapter III, see Definition 3.1.1. These diagonal terms represent the nominal tracking and robust stability, respectively. The off diagonal terms represent additional constraints in the form of robust tracking requirements. However, no specific level of robustness with respect to the tracking requirement is guaranteed.

5.5 Discrete-Time Equivalent Problem

For (5.10) to be useful in design, explicit formulas for the associated matrices must be derived in terms of the continuous-time system and performance weights. This was

done in [5] for the special case where D_{11} and D_{12} are identically equal to zero and the pairs (A, B_1) and (A, C_1) are controllable and observable respectively. None of these assumptions hold for our sampled-data repetitive control formulation, defined by Figure 31.

Remark 5.5.1 *The characterization of the discrete-time equivalent system given in [46] is completely general, however the computational complexity is prohibitive.*

Therefore, in this section we directly extend the results of Bamieh and Pearson [5] to the general case. The formulas given in [5] are a special case of these formulas and can be recovered by substituting in $D_{11} = 0$ and $D_{12} = 0$. The controllability and observability assumptions are technical conditions related to uniqueness of the solution for the special case considered there.

5.5.1 Explicit formulas for the general case

In this section we give explicit formulas for constructing the discrete-time equivalent system from the continuous-time problem data for the general case. The two operators $\left[I - \hat{D}_{11}\hat{D}_{11}^*\right]^{-1}$ and $\left[I - \hat{D}_{11}^*\hat{D}_{11}\right]^{-1}$ play very prominent roles in these formulas. The kernels of these two operators are derived in this section. The derivations are analogous to, but considerably more involved than, those in [5] for the special case discussed above. Theorem 5.5.1 summarizes the required formulas and gives a sufficient condition for their validity, which is a necessary condition for the solvability of the associated H^∞ control problem. The proof of Theorem 1 appears in Appendix D.

The kernels of these operators are most easily derived in the time domain using

differential equation representations. Let $h(t) := \hat{D}_{11}u(t),\ 0 \le t \le \tau$, then we have

$$\dot{x}_1(t) = Ax_1(t) + B_1u(t),\ h(t) = C_1x_1(t) + D_{11}u(t),\ x_1(0) = 0. \tag{5.21}$$

The adjoint operator, $\hat{D}_{11}^*,\ 0 \le t \le \tau$, has the differential equation representation

$$\dot{x}_2(t) = -A^*x_2(t) - C_1^*h(t),\ f(t) = B_1^*x_2(t) + D_{11}^*h(t),\ x_2(\tau) = 0, \tag{5.22}$$

where $f(t) := \hat{D}_{11}^*h(t)$. Thus, $y(t) := \left[I - \hat{D}_{11}^*\hat{D}_{11}\right]u(t),\ 0 \le t \le \tau$, is given by

$$\begin{bmatrix} \dot{x}_2(t) \\ \dot{x}_1(t) \end{bmatrix} = \begin{bmatrix} -A^* & -C_1^*C_1 \\ 0 & A \end{bmatrix}\begin{bmatrix} x_2(t) \\ x_1(t) \end{bmatrix} + \begin{bmatrix} C_1^*D_{11} \\ -B_1 \end{bmatrix}u(t),\ \begin{bmatrix} x_2(\tau) \\ x_1(0) \end{bmatrix} = 0,$$

$$y(t) = \begin{bmatrix} B_1^* & D_{11}^*C_1 \end{bmatrix}x(t) + (I - D_{11}^*D_{11})\,u(t),\ \ x(t) := \begin{bmatrix} x_2(t) \\ x_1(t) \end{bmatrix}. \tag{5.23}$$

The inverse operator, $\hat{R} := \left[I - \hat{D}_{11}^*\hat{D}_{11}\right]^{-1}$, exists whenever $\|\hat{D}_{11}\|_\infty < 1$, which is a necessary condition for the existence of a solution to the H^∞ control problem. When the inverse operator, $\hat{R}$, exists it is the solution to the differential equation

$$\dot{x}(t) = A_Rx(t) + B_Ry(t),\ u(t) = C_Rx(t) + D_Ry(t),\ x_2(\tau) = 0,\ x_1(0) = 0, \tag{5.24}$$

where

$$A_R := \begin{bmatrix} -(A + B_1D_RD_{11}^*C_1)^* & -(C_1^*C_1 + C_1^*D_{11}D_RD_{11}^*C_1) \\ B_1D_RB_1^* & (A + B_1D_RD_{11}^*C_1) \end{bmatrix}, \tag{5.25}$$

$$B_R := \begin{bmatrix} C_1^*D_{11} \\ -B_1 \end{bmatrix}D_R, \tag{5.26}$$

$$C_R := -D_R\begin{bmatrix} B_1^* & D_{11}^*C_1 \end{bmatrix},\ \text{ and }\ D_R := [I - D_{11}^*D_{11}]^{-1}. \tag{5.27}$$

Let $\Gamma(t) := e^{A_Rt}$. If $\Gamma_{11}(\tau)$ is invertible the solution is unique, otherwise any pseudo inverse will give a valid solution. We use $x(t) = e^{A_R(t-t_0)}x(t_0) + \int_{t_0}^t e^{A_R(t-s)}B_Ry(s)ds$

(the variation of constants formula) to solve for the input, $u(t)$, as a function of the output, $y(t)$. Specifically, we get

$$u(t) = C_R \left\{ -e^{A_R t} \begin{bmatrix} \Gamma_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix} \int_0^\tau e^{A_R(\tau - s)} B_R y(s) ds + \int_0^t e^{A_R(t-s)} B_R y(s) ds \right\} + D_R y(t). \quad (5.28)$$

So, by inspection, we have that the kernel $r(t,s)$, of the operator $\hat{R}$, is given by

$$r(t,s) = C_R \left\{ -e^{A_R t} \begin{bmatrix} \Gamma_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix} e^{A_R(\tau - s)} B_R + e^{A_R(t-s)} B_R \mathbf{1}(t-s) \right\} + D_R \delta(t-s). \quad (5.29)$$

The derivation of the other inverse operator, $\hat{Q} := \left[I - \hat{D}_{11} \hat{D}_{11}^* \right]^{-1}$, is quite similar and only the pertinent definitions will be given. The action of the operator $\hat{Q}$ on a function $f(t)$ is given by $\hat{Q} f(t) = \int_0^\tau q(t,s) f(s) ds$, where the kernel $q(t,s)$ is given by

$$q(t,s) = C_Q \left\{ -e^{A_Q t} \begin{bmatrix} \Upsilon_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix} e^{A_Q(\tau - s)} B_Q + e^{A_Q(t-s)} B_Q \mathbf{1}(t-s) \right\} + D_Q \delta(t-s), \quad (5.30)$$

where

$$A_Q := \begin{bmatrix} -(A + B_1 D_{11}^* D_Q C_1)^* & -C_1^* D_Q C_1 \\ (B_1 B_1^* + B_1 D_{11}^* D_Q D_{11} B_1^*) & (A + B_1 D_{11}^* D_Q C_1) \end{bmatrix}, \quad (5.31)$$

$$B_Q := \begin{bmatrix} C_1^* \\ -B_1 D_{11}^* \end{bmatrix} D_Q, \quad (5.32)$$

$$C_Q := -D_Q \begin{bmatrix} D_{11} B_1^* & C_1 \end{bmatrix}, \; D_Q := [I - D_{11} D_{11}^*]^{-1}, \text{ and } \Upsilon(t) := e^{A_Q t}. \quad (5.33)$$

Once again, $q(t,s)$ is a valid representation for $\hat{Q}$ if $\|\hat{D}_{11}\|_\infty < 1$, and it is unique if $\Upsilon_{11}(\tau)$ is invertible. If $\Upsilon_{11}(\tau)$ is singular any pseudo inverse will yield a valid $q(t,s)$.

Remark 5.5.2 *For $D_{11} = 0$, we recover the simple case of [5], i.e. we have*

$$A_R = A_Q = \begin{bmatrix} -A^* & -C_1^* C_1 \\ B_1 B_1^* & A \end{bmatrix}. \tag{5.34}$$

Theorem 5.5.1 (discrete-time equivalent, general case) *If $\|\hat{D}_{11}\|_\infty < 1$ (a necessary condition for solvability), then the finite dimensional state space representation for the discrete-time equivalent problem (5.10) can be calculated directly from:*

$$\acute{A} = \Upsilon_{22}(\tau) - \Upsilon_{21}(\tau)\Upsilon_{11}^{-1}(\tau)\Upsilon_{12}(\tau); \tag{5.35}$$

$$\acute{B}_1 = T_B^* \begin{bmatrix} \Sigma_B^{1/2} \\ 0 \end{bmatrix}, \quad \textit{where} \quad T_B^* \begin{bmatrix} \Sigma_B & 0 \\ 0 & 0 \end{bmatrix} T_B = \Gamma_{21}(\tau)\Gamma_{11}^{-1}(\tau); \tag{5.36}$$

$$\acute{B}_2 = \begin{bmatrix} (\Phi_{21}(\tau) - \Upsilon_{21}(\tau)\Upsilon_{11}^{-1}(\tau)\Phi_{11}(\tau)) \\ (\Phi_{22}(\tau) - \Upsilon_{21}(\tau)\Upsilon_{11}^{-1}(\tau)\Phi_{12}(\tau)) \end{bmatrix}^T \mathcal{B}_Q - \Psi(\tau)B_2, \quad \textit{where} \tag{5.37}$$

$$\mathcal{B}_Q := \begin{bmatrix} -C_1^* D_Q D_{12} \\ B_2 + B_1 D_{11}^* D_Q D_{12} \end{bmatrix}, \quad \textit{and} \quad \textit{let} \quad \mathcal{B}_Q^* =: \begin{bmatrix} \mathcal{B}_Q^R & -\mathcal{B}_Q^L \end{bmatrix}; \tag{5.38}$$

$$\acute{C}_2 = C_2; \quad \textit{and} \quad \begin{bmatrix} \acute{C}_1 & \acute{D}_{12} \end{bmatrix} = \begin{bmatrix} \Sigma_{CD}^{1/2} & 0 \end{bmatrix} T_{CD}, \quad \textit{where} \tag{5.39}$$

$$T_{CD}^* \begin{bmatrix} \Sigma_{CD} & 0 \\ 0 & 0 \end{bmatrix} T_{CD} = \begin{bmatrix} \mathcal{C}_{11} & \mathcal{C}_{12} \\ \mathcal{C}_{12}^* & \mathcal{C}_{22} \end{bmatrix}, \quad \mathcal{C}_{11} := -\Upsilon_{11}^{-1}(\tau)\Upsilon_{12}(\tau), \tag{5.40}$$

$$\mathcal{C}_{12} := -\Upsilon_{11}^{-1}(\tau) \begin{bmatrix} \Phi_{11}(\tau) & \Phi_{12}(\tau) \end{bmatrix} \mathcal{B}_Q, \quad \textit{and} \tag{5.41}$$

$$\mathcal{C}_{22} := \begin{bmatrix} \mathcal{B}_Q^L & \mathcal{B}_Q^R \end{bmatrix} \left\{ \Phi(\tau) \begin{bmatrix} \Upsilon_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix} \Phi(\tau) - \Omega(\tau) \right\} \mathcal{B}_Q + \tau D_{12}^* D_Q D_{12}; \tag{5.42}$$

$$\textit{and} \quad \Phi(t) := \int_0^t \Upsilon(s)ds, \ \Psi(t) := \int_0^t e^{As}ds, \ \textit{and} \ \Omega(t) := \int_0^t \Phi(r)dr. \tag{5.43}$$

The proof is long and involved, provides little insight into the resulting formulas, and may be omitted without loss of continuity. Appendix D contains a detailed sketch of the proof, which may be of independent interest to researchers in related areas.

Remark 5.5.3 *For $D_{11} = 0$ and $D_{12} = 0$, we recover precisely the equations in [5].*

The computation of the discrete-time equivalent system from the sampled data problem parameters is quite straight forward.

Remark 5.5.4 *The integrals involving the matrix exponentials can, in theory, be computed using any symbolic math package such as Maple or Mathematica, however for large matrices the computation time/memory requirements may become excessive.*

If A is well conditioned, $\Psi(\tau)$ can be computed directly from $\Psi(\tau) = A^{-1}(e^{A\tau} - I)$. Similarly, if A_Q is well conditioned, then $\Phi(\tau)$ and $\Omega(\tau)$ can be computed directly from $\Phi(\tau) = A_Q^{-1}(e^{A_Q\tau} - I)$ and $\Omega(\tau) = A_Q^{-1}(\Phi(\tau) - \tau I)$.

Remark 5.5.5 *Numerical evaluation of matrix exponentials is inherently approximate, and the approximation error may be quite significant, see [60].*

All the rest of the calculations are simple linear algebra, except for the factorizations required to calculate $\acute{B}_1$, $\acute{C}_1$, and $\acute{D}_{12}$. However, subject to the associated potential numerical difficulties, $\acute{B}_1$, $\acute{C}_1$, and $\acute{D}_{12}$ can be calculated using the singular value decompositions as indicated by the formulas in Theorem 5.5.1.

Remark 5.5.6 *If $\acute{B}_1$ is a vector, then it can be calculated exactly by taking the square roots of the diagonal elements of $\Gamma_{21}(\tau)\Gamma_{11}^{-1}(\tau)$.*

Similarly, if $\begin{bmatrix} \acute{C}_1 & \acute{D}_{12} \end{bmatrix}$ is a vector, then it can also be calculated exactly. For more on the numerical difficulties associated with use of the singular value decomposition see Appendix C.

Remark 5.5.7 *The resulting equivalent discrete-time system is the same order as the original continuous-time system, which is a property shared by the more complex characterization of [46].*

We have not as yet been able to successfully complete a numerical example verifying that the discrete-time equivalent system of Theorem 5.5.1 is indeed equivalent with respect to the infinity-norm. In fact, as yet we have not even been able to do such an example even for the simple case of [5], nor have the authors. It is not yet clear whether this is due to numerical difficulties or a flaw in the development.

Remark 5.5.8 *The use of the noncausal inverse lifting operator, W_τ^{-1} (5.2), eliminates from the formulas, for the discrete-time equivalent, all of the observability and controllability grammians (see e.g. [12]) normally associated with problems involving delays.*

This is a very nice feature of Bamieh and Pearson's formulation [5], however it does make one wonder whether we have simplified things to an unrealistic extent.

Remark 5.5.9 *It is precisely the presence of grammian type, and other similar, integrals involving products of matrix exponentials that make the discrete-time equivalent of Kabamba and Hara [46] so very difficult to compute.*

With respect to the sampled-data repetitive control problem, it is irrelevant *how* the discrete-time equivalent system is computed. Thus, the sampled-data repetitive control problem solution outlined in this chapter, does not depend on how these issues with respect to computation of discrete-time equivalents are ultimately resolved.

5.6 Singularity of the Discrete-Time Equivalent Problem

Regardless of what method is used to obtain the discrete-time equivalent system, the resulting discrete-time equivalent problem will be the "singular problem", see e.g. [87]. This is an inherent property of well posed sampled-data problems, due to the requirement for all sampled signals to be low pass. The significance of the singular problem is that it cannot be solved by the nice state space methods of [24] for the continuous-time case or [44, 86] for the discrete-time case. There have been several attempts to extend these results to the singular case, see e.g. [85, 78]. None of these methods has evolved to point where their solutions are readily implementable, especially for the discrete-time case. The classical frequency domain based approach, see e.g. [33], has no added difficulty dealing with the singular problem, but there is a problem with obtaining the required factorizations for the discrete-time case. The problem is with the solution of the associated discrete-time algebraic Riccati equation (DARE), see [33, 21, 48]. Considerable work has been done towards a solution to the required DARE [11, 88], but the only closed form solution is obtained by transforming the DARE to an equivalent continuous-time algebraic Riccati equation (CARE) via the bilinear transform and then converting the solution back via the inverse transform [11]. There is a relatively new area of control theory with application to the singular H^∞ problem based on linear matrix inequalities (LMI's), see e.g. [36, 45, 9]. This approach does not have any added difficulties for the singular problem, however most of the work has been for continuous-time systems. Furthermore, the parameterization of all solutions is not as attractive for our purposes as the one in terms of the so called

"central controller" as provided by [24].

A more direct method for purely H^∞ problems is the use of the bilinear transform. The bilinear transform of a singular problem is, with probability one, a nonsingular problem. Thus, the bilinear transformation of the discrete-time equivalent problem yields an equivalent continuous time problem, which satisfies the standard assumptions [24]. The equivalence is with respect to the infinity norm [37], thus providing complete equivalence in this case. There is no loss in design insight, since the true design requirements have already been posed in the sampled-data setting. The resulting standard continuous time equivalent problem can be readily solved using the state space methods of [24] as implemented in MATLAB's μ-tools toolbox [3].

5.7 Continuous-time Equivalent System

Specifically, we convert the discrete-time equivalent problem, $\acute{G}$, to a continuous-time equivalent problem, $G^s = (A^s, B^s, C^s, D^s)$, solve for the corresponding continuous-time controller, $K^s = (A_K^s, B_K^s, C_K^s, D_K^s)$, and convert the controller to its discrete-time equivalent, $K = (A_K, B_K, C_K, D_K)$. The formulas are given by:

$$A^s := \alpha^{-1}(I + \acute{A})^{-1}(\acute{A} - I), \tag{5.44}$$

$$B^s := (I + \acute{A})^{-1}\acute{B}\sqrt{2/\alpha}, \tag{5.45}$$

$$C^s := \acute{C}(I + \acute{A})^{-1}\sqrt{2/\alpha}, \text{ and} \tag{5.46}$$

$$D^s := \acute{D} - \acute{C}(I + \acute{A})^{-1}\acute{B}; \tag{5.47}$$

and

$$A_K := (I - \alpha A_K^s)^{-1}(I + \alpha A_K^s), \tag{5.48}$$

$$B_K := (I + A_K)B_K^s\sqrt{\alpha/2}, \tag{5.49}$$

$$C_K := \sqrt{2}C_K^s(I + A_K)\sqrt{\alpha/2}, \text{ and} \tag{5.50}$$

$$D_K := D_K^s + C_K(I + A_K)^{-1}B_K; \tag{5.51}$$

where $\alpha > 0$ is the bilinear mapping parameter. It is easily verified that the above bilinear transformations are inverses of each other. The first one maps from the z-domain to the s-domain via $s = \alpha^{-1}(z-1)/(z+1)$, such that the unit disk is mapped to the left half plane, and the second one maps from the s-domain to the z-domain via $z = (1+\alpha s)/(1-\alpha s)$. For more on state space to state space bilinear transforms see [37, 62].

Remark 5.7.1 *The resulting discrete-time controller, $K = (A_K, B_K, C_K, D_K)$, is precisely the controller, $C = [C_1,\ C_2]$, for the sampled-data problem of Figure 31.*

5.8 Recovering a Repetitive Controller Structure

The solution provided by MATLAB's μ-tools toolbox is the so called "central controller", K^c, and all controllers satisfying $\|\mathcal{F}(G^s, K^s)\|_\infty < \gamma$ are parameterized by

$$K^s = \mathcal{F}(K^s, M) := K_{11}^c + K_{12}^c M(I - K_{22}^c M)^{-1}K_{21}^c, \tag{5.52}$$

where $\|M\|_\infty < \gamma$. While the continuous-time repetitive performance requirement, as represented by the weighting function, W_1, may not yield a central controller with a digital repetitive structure, it should be "close" to repetitive. Furthermore, it is reasonable to expect that, for appropriately selected weighting functions, there will exist a minimal order (same order as the central controller) solution that has a digital

repetitive structure. A large class of minimal order controllers is parameterized by all constant matrices M, such that $\|M\|_\infty < \gamma$.

Remark 5.8.1 *Since all of the controllers parameterized by (5.52) satisfy the robustness and tracking requirements of the H^∞ optimal formulation, we are only concerned with how repetitive (easy to implement) the controllers are.*

Since there are several free parameters and the problem is highly nonconvex, traditional search algorithms are not suitable. However, since all solutions are admissible (satisfy the continuous-time requirement imposed by the H^∞ formulation), the new search technique known as genetic algorithms is ideally suited to the task.

5.8.1 Genetic algorithms

Genetic algorithms (GA's) are so named because they are based on concepts arising in natural selection, see [42, 38, 83] and references therein for background. GA's have proven to be very effective at finding global, or "near global", maxima for highly nonconvex optimization problems. The basic concept is as follows:

1. Identify the set of free variables (design parameters), which the GA will manipulate as it searches for maxima.

2. Choose an appropriate *fitness* measure (objective function) such that the fitness value is larger for "better" sets of design parameters.

3. Generate an initial *population* of suitable (admissible) sets of design parameters.

4. Apply selection (based on fitness) in conjunction with the "crossover" and "mutation" operations, see e.g. [42, 38, 83], to the current population, to construct a new population of admissible sets of parameters (the next generation).

Step (4) is repeated until "satisfactory" results are obtained. If the fitness measure, crossover and mutation probabilities, and population size are all appropriately chosen, then the algorithm will likely converge to a (hopefully global) maximum. In our case we don't necessarily need to find the global maximum, or even converge; we just want to obtain controllers that are "more" repetitive. Obviously, if a generated "candidate" set of parameters for the next generation is not admissible, i.e. $\|M\|_\infty \geq \gamma$, then it must be either discarded or adjusted (rescaled). GA's have been used successfully to obtain satisfactory solutions to various difficult optimization problems related to control systems, see [47, 101, 52, 106] and references therein.

5.8.2 Repetitive structure via GA's

We propose to use a GA, which is a slight modification of the one used in [54], to find controllers K^s such that the bilinear transform, K, of K^s has a digital repetitive structure. The design parameters are the entries of the constant matrix M. A set of parameters (matrix entries) is admissible if $\|M\|_\infty < \gamma$. To determine a fitness measure that reflects "how repetitive" the corresponding digital controller is, we must take a closer look at the structure of digital repetitive controllers. Let $d(z) =: N_d/D_d$ and $q(z) =: N_q/D_q$, where N_d, D_d, N_q, and D_q are polynomials in z^{-1}. Now, we can

write

$$K(z) = \frac{d(z)}{1 - q(z)z^{-L}} = \frac{N_d D_q}{D_d D_q - N_q D_d z^{-L}} =: \frac{N_K}{D_K}. \tag{5.53}$$

Let the orders of the polynomials be given by $m := ord(D_d D_q)$, $p := ord(N_q D_d)$ and $n := ord(D_K)$. Define the coefficients a_i by $D_K = a_0 + a_1 z^{-1} + \cdots + a_n z^{-n}$. In general (and in particular for proper chosen performance weights W_1) we have $n = p + L$. If $K(z)$ has a repetitive structure, then there will be a band of zero coefficients, i.e. $a_i = 0, \;\; m+1 < i < L-p-1$. Let the normalized coefficients of interest be given by $\hat{a}_j := a_i/a_n, \;\; j := i-m-1, \;\; m+1 < i < L-p-1$. Define the *degree of repetitiveness* by $r := L - m - p - 2$. For a given r, let the fitness function be given by

$$f_r(M, \beta) := \left(\beta + \sum_{j=1}^{r} |\hat{a}_j| \right)^{-1}, \tag{5.54}$$

where the parameter $\beta > 0$ determines the value, β^{-1}, at which the fitness function saturates. Ideally, the GA utilizing the fitness function f_r would seek to maximize the degree of repetitiveness, r. Choosing β to be inversely proportional to r would seem a logical choice. The only difficulty lies in choosing a $\beta(r)$ such that the algorithm converges to $\beta(r)^{-1}$ for some (hopefully maximal) r. In the absence of such a $\beta(r)$, the GA can be used for each feasible r separately using some appropriate constant $\beta > 0$.

While the fitness function $f_r(M, \beta)$ has intuitive appeal, it may be susceptible to numerical difficulties, for high order $K(z)$, due to the dependence on very precise values of the polynomial coefficients, a_i. An alternative formulation with potentially better numerical properties is based on trying to make the controller, K, have L

poles as close as possible to the unit circle. Recall that in the ideal case a digital repetitive controller would have L poles *on* the unit circle. Such a criterion will not directly enforce a digital repetitive structure, unless the ideal case is achieved. However, there are no numerical problems and it is computationally efficient. Let $P_i = x_i + jy_i, \quad i = 1, \ldots, L$ be the L poles of K^s (eigenvalues of A_K^s) closest to the imaginary ($j\omega$) axis. Denote the normalized poles by $\hat{p}_i := x_i/y_i + j$ and define the fitness function

$$f(M, \delta) = \left(\delta + \sum_{i=1}^{L} |\hat{p}_i| \right)^{-1}, \tag{5.55}$$

where the constant parameter $\delta > 0$ determines the saturation value for f just as β did for f_r. The relative merits of the two fitness functions, $f_r(M, \beta)$ and $f_r(M, \delta)$, is a topic for further research upon resolution of the problems with calculating a discrete-time equivalent system.

5.9 Conclusion

In this chapter we proposed that sampled-data repetitive controllers be defined to be those controllers having a digital repetitive structure and satisfying a continuous-time performance requirement (tracking/disturbance rejection), see Definition 5.4.1. We posed a two degree of freedom H^∞ optimal sampled-data tracking performance with robust stability problem, see Figure 31, directly satisfying the second part of Definition 5.4.1. We extended the results of [5], on discrete-time equivalents, to the general case required to solve our H^∞ problem formulation. We also noted that numerical computation of discrete-time equivalents has not yet been resolved for either

the approach of [5] or the approach of [46]. Whatever method eventually proves to be practically applicable to computing discrete-time equivalents, the resulting equivalent problem will be singular. We gave explicit state-space formulas (5.44) through (5.47) for converting the resulting singular discrete-time equivalent problem to an equivalent continuous-time problem satisfying the standard assumptions. There is no loss in design insight due to this conversion, since the actual problem has already been posed in the sampled-data setting. The resulting standard continuous-time equivalent problem can be solved using MATLAB's μ-tools toolbox [3]. The resulting parameterization of all solutions in terms of the central controller can then be exploited to recover a digital controller having a repetitive structure. The corresponding digital repetitive controller can be calculated from the resulting continuous-time controller using the state-space formulas (5.48) through (5.51). We posed the problem of searching the parameterization of all solutions for the "most repetitive" solution in terms of genetic algorithms. With appropriately selected weighting functions, see Subsection 5.4.1, all solutions should correspond to digital controllers having "nearly" repetitive structures. It is expected that controllers with digital repetitive structure will be readily obtained. Numerical investigations await the resolution of the problems with calculating discrete-time equivalent systems.

CHAPTER VI

Contributions and Future Directions

This dissertation has made several contributions to repetitive control theory and related areas, and each contribution has suggested at least one direction for future research. These contributions and directions were obtained by applying H^∞ optimal control theory to repetitive controller design. Further contributions and new directions were obtained by the addition of sampled-data system theory to the mix. By applying H^∞ optimal control theory to the repetitive control design problem (quantifying the trade-off between performance and robustness) new insight into the repetitive control design problem was obtained. The new insight gained by applying H^∞ optimal control included directly applicable design procedures and new approaches to "classical" (designer based) control design. The continuous-time formulation of the robust performance problem of Chapter III suggests possible new approaches to classical repetitive controller design and holds the promise, subject to the availability of more powerful computers, of being a directly applicable design technique. Chapter III also provides a significant extension to existing H^∞ optimal control theory for infinite dimensional systems and raises issues with respect to obtaining exact minimal realizations of transfer function matrices. The issues with respect to obtaining exact minimal realizations of transfer function matrices are discussed, and partially

resolved, in Appendix C. The continuous-time formulation of the nominal performance with robust stability problem of Chapter IV provides both multiple direct design procedures and multiple new approaches to classical design. Furthermore, the resulting new direct design procedures can be carried out, with existing computational capabilities, for practical applications. In Chapter V we add sampled-data system theory to the mix and propose a precise definition of sampled-data repetitive control, which is a natural combination of the practical requirements that give rise individually to sampled-data theory and repetitive control theory. We formulate a nominal performance with robust stability H^∞ optimal sampled-data control problem and a procedure for recovering controllers satisfying the natural definition of sampled-data repetitive control. Also in Chapter V, we extend existing sampled-data theory on obtaining discrete-time equivalent systems. The H^∞ optimal sampled-data repetitive control formulation of Chapter V holds great promise for direct design and the potential for insight into classical design, but realization depends on the resolution of the present difficulties with calculating discrete-time equivalent systems. As is always the case in repetitive control design, we considered two degree of freedom control (i.e. two distinct controllers) throughout this dissertation.

In Chapter III we pose a robust performance H^∞ optimal control problem, see Definition 3.1.1. By constraining the controller C_1 to have a repetitive structure we obtain the generalized repetitive structure of Figure 8, which corresponds uniquely to the solution of the constrained problem. This generalized structure has obvious implications for classical repetitive control design, which should be the subject of

further research. The constrained problem has the form of a vector H^∞ problem with a scalar infinite dimensional inner factor (e.g. a delay). New H^∞ optimal control theory had to be developed to solve this class of problems. The solution to this class of vector H^∞ optimal control problems is given by Theorem 3.2.1. The extensive derivation and proof of Theorem 3.2.1 appears in Appendix B. The extension of Theorem 3.2.1 to vector problems of higher dimensions is obvious. The extension to non-scalar infinite dimensional inner factors is not obvious, but the mathematical techniques used in the derivation should prove helpful.

In Chapter IV we formulate a nominal repetitive performance with robust stability design problem. The design problem is decoupled into a finite dimensional design which stabilizes and approximately inverts the nominal plant and a two-block (nominal performance with robust stability) H^∞ optimal control problem for an infinite dimensional equivalent plant. The decoupling of control design in this way is a common technique when designing for performance and the limits to the decoupling are addressed by Lemma 4.1.1 and Lemma 4.1.2. The finite dimensional design to stabilize and approximately invert the nominal plant is fairly well understood and is not the subject of this dissertation. The resulting infinite dimensional two-block H^∞ optimal control problem is solved using a special case of the general theory of [95]. The solution to the special case is given by Theorem 4.2.1. This formulation leads to the modified repetitive controller structure of Figure 17, where the novel feature is the inclusion of the inner factor, $m_p(s)$, corresponding to the RHP zeros of non-minimum phase plants. This new repetitive controller structure indicates a path for

future research into classical repetitive control design for non-minimum phase plants. Using properly selected weighting functions, see Section 4.6, the resulting controllers admit low order approximations providing classical repetitive action with the modified repetitive structure of Figure 17, for direct designs. By formulating a sensitivity improvement problem, see Definition 4.3.3, the very same solution has an alternative interpretation as the add-on portion of the new cascade repetitive structure of Figure 19. This cascade repetitive structure can, in turn, be thought of as either an add-on to an existing controller or a method for robustly cascading two repetitive controllers. Once again, this has implications for future research on classical repetitive control design, i.e. research into the classical design of cascade repetitive controllers. Furthermore, the numerical results for the direct design are most promising and show direct applicability to practical repetitive controller design using existing computational facilities. Finally, this formulation can be extended to plants with significant nonlinearities by the addition of dynamic inversion control theory.

In Chapter V we consider H^∞ optimal sampled-data repetitive control, where by sampled-data we mean that the discrete-time controller will be designed directly against continuous-time requirements. A natural definition for sample-data repetitive controller design, see Definition 5.4.1, is the direct design of discrete-time controllers, with digital repetitive structures, satisfying continuous-time requirements. No previous work on sampled-data repetitive control satisfies this definition. Our procedure consist of two steps: obtaining the parameterization of all solutions to a repetitive performance with robust stability H^∞ optimal sampled-data problem, and the

subsequent extraction of digital repetitive controllers from the parameterization of all solutions. The solution of the repetitive performance with robust stability H^∞ optimal sampled-data problem requires the computation of a discrete-time equivalent problem, from the continuous-time plant and requirements, as represented by the weighting functions. In an effort to accomplish this we extend the results of [5] to the general case, see Theorem 5.5.1. However, accurate numerical computation of discrete-time equivalents has not been adequately resolved. The parameterization of all solutions to the discrete-time equivalent problem is obtained by taking a bilinear transform and using the state space solution of [24] for the resulting standard problem. The problem of searching for the "most repetitive" solution is then formulated in terms of genetic algorithms. Genetic algorithms are a powerful new search technique that performs very well for highly nonconvex problems. All problems related to controller structure are highly non-convex [8]. The novel use of genetic algorithms to search parameterizations, of all controllers satisfying some norm-based optimization problem, for controllers possessing additional desirable properties, has great promise for a broad range of control applications. While our H^∞ optimal sampled-data repetitive control formulation cannot be fully evaluated until the problems with respect to computation of the discrete-time equivalent problem are resolved, the formulation appears to have the potential to be a practical and highly effective design procedure.

Appendix A

Notation

C: Set of complex numbers (complex plane).

C_+: Closed right half plane, $C_+ = \{s \in C : Re(s) \geq 0\}$.

C_-: Open left half plane, $C_- = \{s \in C : Re(s) < 0\}$.

$\mathcal{P}(R)$: Set of Poles of the rational function R.

$\mathcal{P}_+(R)$: Unstable poles of R, $\mathcal{P}_+(R) := \{p \in \mathcal{P}(R) : Re(p) \geq 0\}$.

$\mathcal{P}_-(R)$: Stable poles of R, $\mathcal{P}_-(R) := \{p \in \mathcal{P}(R) : Re(p) < 0\}$.

L^2: Hilbert space of square integrable functions on $Re(s) = 0$.

H^2: L^2 functions which admit analytic extension to C_+.

H_n^2: $n \times 1$ vector valued functions with elements in H^2.

$H_2^\perp$: Orthogonal complement of H^2 in L^2, $H_2^\perp := L^2 \ominus H^2$.

L^∞: Banach space of functions essentially bounded on $Re(s) = 0$.

H^∞: Essentially bounded analytic functions on C_+.

IRH^∞: Rational H^∞ functions with real coefficients.

$MH^2 := \{Mw : w \in H_n^2\}$, where M is a $n \times n$ inner matrix.

$H(M)$: Orthogonal complement of MH^2, $H(M) = H_n^2 \ominus MH^2$.

$\mathbf{P}_+$: Orthogonal projection from L^2 to H^2.

$\mathbf{P}_-$: Orthogonal projection from L^2 to $H_2^\perp$.

$\mathbf{P}_{H(M)}$: Orthogonal projection from L^2 to $H(M)$.

$\mathbf{M}_W$: Multiplication operator, $\mathbf{M}_W x := Wx$, for $x \in H^2$, $W \in L^\infty$.

$\mathbf{\Upsilon}_F$: Toeplitz operator with symbol F, $\mathbf{\Upsilon}_F = \mathbf{P}_+\mathbf{M}_F$,

$\mathbf{\Gamma}_F$: Hankel operator with symbol F, $\mathbf{\Gamma}_F = \mathbf{P}_-\mathbf{M}_F$,

$\mathbf{A}^*$: Adjoint of the operator $\mathbf{A}$.

F^T: Transpose of the matrix F.

F^*: Complex conjugate (Hermitian) transpose of the matrix F.

$R(s)^*$: "Star" of the matrix $R(s)$, $R(s)^* := R^T(-s)$.

$ord(R)$: Degree of the least common denominator of the matrix R.

$L^p[0,\infty)$: $\{f(t) : \int_0^\infty |f(t)|^p dt < \infty\}$.

$L_e^p[0,\infty)$: $\left\{f(t) : \int_0^T |f(t)|^p dt < \infty, \ \forall\, T < \infty\right\}$.

$L^p[0,\tau]$: $\{f(t) : \int_0^\tau |f(t)|^p dt < \infty\}$.

$l_{L^p[0,\tau]}$: The space of infinite sequences with elements in $L^p[0,\tau]$.

$l_{\mathbb{R}^n}$: The space of infinite sequences with elements in $\mathbb{R}^n$, where x is an n-vector.

$\mathbf{1}(t)$: The unit step function.

$\delta(t)$: The unit impulse function.

$\|F\|_\infty := \sup\{|F(s)| \ : \ Re(s) > 0\} = ess\sup_{\omega\in\mathbb{R}}\{|F(j\omega)|\}$.

$ess\sup$: The essential (almost everywhere) supremum (generalized maximum).

$\|D\|$: Operator norm (largest singular value) of the matrix D.

$\|\mathbf{A}\|$: Operator norm, square root of the largest value of the spectrum of $\mathbf{A}$.

$\sigma(\mathbf{A})$: The spectrum on the operator $\mathbf{A}$.

σ_d: The discrete spectrum (isolated singularities or singular-values) of an operator.

σ_d: The essential spectrum ($\sigma = \sigma_d \bigcup \sigma_e$) of an operator.

$\|\mathbf{A}\|_e$: Essential norm, square root of the largest value of $\sigma_e(\mathbf{A})$.

$\|f(t)\|_2 := \{\int_0^\infty |f(t)|^2 dt\}^{1/2}$.

$a \bigvee b := \max\{a, b\}$.

$|S|$: Magnitude, or absolute value, of the possibly complex valued function S.

RHP: Right half (complex, $s = \sigma + j\omega$) plane.

LTI: Linear time-invariant.

LCD: Least common denominator.

SISO: Single input single output (system).

SIMO: Single input multiple output (system).

MIMO: Multiple input multiple output (system).

SVD: Singular value decomposition.

Appendix B

Derivation and Proof of Theorem 3.2.1

In this appendix we derive Theorem 3.2.1, thus proving necessity, and then prove sufficiency. We also give explicit formulas for constructing R_σ and the singular-vector, x, corresponding to any σ which makes R_σ. Throughout this appendix we make free use, without comment, of the definitions given in Appendix A.

B.1 Singular-value/Singular-vector Problem

We start by stating the first genericity assumption, which allows us to find the unique optimal performance, γ_{opt}, by solving the singular-value/singular-vector problem for the operator **A**.

Assumption 1 (genericity) $\gamma_{opt} = \|\mathbf{A}\| > \|\mathbf{A}\|_e = max\{\|V_3\|_\infty, \|D\|\}$.

Assumption 1 is satisfied for a large class of practical problems. If it is not satisfied, then the optimal interpolant,$\tilde{Q}^{opt}$, is not unique.

Remark B.1.1 *Since $\gamma_{opt} = \|\mathbf{A}\|$, from (3.15), we see that $\|\mathbf{A}\| = \|V_3\|_\infty$ iff $R_1 = R_2 = 0$. This follows because R_1 and R_2 are rational and $\tilde{Q}_1, \tilde{Q}_2 \in H^\infty$ and therefore cannot invert m.*

Under Assumption 1, we have $\|\mathbf{A}\| = \sigma_{max}$, where $\sigma_{max} := max\{\sigma : \sigma^2 \in \sigma_d(\mathbf{A})\}$ is the largest singular-value of $\mathbf{A}$. Hence, we consider the singular-value /singular-vector problem for the operator $\mathbf{A}$, i.e. $0 \neq x \in H^2$ is a singular-vector iff there exists $\sigma \geq 0$, such that $(\sigma^2 I - \mathbf{A}^*\mathbf{A})\, x = 0$, which can be written as (see Definition 3.2.1)

$$\sigma^2 x - \mathbf{P}_+ \mathbf{M}_{V_3}^* \mathbf{M}_{V_3} x - \left[\begin{array}{cc} \mathbf{M}_{V_1}^* & \mathbf{M}_{V_2}^* \end{array}\right] \mathbf{P}_{H(M)} \left[\begin{array}{c} \mathbf{M}_{V_1} \\ \mathbf{M}_{V_2} \end{array}\right] x = 0. \tag{B.1}$$

From the fact that $\mathbf{M}_W^* = \mathbf{P}_+ \mathbf{M}_{W^*}$, we have

$$\sigma^2 x - \mathbf{P}_+ V_3^* V_3 x - \mathbf{P}_+ \left[\begin{array}{cc} V_1^* & V_2^* \end{array}\right] \mathbf{P}_{H(M)} \left[\begin{array}{c} V_1 \\ V_2 \end{array}\right] x = 0. \tag{B.2}$$

Remark B.1.2 *Let $w \in H^2$, then $\mathbf{P}_{H(M)} w = w - \mathbf{P}_{MH^2} w = w - M\mathbf{P}_+ M^* w$.*

Thus, we have

$$\begin{aligned} \mathbf{P}_{H(M)} \left[\begin{array}{c} V_1 \\ V_2 \end{array}\right] x &= \left[\begin{array}{c} V_1 \\ V_2 \end{array}\right] x - M\mathbf{P}_+ M^* \left[\begin{array}{c} V_1 \\ V_2 \end{array}\right] x \\ &= M\left(M^* \left[\begin{array}{c} V_1 \\ V_2 \end{array}\right] x - \mathbf{P}_+ M^* \left[\begin{array}{c} V_1 \\ V_2 \end{array}\right] x \right) \\ &= M\mathbf{P}_- M^* \left[\begin{array}{c} V_1 \\ V_2 \end{array}\right] x = M\mathbf{P}_- \left[\begin{array}{c} R_1 \\ R_2 \end{array}\right] m^* x. \end{aligned} \tag{B.3}$$

Finally, we substitute back into (B.2), and get

$$\sigma^2 x - \mathbf{P}_+ V_3^* V_3 x - \mathbf{P}_+ \left[\begin{array}{cc} R_1^* & R_2^* \end{array}\right] m\mathbf{P}_- \left[\begin{array}{c} R_1 \\ R_2 \end{array}\right] m^* x = 0. \tag{B.4}$$

Remark B.1.3 *Thus the singular-value/singular-vector problem has a Hankel plus Toeplitz form, see e.g. [33],*

$$(\sigma^2 \mathbf{I} - \Upsilon_{V_3}^* \Upsilon_{V_3} - \Gamma_{m^*\tilde{R}}^* \Gamma_{m^*\tilde{R}}) x = 0, \quad \text{where} \quad \tilde{R} := \left[\begin{array}{c} R_1 \\ R_2 \end{array}\right]. \tag{B.5}$$

The problem is now in the form of a singular-value/singular-vector problem that can be solved using basic projection operations.

B.2 Finite Rank Projection Equations

From the first two terms of (B.4), we have

$$\sigma^2 x - \mathbf{P}_+ V_3^* V_3 x = (\sigma^2 \mathbf{I} - \mathbf{P}_+ V_3^* V_3) x = \mathbf{P}_+ (\sigma^2 I - V_3^* V_3) x. \tag{B.6}$$

By Assumption 1, $\sigma_{max} > \|V_3\|_\infty$, so for any σ of interest $(\sigma^2 I - V_3^* V_3)$ is nonsingular. Thus, the spectral factorization,

$$(\sigma^2 I - V_3^* V_3) = G_\sigma^{-*} G_\sigma^{-1}, \qquad G_\sigma, G_\sigma^{-1} \in H^\infty, \tag{B.7}$$

is well defined. In operator notation, we have

$$(\sigma^2 \mathbf{I} - \mathbf{M}_{V_3}^* \mathbf{M}_{V_3}) = \mathbf{M}_{G_\sigma^{-1}}^* \mathbf{M}_{G_\sigma^{-1}}. \tag{B.8}$$

Now we define a new vector y, related to the singular-vector x, by

$$y := \mathbf{M}_{G_\sigma^{-1}} x = G_\sigma^{-1} x \in H^2 \Rightarrow x = G_\sigma y. \tag{B.9}$$

Then, we substitute (B.9) into (B.4) and multiply by $\mathbf{M}_{G_\sigma}^*$ from the left, which gives

$$y - \mathbf{M}_{G_\sigma}^* \mathbf{P}_+ \begin{bmatrix} R_1^* & R_2^* \end{bmatrix} m \mathbf{P}_- \begin{bmatrix} R_1 \\ R_2 \end{bmatrix} m^* G_\sigma y = 0. \tag{B.10}$$

In order to simplify this expression, we note that

$$\mathbf{M}_{G_\sigma}^* \mathbf{P}_+ \begin{bmatrix} \mathbf{M}_{R_1^*} & \mathbf{M}_{R_2^*} \end{bmatrix} = \mathbf{M}_{G_\sigma}^* \begin{bmatrix} \mathbf{M}_{R_1}^* & \mathbf{M}_{R_2}^* \end{bmatrix} = \mathbf{P}_+ \mathbf{M}_{G_\sigma^*} \begin{bmatrix} \mathbf{M}_{R_1^*} & \mathbf{M}_{R_2^*} \end{bmatrix}, \tag{B.11}$$

and make the following definitions

$$\hat{R} := \begin{bmatrix} \hat{R}_1 \\ \hat{R}_2 \end{bmatrix} := \begin{bmatrix} R_1 G_\sigma \\ R_2 G_\sigma \end{bmatrix} = G_\sigma \tilde{R} \quad \text{and} \quad \hat{V} := \begin{bmatrix} \hat{V}_1 \\ \hat{V}_2 \end{bmatrix} := \begin{bmatrix} V_1 G_\sigma \\ V_2 G_\sigma \end{bmatrix}. \tag{B.12}$$

Thus, we have

$$y - \mathbf{P}_+\hat{R}^* m \mathbf{P}_- \hat{R} m^* y = 0. \tag{B.13}$$

This equation is very hard (if not impossible) to directly solve for y, since it contains infinite rank projections. Therefore, we must simplify it by expanding it into computable terms (finite rank projections). We define an orthogonal decomposition of the vector y, by

$$y = u + mv, \quad u \in H(m) \subseteq H^2, \quad v \in H^2 \Rightarrow mv \in mH^2 \subseteq H^2. \tag{B.14}$$

Now, we expand (B.13) into computable terms. Starting with the rightmost projection operation, we have

$$\mathbf{P}_-\hat{R}m^*y = \mathbf{P}_-\hat{R}m^*u + \mathbf{P}_-\hat{R}v = \hat{R}(m^*u) - \mathbf{P}_+\hat{R}(m^*u) + \mathbf{P}_-\hat{R}v. \tag{B.15}$$

Then,

$$\begin{aligned}
\mathbf{P}_+\hat{R}^* m(\mathbf{P}_-\hat{R}m^*y) &= \mathbf{P}_+\hat{R}^* m\hat{R}(m^*u) - \mathbf{P}_+\hat{R}^* m\mathbf{P}_+\hat{R}(m^*u) + \mathbf{P}_+\hat{R}^* m\mathbf{P}_-\hat{R}v \\
&= \mathbf{P}_+\hat{R}^*\hat{R}u - \hat{R}^* m\mathbf{P}_+\hat{R}(m^*u) + \mathbf{P}_-\hat{R}^* m\mathbf{P}_+\hat{R}(m^*u) \\
&\quad + \hat{R}^* m\mathbf{P}_-\hat{R}v - \mathbf{P}_-\hat{R}^* m\mathbf{P}_-\hat{R}v \\
&= \hat{R}^*\hat{R}u - \mathbf{P}_-\hat{R}^*\hat{R}u - \hat{R}^* m\mathbf{P}_+\hat{R}(m^*u) + \mathbf{P}_-\hat{R}^* m\mathbf{P}_+\hat{R}(m^*u) \\
&\quad + \hat{R}^* m\mathbf{P}_-\hat{R}v - \mathbf{P}_-\hat{R}^* m\mathbf{P}_-\hat{R}v.
\end{aligned} \tag{B.16}$$

We define $u_\perp := m^*u \in H_2^\perp$ and substitute into (B.13), which gives

$$y = \hat{R}^*\hat{R}u - \mathbf{P}_-\hat{R}^*\hat{R}u - \hat{R}^* m\mathbf{P}_+\hat{R}u_\perp + \mathbf{P}_-\hat{R}^* m\mathbf{P}_+\hat{R}u_\perp$$

$$+\hat{R}^* m\mathbf{P}_- \hat{R}v - \mathbf{P}_- \hat{R}^* m\mathbf{P}_- \hat{R}v. \tag{B.17}$$

Now we take the projections of y onto the orthogonal subspaces mH^2 and $H(m)$, to get equations for v and u, i.e. $v = m^*\mathbf{P}_{mH^2}y = \mathbf{P}_+ m^* y$ and $u = \mathbf{P}_{H(m)}y = y - mv$, which gives

$$v = \mathbf{P}_+ \hat{R}^* \hat{R}u_\perp - \mathbf{P}_+ \hat{R}^* \mathbf{P}_+ \hat{R}u_\perp + \mathbf{P}_+ \hat{R}^* \mathbf{P}_- \hat{R}v, \tag{B.18}$$

and

$$\begin{aligned}(1 - \hat{R}^*\hat{R})u = &-\mathbf{P}_- \hat{R}^* \hat{R}u - \hat{R}^* m\mathbf{P}_+ \hat{R}u_\perp + \mathbf{P}_- \hat{R}^* m\mathbf{P}_+ \hat{R}u_\perp \\ &-m\mathbf{P}_+ \hat{R}^* \hat{R}u_\perp + m\mathbf{P}_+ \hat{R}^* \mathbf{P}_+ \hat{R}u_\perp \\ &+\hat{R}^* m\mathbf{P}_- \hat{R}v - \mathbf{P}_- \hat{R}^* m\mathbf{P}_- \hat{R}v - m\mathbf{P}_+ \hat{R}^* \mathbf{P}_- \hat{R}v.\end{aligned} \tag{B.19}$$

Remark B.2.1 *All of the projections are now finite dimensional projections of the two basic forms, $\mathbf{P}_+ fg_\perp$ and $\mathbf{P}_- fg$, where $f \in L^\infty$ is rational and $g_\perp \in H_2^\perp$ and $g \in H^2$ are possibly infinite dimensional.*

Define, $f^+ := \mathbf{P}_+ f$ and $f^- := \mathbf{P}_- f$. Then, we have $\mathbf{P}_+ fg_\perp = \mathbf{P}_+ f^+ g_\perp$ and $\mathbf{P}_- fg = \mathbf{P}_- f^- g$, where $f^+ \in H^2 \cap L^\infty$ and $f^- \in H_2^\perp \cap L^\infty$ are *rational.*

B.3 Projection Formulas

In this section we give formulas for the projection onto H_2 of $f^+ g_\perp$, where $f^+ \in H^2 \cap L^\infty$ is rational and $g_\perp \in H_2^\perp$ is possibly infinite dimensional; and for the dual problem of taking the projection on $H_2^\perp$ of $\mathbf{P}_- f^- g$, where $f^- \in H_2^\perp \cap L^\infty$ is rational and $g \in H^2$ is possibly infinite dimensional. The solution of the first problem is just

the portion of the partial fraction expansion of $f^+(s)g_\perp(s)$ corresponding to the poles of $f^+(s)$. The only nontrivial part is the calculation of the coefficients for the poles of $f^+(s)$. For the general case we have that the coefficients a_{k-n} of $\frac{1}{(s-p_i)^n}$ are given by

$$a_{k-n} = (\frac{1}{n!})\frac{d^n}{ds^n}\left((s-p_i)^k f^+(s)g_\perp(s)\right)\Big|_{s=p_i}, \quad n = 1,\ldots,k, \tag{B.20}$$

where p_i is a k^{th} order pole of $f^+(s)$. From Leibnitz's formula, with $f_1(s) := (s-p_i)^k f^+(s)$, we have

$$a_{k-n} = (\frac{1}{n!})\sum_{l=0}^{k-n}\left(\begin{array}{c} n-k \\ l \end{array}\right) f_1^{(l)}(p_i)g_\perp^{(n-k-l)}(p_i). \tag{B.21}$$

The solution of the dual problem is just the dual of the above.

Remark B.3.1 *If we consider $g_\perp$ to be an unknown, then the projection represents a set of interpolation constraints on the unknown $g_\perp$. Furthermore, all of the terms in the projection are linearly independent functions of frequency and each term introduces an independent constraint on the unknown $g_\perp$.*

In this context it is clear that the repeated pole case is of exactly the same structure as the distinct pole case. Thus, we can consider the distinct pole case without loss of generality.

For the case where all of the poles of $f^+(s)$ are distinct, the projection onto H^2 is of the form

$$\mathbf{P}_+ f^+(s)g_\perp(s) = P_{f^+}\Phi_{g_\perp}, \tag{B.22}$$

where the entries of the row vector P_{f+} are the terms of the partial fraction expansion of $f^+(s)$ and Φ_g is a column vector with entries, $(\Phi_{g_\perp})_i = g_\perp(p_i)$, where the p_i are the poles of $f^+(s)$. For the dual problem, we have

$$\mathbf{P}_- f^-(s)g(s) = P_{f-}\Phi_g, \tag{B.23}$$

where the entries of the row vector P_{f-} are the terms of the partial fraction expansion of $f^-(s)$ and Φ_g is a column vector with entries, $(\Phi_g)_i = g(p_i)$, where the p_i are the poles of $f^-(s)$.

B.4 Necessary Interpolation Constraints

In this section we derive necessary interpolation constraints on u and v, that must be satisfied for σ to be a singular-value of the operator $\mathbf{A}$, see Definition 3.2.1. These constraints constitute a set of unknowns. We also construct a set of equations in these unknowns that must be satisfied in order for x to be the corresponding singular-vector. Therefore, these equations are also necessary conditions for σ to be a singular-value. Several assumptions are made in this section. Assumptions 2-3 are without loss of generality and simplify the construction. Assumptions 4-6 are not without loss of generality, but they ensure that the number of linearly independent equations equals the number of unknowns, and are typically satisfied.

Assumption 2 (genericity) *No repeated poles in* $\hat{R}_1, \hat{R}_2$, *or* $\hat{R}^*\hat{R} = \hat{V}^*\hat{V}$.

From Section B.3, Assumption 2 can be made without loss of generality. However, it allows the simple form of the projection formulas, (B.22) and (B.23), to be used. Now, we make the another simplifying assumption that is also without loss of generality.

Assumption 3 (genericity) $\mathcal{P}(R_1) \cap \mathcal{P}(R_2) = \emptyset$ *and there are no pole-zero cancellations in forming* $\hat{R}_1 = G_\sigma R_1$ *and* $\hat{R}_2 = G_\sigma R_2$.

Assumption 3 has no effect on the form of what follows. If the first condition is violated both the number of unknowns and the number of linearly independent equations in these unknowns are reduced by twice the number of common stable poles and three times the number of common unstable poles. If the second condition is violated both the number of unknowns and the number of linearly independent equations in these unknowns are reduced by the number of pole/zero cancellations.

Under Assumptions 2-3 we can explicitly denote the poles of $\hat{R}$. Denote the stable poles of R_1, $\mathcal{P}_-(R_1)$, by α_i, $i = 1, \ldots, n_1$, and the unstable poles of R_1, $\mathcal{P}_+(R_1)$, by β_i, $i = 1, \ldots, l_1$. Similarly, denote the stable poles of R_2, $\mathcal{P}_-(R_2)$, by α_i, $i = n_1 + 1, \ldots, n_1 + n_2$, and the unstable poles of R_2, $\mathcal{P}_+(R_2)$, by β_i, $i = l_1 + 1, \ldots, l := l_1 + l_2$. Finally, denote $\mathcal{P}(G_\sigma) = \mathcal{P}_-(G_\sigma)$ by α_i, $i = n_1 + n_2 + 1, \ldots, n := n_1 + n_2 + n_3$.

Remark B.4.1 *The following equations summarize the relationship between the poles:*

$$\mathcal{P}(V_1) = \mathcal{P}_-(V_1) = \{\alpha_i, \quad i = 1, \ldots, n_1\} \cup \{-\beta_i, \quad i = 1, \ldots, l_1\}, \tag{B.24}$$

$$\mathcal{P}(V_2) = \mathcal{P}_-(V_2) = \{\alpha_i, \quad i = n_1 + 1, \ldots, n_1 + n_2\} \cup \{-\beta_i, \quad i = l_1 + 1, \ldots, l\}, \tag{B.25}$$

$$\mathcal{P}(m_1^*) = \mathcal{P}_+(m_1^*) = \mathcal{P}_+(R_1), \quad \mathcal{P}(m_2^*) = \mathcal{P}_+(m_2^*) = \mathcal{P}_+(R_2), \quad \textit{and} \tag{B.26}$$

$$\mathcal{P}(\hat{V}) = \mathcal{P}_-(\hat{V}) = \{\alpha_i, \quad i = 1, \ldots, n\} \cup \{-\beta_j, \quad j = 1, \ldots, l\}. \tag{B.27}$$

Remark B.4.2 *Let* $W \in L^\infty$ *be a rational function, then*

$$\mathcal{P}(\mathbf{P}_+ W) = \mathcal{P}_-(\mathbf{P}_+ W), \quad \text{and} \quad \mathcal{P}(\mathbf{P}_- W) = \mathcal{P}_+(\mathbf{P}_- W). \tag{B.28}$$

Under Assumptions 2-3, it can be seen that (B.18) and (B.19) are of the form

$$v = \hat{P}_1\Phi_1 + \hat{P}_2\Phi_2 + \hat{P}_3\Phi_3, \tag{B.29}$$

and

$$(1 - \hat{V}^*\hat{V})u = P_1\Phi_1 + P_2\Phi_2 + P_3\Phi_3 + P_4\Phi_4 + P_5\Phi_5, \tag{B.30}$$

where the interpolation constraints, on the (possibly) infinite dimensional components u, v and $u_\perp = m^*u$ of the vector $y = u + mv$, are given by

$$\Phi_1 := \begin{bmatrix} u_\perp(-\beta_1) \\ \vdots \\ u_\perp(-\beta_l) \end{bmatrix}, \quad \Phi_2 := \begin{bmatrix} u_\perp(\alpha_1) \\ \vdots \\ u_\perp(\alpha_n) \end{bmatrix}, \quad \Phi_3 := \begin{bmatrix} v(\beta_1) \\ \vdots \\ v(\beta_l) \end{bmatrix},$$

$$\Phi_4 := \begin{bmatrix} u(-\alpha_1) \\ \vdots \\ u(-\alpha_n) \end{bmatrix}, \text{ and } \Phi_5 := \begin{bmatrix} u(\beta_1) \\ \vdots \\ u(\beta_l) \end{bmatrix}; \tag{B.31}$$

and the row vectors of frequency functions are determined by

$$\mathbf{P}_+\hat{R}^*\hat{R}u_\perp - \mathbf{P}_+\hat{R}^*\mathbf{P}_+\hat{R}u_\perp =: \hat{P}_1\Phi_1 + \hat{P}_2\Phi_2, \tag{B.32}$$

$$\mathbf{P}_+\hat{R}^*\mathbf{P}_-\hat{R}v =: \hat{P}_3\Phi_3, \tag{B.33}$$

$$-\hat{R}^*m\mathbf{P}_+\hat{R}u_\perp + \mathbf{P}_-\hat{R}^*m\mathbf{P}_+\hat{R}u_\perp - m\mathbf{P}_+\hat{R}^*\hat{R}u_\perp$$

$$+m\mathbf{P}_+\hat{R}^*\mathbf{P}_+\hat{R}u_\perp =: P_1\Phi_1 + P_2\Phi_2, \tag{B.34}$$

$$\hat{R}^*m\mathbf{P}_-\hat{R}v - \mathbf{P}_-\hat{R}^*m\mathbf{P}_-\hat{R}v - m\mathbf{P}_+\hat{R}^*\mathbf{P}_-\hat{R}v =: P_3\Phi_3, \text{ and} \tag{B.35}$$

$$-\mathbf{P}_-\hat{R}^*\hat{R}u =: P_4\Phi_4 + P_5\Phi_5. \tag{B.36}$$

Remark B.4.3 *The rightmost projection operator determines the interpolation constraints and therefore the number of unknowns (dimension of the associated column vector), while the leftmost projection operator determines the form of the frequency functions (row vector entries).*

Remark B.4.4 *As a practical matter, it is important to perform all pole-zero cancellations before taking the projections, in order to obtain the correct number of unknowns.*

If all pole-zero cancellations are not done before taking the projections, the extra unknowns, corresponding to the poles that were not canceled, will have coefficients that are identically zero.

Remark B.4.5 *From (B.22) and (B.23), it follows that when there is only one projection operator in a given term, the corresponding row vector necessarily has entries which are linearly independent frequency functions.*

Remark B.4.6 Φ_1, Φ_3 *and* Φ_5 *each contribute* l *(the number of unstable poles of* $\hat{R}$*) unknowns and* Φ_2 *and* Φ_4 *each contribute* n *(the number of stable poles of* $\hat{R}$*) unknowns. So, there are a total of* $2n + 3l$ *unknowns.*

Therefore, in order to solve for the unknowns, $2n + 3l$ linearly independent equations in these unknowns must be found. We get l equations from (B.18) by evaluating it at β_i, $i = 1 \ldots l$.

Remark B.4.7 *The lefthand side of (B.18) evaluated at the β_i is precisely Φ_3. Thus, these equations are clearly necessary conditions, since Φ_3 already appears on the right-hand side.*

Remark B.4.8 *Although there is no loss of generality in Assumption 2, the computation of these l equations is much more difficult for the repeated pole case. That is, the right hand side must be differentiated, with respect to s, $(k-1)$ times for a k^{th} order repeated pole in either $\hat{R}_1$ or $\hat{R}_2$.*

To get the remaining $2n+2l$ required equations we consider (B.19). Since $u \in H^2$, we know that the righthand side must equal zero at the unstable zeros of $(1-\hat{V}^*\hat{V})$, i.e. $\{\gamma_i \in C_+ : (1-\hat{V}^*(\gamma_i)\hat{V}(\gamma_i)) = 0\}$. Similarly, $m^*u = u_\perp \in H_2^\perp$ implies that either m^* or the righthand side of (B.19) must equal zero at the stable zeros of $(1-\hat{V}^*\hat{V})$, i.e. $\{\gamma_i \in C_- : (1-\hat{V}^*(\gamma_i)\hat{V}(\gamma_i)) = 0\}$. Thus the following assumption is necessary in order for there to be enough equations.

Assumption 4 $m^*(\gamma_i) \neq 0$, *for all* $\gamma_i \in C_+$, *such that* $(1-\hat{V}^*(\gamma_i)\hat{V}(\gamma_i)) = 0$.

Under Assumption 4, the righthand side of (B.19) must be zero at the stable zeros $(1-\hat{V}^*\hat{V})$. Therefore, the righthand side of (B.19) must equal zero at *all* of the zeros of $(1-\hat{V}^*\hat{V})$. To have enough linearly independent equations we must have the number of zeros of $(1-\hat{V}^*\hat{V})$ be twice the number of poles of $\hat{V}$. Thus, we require another assumption.

Assumption 5 $k = 2(n+l)$, *where k is the number of zeros of* $(1-\hat{V}^*\hat{V})$.

Assumption 5 is not without loss of generality and consists of two parts. First, it requires that $ord(\hat{V}^*\hat{V}) = 2ord(\hat{V})$, which is a standard assumption. Second, it requires that the number of zeros of $(1 - \hat{V}^*\hat{V})$ equals the number of poles of $\hat{V}^*\hat{V}$, which is almost always satisfied. Finally, since the equations obtained from repeated zeros are not linearly independent, we require one more assumption.

Assumption 6 *There are no repeated zeros in* $(1 - \hat{V}^*\hat{V})$.

Assumption 6 is satisfied for almost all σ. Assumptions 4-6 are sufficient to ensure that the number of linearly independent equations in the unknowns is equal to the number of unknowns. Assumption 6 can be restated in terms of the eigenvalues of a Hamiltonian matrix, see Remark B.4.10 below.

We now show that, the zeros of $(1 - \hat{V}^*\hat{V})$ are the eigenvalues of a Hamiltonian matrix. Again, we consider $1 - \hat{V}^*\hat{V} = 1 - \hat{R}^*\hat{R} = 0$. From the definition of $\hat{R}$, we have

$$0 \;=\; (1 - \hat{R}^*\hat{R}) \;=\; 1 - G_\sigma^* G_\sigma (R_1^* R_1 + R_2^* R_2). \tag{B.37}$$

If we multiply the last expression by $G_\sigma^{-*} G_\sigma^{-1}$, then the zeros are unchanged. Thus, we have

$$\begin{aligned} 0 \;&=\; G_\sigma^{-*} G_\sigma^{-1} - (R_1^* R_1 + R_2^* R_2) \;=\; \sigma^2 - (R_3^* R_3 + R_1^* R_1 + R_2^* R_2) \\ &=\; \sigma^2 - R^* R \;=\; \sigma^2 - F^* \Theta \Theta^* F \;=\; \sigma^2 I - F^* F. \end{aligned} \tag{B.38}$$

The zeros of $(\sigma^2 I - F^*F)$ are the poles of $(\sigma^2 I - F^*F)^{-1}$, and $\hat{F}^*\hat{F} = F^*F$, where $\hat{F} = \begin{bmatrix} V_1^T & V_2^T & V_3^T \end{bmatrix}^T$. Therefore, the following lemma can be used to calculate the zeros of $(1 - \hat{V}^*\hat{V})$.

Lemma B.4.1 *Let $\hat{F} = C(sI - A)^{-1}B + D$ be a minimal realization. If $\sigma > \|\mathbf{A}\|_e$, then the poles of $(\sigma^2 I - \hat{F}^*\hat{F})^{-1}$ are the eigenvalues of*

$$H_\sigma(\hat{F}) := \begin{bmatrix} A + B_\sigma C_\sigma & -B_\sigma B_\sigma^T \\ C^T C + C_\sigma^T C_\sigma & -(A + B_\sigma C_\sigma)^T \end{bmatrix}, \tag{B.39}$$

where $D_\sigma^T D_\sigma := \sigma^2 I - D^T D$, $B_\sigma := B D_\sigma^{-1}$ and $C_\sigma := D_\sigma^{-T} D^T C$.

Proof: $\hat{F}^* = (C(-sI - A)^{-1}B + D)^T = -B^T(sI + A^T)^{-1}C^T + D^T$, so we have

$$(\sigma^2 I - F^*F) = (\sigma^2 I - D^T D) - D^T C(sI - A)^{-1}B + B^T(sI + A^T)^{-1}C^T D$$
$$+B^T(sI + A^T)^{-1}C^T C(sI - A)^{-1}B. \tag{B.40}$$

By Lemma 3.2.3, D_σ^{-1} exist for all $\sigma > \|\mathbf{A}\|_e$. Thus, we have

$$(\sigma^2 I - \hat{F}^*\hat{F}) = D_\sigma^T(I - C_\sigma(sI - A)^{-1}B_\sigma + B_\sigma^T(sI + A^T)^{-1}C_\sigma^T$$
$$+B_\sigma^T(sI + A^T)^{-1}C^T C(sI - A)^{-1}B_\sigma)D_\sigma. \tag{B.41}$$

We observe that

$$I - C_\sigma(sI - A)^{-1}B_\sigma + B_\sigma^T(sI + A^T)^{-1}C_\sigma^T + B_\sigma^T(sI + A^T)^{-1}C^T C(sI - A)^{-1}B_\sigma$$
$$= I - \begin{bmatrix} C_\sigma & -B_\sigma^T \end{bmatrix} \begin{bmatrix} (sI - A) & 0 \\ -C^T C & (sI + A^T) \end{bmatrix}^{-1} \begin{bmatrix} B_\sigma \\ C_\sigma^T \end{bmatrix}. \tag{B.42}$$

Hence, the poles of $(\sigma^2 I - \hat{F}^*\hat{F})^{-1}$ are the poles of

$$\left(I - \begin{bmatrix} C_\sigma & -B_\sigma^T \end{bmatrix} \begin{bmatrix} (sI - A) & 0 \\ -C^T C & (sI + A^T) \end{bmatrix}^{-1} \begin{bmatrix} B_\sigma \\ C_\sigma^T \end{bmatrix} \right)^{-1}, \tag{B.43}$$

which, by the matrix inversion lemma [43], are the poles of $\left[sI - H_\sigma(\hat{F})\right]^{-1}$. Since (A, B, C, D) is a minimal realization of $\hat{F}$, we have that the poles of $(\sigma^2 I - \hat{F}^*\hat{F})^{-1}$ are the eigenvalues of $H_\sigma(\hat{F})$. □

Remark B.4.9 *If $\hat{F}$ is strictly proper, (i.e. $D = 0$), then $H_\sigma(\hat{F})$ reduces to*

$$H_\sigma(\hat{F}) := \begin{bmatrix} A & -\sigma^{-2}BB^T \\ C^TC & -A^T \end{bmatrix}. \tag{B.44}$$

Remark B.4.10 *From Lemma B.4.1 it follows that, the zeros of $(1 - \hat{V}^*\hat{V})$ are the eigenvalues of $H_\sigma(\hat{F})$. So, Assumption 6 a can be restated in terms of the eigenvalues of $H_\sigma(\hat{F})$. That is, Assumption 6 is equivalent to: The k eigenvalues, γ_i, $i = 1, \ldots, k$, of $H_\sigma(\hat{F})$ are distinct.*

Under Assumptions 1-6, the following set of $2n+3l$ equations in $2n+3l$ unknowns, are necessary conditions for σ to be a singular-value of $\mathbf{A}$:

$$\begin{bmatrix} v(\beta_1) \\ \vdots \\ v(\beta_l) \end{bmatrix} = \begin{bmatrix} \hat{P}_1(\beta_1) \\ \vdots \\ \hat{P}_1(\beta_l) \end{bmatrix} \Phi_1 + \begin{bmatrix} \hat{P}_2(\beta_1) \\ \vdots \\ \hat{P}_2(\beta_l) \end{bmatrix} \Phi_2 + \begin{bmatrix} \hat{P}_3(\beta_1) \\ \vdots \\ \hat{P}_3(\beta_l) \end{bmatrix} \Phi_3, \tag{B.45}$$

$$\begin{aligned} 0 = &\begin{bmatrix} P_1(\gamma_1) \\ \vdots \\ P_1(\gamma_k) \end{bmatrix} \Phi_1 + \begin{bmatrix} P_2(\gamma_1) \\ \vdots \\ P_2(\gamma_k) \end{bmatrix} \Phi_2 + \begin{bmatrix} P_3(\gamma_1) \\ \vdots \\ P_3(\gamma_k) \end{bmatrix} \Phi_3 \\ &+ \begin{bmatrix} P_4(\gamma_1) \\ \vdots \\ P_4(\gamma_k) \end{bmatrix} \Phi_4 + \begin{bmatrix} P_5(\gamma_1) \\ \vdots \\ P_5(\gamma_k) \end{bmatrix} \Phi_5. \end{aligned} \tag{B.46}$$

It is clear from the construction that these equations are independent so long as the γ_i are distinct. Note that any u constructed from these equations will necessarily be in $H(M)$.

To simplify (B.45) and (B.46), we make the following definitions:

$$\hat{r}_1 := \begin{bmatrix} \hat{P}_1(\beta_1) \\ \vdots \\ \hat{P}_1(\beta_l) \end{bmatrix}, \quad \hat{r}_2 := \begin{bmatrix} \hat{P}_2(\beta_1) \\ \vdots \\ \hat{P}_2(\beta_l) \end{bmatrix}, \quad (\hat{r}_3 + I) := \begin{bmatrix} \hat{P}_3(\beta_1) \\ \vdots \\ \hat{P}_3(\beta_l) \end{bmatrix}, \tag{B.47}$$

$$r_1 := \begin{bmatrix} P_1(\gamma_1) \\ \vdots \\ P_1(\gamma_k) \end{bmatrix}, \quad r_2 := \begin{bmatrix} P_2(\gamma_1) \\ \vdots \\ P_2(\gamma_k) \end{bmatrix}, \quad r_3 := \begin{bmatrix} P_3(\gamma_1) \\ \vdots \\ P_3(\gamma_k) \end{bmatrix}, \tag{B.48}$$

$$r_4 := \begin{bmatrix} P_4(\gamma_1) \\ \vdots \\ P_4(\gamma_k) \end{bmatrix}, \quad \text{and} \quad r_5 := \begin{bmatrix} P_5(\gamma_1) \\ \vdots \\ P_5(\gamma_k) \end{bmatrix}. \tag{B.49}$$

Since the lefthand side of (B.45) is precisely Φ_3, (B.45) and (B.46) can be rewritten as

$$0 = \hat{r}_1\Phi_1 + \hat{r}_2\Phi_2 + \hat{r}_3\Phi_3, \tag{B.50}$$

$$0 = r_1\Phi_1 + r_2\Phi_2 + r_3\Phi_3 + r_4\Phi_4 + r_5\Phi_5. \tag{B.51}$$

Finally, we define

$$R_\sigma := \begin{bmatrix} \hat{r}_1 & \hat{r}_2 & \hat{r}_3 & 0 & 0 \\ r_1 & r_2 & r_3 & r_4 & r_5 \end{bmatrix}. \tag{B.52}$$

The matrix R_σ is, by construction, singular whenever σ is a singular-value of $\mathbf{A}$. That is, the set of $2n + 3l$ equations in $2n + 3l$ unknowns representing the necessary conditions for σ to be a singular-value can be written as, $R_\sigma\Phi = 0$, where $\Phi := \begin{bmatrix} \Phi_1^T & \Phi_2^T & \Phi_3^T & \Phi_4^T & \Phi_5^T \end{bmatrix}^T$.

B.5 Sufficient Conditions for Singular-values of A

Thus far we have constructed a $(2n + 3l) \times (2n + 3l)$ matrix, R_σ, that is singular whenever σ is a singular-value for the operator $\mathbf{A}$, defined by (3.16). In this section we prove that if R_σ is singular, for some σ, then this value of σ is a singular-value of the operator $\mathbf{A}$. This implies that the singular-values can be found by searching

for the values of σ that make R_σ singular. Furthermore, given a singular R_σ, we can find a Φ, such that $R_\sigma\Phi = 0$, and construct v and u from Φ, using (B.29) and (B.30). Thus, for any singular-value, σ, R_σ is singular, and we can calculate the corresponding singular-vector, $x = G_\sigma y = G_\sigma(u + mv)$. Thus, by construction we have necessity (the only if part). The proof of sufficiency (the if part) is a little more subtle, but is much shorter. First, we prove that if R_σ is singular, i.e. there exist a $\Phi' \neq 0$, such that $R_\sigma\Phi' = 0$, then the pair (σ, x), corresponding to Φ', is a singular-value/singular-vector pair for **A**.

Proof [sufficiency]: Suppose that we have a vector $\Phi' \neq 0$ satisfying $R_\sigma\Phi' = 0$, construct v and u from (B.29) and (B.30), by using Φ' in place of Φ on the righthand sides of (B.29) and (B.30). Then, to complete the proof we need to show that, Φ corresponding to v and u, from (B.29) and (B.30), is precisely Φ'. By construction we have

$$v = \hat{P}_1\Phi_1' + \hat{P}_2\Phi_2' + \hat{P}_3\Phi_3' = \hat{P}_1\Phi_1 + \hat{P}_2\Phi_2 + \hat{P}_3\Phi_3 \tag{B.53}$$

and

$$\begin{aligned}(1 - \hat{V}^*\hat{V})u &= P_1\Phi_1' + P_2\Phi_2' + P_3\Phi_3' + P_4\Phi_4' + P_5\Phi_5' \\ &= P_1\Phi_1 + P_2\Phi_2 + P_3\Phi_3 + P_4\Phi_4 + P_5\Phi_5.\end{aligned} \tag{B.54}$$

Thus it is sufficient to show that $\begin{bmatrix} P_1 & P_2 & P_3 & P_4 & P_5 \end{bmatrix}$ has linearly independent entries as functions of the frequency variable. From (B.34) we see that P_1 comes only from $m\mathbf{P}_+\hat{R}^*\hat{R}u_\perp$ and has linearly independent entries of the form $\frac{m(s)a_i}{s+\beta_i}$. Furthermore, the term $\hat{R}^*m\mathbf{P}_+\hat{R}u_\perp$ ensures that P_2 has independent entries since its contributions are independent of all the others due to the multiplication by $\hat{R}^*(s)m(s)$.

Thus, each entry of P_2 has a term of the form $\frac{\hat{R}^*(s)m(s)b_i}{s-\alpha_i}$. Similarly, from (B.35) we see that P_3 has linearly independent entries and each entry has a term of the form $\frac{\hat{R}^*(s)m(s)c_i}{s-\beta_i}$. Finally, from (B.36) we have that P_4 has linearly independent entries of the form $\frac{d_i}{s+\alpha_i}$ and P_5 has linearly independent entries of the form $\frac{e_i}{s-\beta_i}$. From the preceding it is easy to see that $[\ P_1 \quad P_2 \quad P_3 \quad P_4 \quad P_5\ ]$ has linearly independent entries.

Remark B.5.1 *The coefficients a_i, b_i, c_i, d_i and e_i of the entries are non-zero.*

Indeed, if any of the coefficients were zero this would mean that there was a pole-zero cancellation in one of the rational functions which would mean that the unknown (interpolation point) corresponding to this zero coefficient is not needed. □

B.6 Construction of the Singular-vector x^0

The singular-vector x^0, corresponding to $\sigma = \sigma_{max} := \gamma_{opt}$, can be constructed directly from the interpolation constraints, Φ^0, obtained from $R_{\sigma_{max}}\Phi^0 = 0$. Specifically, we have $x^0 = G_{\sigma_{max}}y^0 = G_{\sigma_{max}}(u^0 + mv^0)$. Now, v^0 and u^0 can be constructed from Φ^0 using (B.29) and (B.30), i.e. $v^0 = \hat{P}_1\Phi_1^0 + \hat{P}_2\Phi_2^0 + \hat{P}_3\Phi_3^0$ and $(1 - \hat{V}^*\hat{V})u^0 = P_1\Phi_1^0 + P_2\Phi_2^0 + P_3\Phi_3^0 + P_4\Phi_4^0 + P_5\Phi_5^0$. From (B.32) and (B.33), it can be seen that v^0 is purely rational. Similarly, from (B.34) through (B.36), it can be seen that u^0 can be written as $u^0 = u_1^0 + mu_2^0$, where u_1^0 and u_2^0 are rational functions. Thus, x^0 can be written as $x^0 = g + mh$, where $g := G_\sigma u_1^0$ and $h := G_\sigma(v^0 + u_2^0)$ are rational functions.

Appendix C

Realization of Transfer Function Matrices

In this appendix we look at the issues and problems associated with obtaining exact minimal realizations of transfer function matrices. We show that the relationship between the Markov parameters and the coefficients of state space realizations is Vandermonde in nature. We also show that the standard methods, e.g. [12], which use Markov parameters, are averaging methods and thus only provide estimates of the state space realization. The averaging is necessary due to the fact that the relationship is numerically ill-conditioned. We present an exact method for obtaining the state space realization from the Markov parameters. It is, of course, only usable for "low" order systems with "small" dynamic range of pole locations, where "low" and "small" both depend on the available computational accuracy (machine precision). We also present a method that has no more numerical difficulties than the single input single output ($SISO$) case. There are numerous direct methods for $SISO$ transfer functions that do not have significant numerical problems, i.e. are not numerically ill-conditioned, e.g. the "tf2ss" command in MATLAB. Our direct method for obtaining exact minimal state space realizations of transfer function matrices requires the construction of a minimal block diagram consisting of proper $SISO$ transfer functions. This method is completely general and is illustrated by solving the example

arising in Chapter III. Our method for constructing this minimal block diagram is based on the familiar least common denominator (LCD) of all minors of the transfer function matrix method for determining the characteristic polynomial, see e.g. [12]. Typically it is quite simple to construct the minimal block diagram as is the case for the problem considered in [73]. For certain pathological cases it becomes somewhat involved, but it is still quite tractable, as we show by means of a couple of examples.

C.1 Vandermonde Structure of the Relationship

In this section we show that the mathematical relationship between the Markov parameters of a transfer function matrix and any state space realization has a Vandermonde structure. The Markov parameters are defined by

$$G(s) = \sum_{k=0}^{\infty} H(k)s^{-k}, \tag{C.1}$$

where the $H(k)$ are the Markov parameters. It can be shown, see [12], that they are related to all state space realizations, (A,B,C,D), by

$$H(0) = D, \quad H(i+1) = CA^iB, \quad i = 0,1,2,\ldots . \tag{C.2}$$

Let $G(s)$ be a $p \times m$ transfer function matrix, then the $H(k)$ are also $p \times m$ matrices. For simplicity we consider the distinct eigenvalue case, which is generic in that matrices with distinct eigenvalues are dense in the set of all matrices [43]. Thus, the numerical properties of matrices with distinct eigenvalues are representative of the numerical properties of all matrices. Without loss of generality, let A be a diagonal matrix with diagonal elements (eigenvalues) λ_i. Then, we have the following set of

equations:

$$\sum_{j=1}^{n} C_{ij}B_{jk} = H_{ik}(1) =: h_{ik}^1, \tag{C.3}$$

$$\sum_{j=1}^{n} C_{ij}\lambda_j B_{jk} = H_{ik}(2) =: h_{ik}^2, \tag{C.4}$$

$$\sum_{j=1}^{n} C_{ij}\lambda_j^2 B_{jk} = H_{ik}(3) =: h_{ik}^3, \tag{C.5}$$

$$\vdots \quad \vdots \quad \vdots$$

$$\sum_{j=1}^{n} C_{ij}\lambda_j^{n-1} B_{jk} = H_{ik}(n) =: h_{ik}^n. \tag{C.6}$$

Let $\lambda = \begin{bmatrix} \lambda_1 & \cdots & \lambda_n \end{bmatrix}$ and $\alpha_{ik}^j = C_{ij}B_{jk}$, and define

$$h_{ik} := \begin{bmatrix} h_{ik}^1 \\ \vdots \\ h_{ik}^n \end{bmatrix}, \quad \alpha_{ik} := \begin{bmatrix} \alpha_{ik}^1 \\ \vdots \\ \alpha_{ik}^n \end{bmatrix}, \quad \text{and} \quad \Lambda := \begin{bmatrix} \lambda^0 \\ \lambda^1 \\ \vdots \\ \lambda^{n-1} \end{bmatrix}, \tag{C.7}$$

where $\lambda^k = \begin{bmatrix} \lambda_1^k & \cdots & \lambda_n^k \end{bmatrix}$. Then we have the Vandermonde relationship given by

$$\Lambda\alpha_{ik} = h_{ik}. \tag{C.8}$$

This Vandermonde relationship is inherent to all state space realizations, since they are all equivalent and related by similarity transforms. Notice that only n Markov parameters are required to calculate the strictly proper part, (A, B, C), of the state space realization. The standard methods [12], on the other hand, typically use $2n$ Markov parameters. The averaging over a larger data set helps overcome the numerical ill-conditioning of the relationship. The drawback of the averaging methods are that the answer is not exact even for systems where direct use of the Vandermonde relationship is numerically sound.

To illustrate the problems with the approximation properties of the standard Markov based realization methods we present a third order example for which the exact Vandermonde relationship can be used. All calculation were done using MATLAB. Using the random number generator we obtain the random third order 3×2 transfer function matrix

$$G = \frac{1}{d}\begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \\ g_{31} & g_{32} \end{bmatrix}, \quad \text{where} \tag{C.9}$$

$$g_{11} = 0.7254s^3 + 0.2096s^2 + 0.3581s^1 - 0.0140, \tag{C.10}$$

$$g_{12} = 0.9995s^3 - 0.2321s^2 + 0.0223s^1 + 0.0236, \tag{C.11}$$

$$g_{21} = 0.8886s^3 + 0.5431s^2 + 0.1100s^1 + 0.0005, \tag{C.12}$$

$$g_{22} = 0.2332s^3 + 0.3300s^2 + 0.8220s^1 + 0.0629, \tag{C.13}$$

$$g_{31} = 0.3063s^3 + 0.1610s^2 + 0.5361s^1 + 0.0097, \tag{C.14}$$

$$g_{32} = 0.3510s^3 + 0.7905s^2 + 0.5093s^1 + 0.0376, \quad \text{and} \tag{C.15}$$

$$d = 1.0000s^3 - 0.7939s^2 - 0.5872s^1 + 0.1097. \tag{C.16}$$

Using the direct Vandermonde method (which can be readily verified to give the correct transfer function matrix to within a few machine epsilons, 10^{-16} for MATLAB), we obtain

$$A = \begin{bmatrix} 1.2055 & 0 & 0 \\ 0 & -0.5710 & 0 \\ 0 & 0 & 0.1594 \end{bmatrix}, \quad B = \begin{bmatrix} 1.1550 & 0.8485 \\ 0.4816 & 0.4237 \\ 0.3626 & 0.1788 \end{bmatrix},$$

$$C = \begin{bmatrix} 0.9286 & -0.4564 & -0.1853 \\ 1.1550 & -0.0811 & -0.1279 \\ 0.6646 & -0.4816 & -0.3626 \end{bmatrix}, \quad \text{and} \quad D = \begin{bmatrix} 0.7254 & 0.9995 \\ 0.8886 & 0.2332 \\ 0.3063 & 0.3510 \end{bmatrix}. \tag{C.17}$$

For comparison we consider the Markov based method of [12], which uses singular-value decomposition (SVD) reduction of the $2n \times 2n$ Hankel matrix of Markov parameters. To allow for direct comparison we diagonalize the result and obtain

$$\hat{A} = \begin{bmatrix} 1.2085 & 0 & 0 \\ 0 & -0.3976 & 0 \\ 0 & 0 & 1.1740 \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} -1.0854 & -1.1718 \\ -0.4957 & -0.6518 \\ -0.6590 & 0.3903 \end{bmatrix},$$

$$\hat{C} = \begin{bmatrix} -0.7220 & -0.4518 & 0.1978 \\ -0.9204 & -0.5203 & 0.3033 \\ -0.8701 & -0.4884 & -0.6599 \end{bmatrix}, \text{ and } \hat{D} = \begin{bmatrix} 0.7254 & 0.9995 \\ 0.8886 & 0.2332 \\ 0.3063 & 0.3510 \end{bmatrix}. \quad \text{(C.18)}$$

Note that D and $\hat{D}$ are calculated in exactly the same way and are therefore identical. Note that the error in the poles (diagonal entries of $\hat{A}$) increases as you go down the diagonal. This is because they are ordered according to the magnitude of the corresponding singular-value. Increasing error with decreasing magnitude of the singular-values is an inherent property of the SVD. Specifically, the error in the first pole (1.2055) is only 0.25 percent, the error in the second pole ($-.5710$) is a whopping 30 percent, and the last pole has an incredible 636 percent error. These errors are fairly typical, although sometimes the errors are much smaller, particularly the worst case error is often less than 100 percent. The approximation error in the B and C matrices is also significant, thus the approximation of the zeros is also poor. To more closely examine the error in the zero locations, define $\hat{G}$ to be the transfer function matrix corresponding to $(\hat{A}, \hat{B}, \hat{C}, \hat{D})$. Then, we have

$$\hat{G} = \frac{1}{\hat{d}} \begin{bmatrix} \hat{g}_{11} & \hat{g}_{12} \\ \hat{g}_{21} & \hat{g}_{22} \\ \hat{g}_{31} & \hat{g}_{32} \end{bmatrix}, \quad \text{where} \quad \text{(C.19)}$$

$$\hat{g}_{11} = 0.7254s^3 - 0.5941s^2 + 0.0313s^1 - 0.4035, \quad \text{(C.20)}$$

$$\hat{g}_{12} = 0.9995s^3 - 1.4876s^2 + 0.5552s^1 - 0.2117, \quad \text{(C.21)}$$

$$\hat{g}_{21} = 0.8886s^3 - 0.6379s^2 - 0.0573s^1 - 0.5083, \quad \text{(C.22)}$$

$$\hat{g}_{22} = 0.2332s^3 + 0.1691s^2 + 0.1631s^1 - 0.7917, \quad \text{(C.23)}$$

$$\hat{g}_{31} = 0.3063s^3 - 0.1214s^2 + 0.1199s^1 - 0.4513, \quad \text{(C.24)}$$

$$\hat{g}_{32} = 0.3510s^3 + 0.3250s^2 - 0.1847s^1 - 0.8095, \quad \text{and} \quad \text{(C.25)}$$

$$\hat{d} = 1.0000s^3 - 1.9849s^2 + 0.4716s^1 + 0.5641. \quad \text{(C.26)}$$

Clearly, the numerators (zeros) of the transfer functions are highly inaccurate. To see how strongly these errors show up in the zero locations, we consider the roots, r_{11}, of g_{11} and the roots, $\hat{r}_{11}$, of $\hat{g}_{11}$

$$r_{11} = \{0.0381 - 0.1635 \pm j0.6924\} \text{ and } \hat{r}_{11} = \{1.680 + .1602 \pm j0.3946\}. \quad \text{(C.27)}$$

The error in the real zero is very extreme (over 2900 percent) and the error in the complex zero is quite significant (2.02 percent for the real part and 43 percent for the imaginary part). The large error one may encounter with the indirect Markov methods indicates the utility of using an exact method whenever possible. The exact Vandermonde method is almost never usable in MATLAB. Indeed, for seventh order systems and moderate dynamic range of pole locations the Vandermonde matrix becomes so ill-conditioned that reasonable solutions from it are impossible using MATLAB. However, using arbitrary precision packages, such as Maple or Mathematica, the only limit is the computation time and memory usage, which both increase very rapidly, due to the numerical ill-conditioning of the Vandermonde structure. The direct method described in the next section avoids this problem because it does not

use Markov parameters. It is a *direct* method in the same sense as the direct methods used for $SISO$ systems in any standard control design software package.

C.2 Exact Minimal Realizations

The method presented in this section is a completely general method for obtaining the state space realization of all proper rational transfer function matrices. Furthermore, the resulting realization is exact, up to the errors on the order of the machine precision associated with numerical multiplication and addition. The method consists of three steps:

1. Obtain a minimal block diagram consisting of proper rational $SISO$ transfer functions.

2. Obtain minimal state space realizations of the individual $SISO$ blocks.

3. Combine the individual realizations in accordance with the minimal block diagram.

The first step is very straight forward, except for pathological cases, and algorithms could be readily developed and implemented. The second step is already implemented on any standard control software package. The third step is very straight forward and has been implemented in MATLAB's μ-tools toolbox. Specifically, the MATLAB function is "*sysic.m*", which calculates the overall state-space description of the system from a detailed description of the "sys(tem) i(nter)c(onnection)". Ideally, a single algorithm would implement all three steps from a single command line

(program call). This is certainly achievable for the non-pathological cases. Certain pathological cases may be better handled with the aid of a graphical interface such as MATLAB's SIMULINK.

C.2.1 Construction of the minimal block diagram

The minimal block diagram is constructed with the aid of the LCD of the minors method for determining the characteristic polynomial. Pathological cases are those in which a pole/zero cancellation occurs during the construction of a minor. We will show how to handle such cases by considering a specific example. If the entries of the transfer function matrix have *no* common poles, then the minimal block diagram can be written down directly with each entry being a separate block. Thus, we consider only cases with common poles. Typically transfer function matrices arising from physical systems will have common poles. In general transfer function matrices arising from physical systems will have a common zero corresponding to each common pole. In this case the minimal block diagram can be written down by inspection with the aid of the LCD's of the minors. If there are *not* corresponding common zeros, then a slightly modified procedure must be used. We give an example showing how to handle such cases. The procedure is not complicated and it is believed that this case can be easily handled by a general algorithm for exact minimal state space realizations. Finally, we show how easy it is to construct the exact minimal realization for typical problems such as the one in Chapter III.

Consider the transfer function matrix

$$G = \begin{bmatrix} \frac{a}{s(s+1)} & \frac{b}{s(s+2)} \\ \frac{c}{s(s+3)} & \frac{d}{s(s+4)} \end{bmatrix}. \tag{C.28}$$

From the minors consisting of the individual elements, we know that the characteristic equation must include $p(s) = s(s+1)(s+2)(s+3)(s+4)$. Generically, it would include an s^2 term from the minor consisting of the determinant of G. The pathological case we are interested in is the case where the second s term cancels and $p(s)$ is the characteristic equation. This will be true iff $6ad = 4bc$, so letting $a = 2$, $b = 3$, $c = 1$, and $d = 1$ makes $p(s)$ the characteristic polynomial. Thus, the minimal block diagram must contain *only* one block with a pole at the origin. Consider the minimal block diagram shown in Figure 32. Setting this diagram equal to G results in the equivalence conditions: $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = -\gamma$, $\beta_1 = -(\delta + \gamma)$, $\beta_2 = -(\delta + 2\gamma)$, $\beta_3 = -(\delta + 3\gamma)$, $\beta_4 = -(\delta + 4\gamma)$, $r_1\delta c_1 = a$, $2r_1\delta c_2 = b3r_2\delta c_1 = c$, $4r_2\delta c_2 = d$. Using $a = 2$, $b = 3$, $c = 1$, and $d = 1$ and letting $\delta = 1$, the last four conditions become

$$\begin{array}{ll} r_1c_1 = 2 & r_1c_2 = 2/3 \\ r_2c_1 = 1/3 & r_2c_2 = 1/4 \end{array}. \tag{C.29}$$

These equations are satisfied for infinitely many choices of the remaining variables. One solution is $r_1 = 1$, $r_2 = 1/6$, $c_1 = 2$, and $c_2 = 3/2$. We have already arbitrarily chosen $\delta = 1$ and we must still arbitrarily choose γ in order to specify a solution, thus the block diagram of Figure 32 had *two* extra degrees of freedom. Choosing $\gamma = 1$ yields the block diagram shown in Figure 33 which is a minimal block diagram for G.

We now look at the case where the common poles do not have corresponding common zeros. Without loss of generality we can consider the biproper case, since

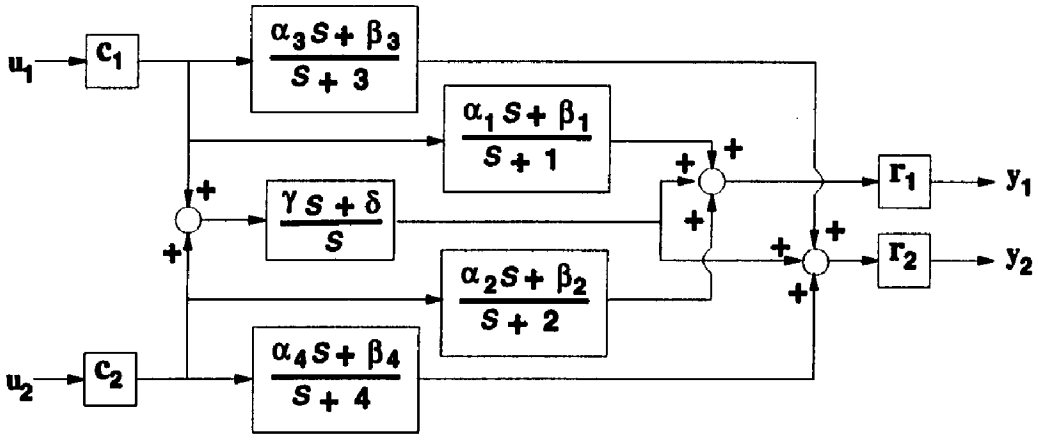


Figure 32: Solution of the pathological case, general form.

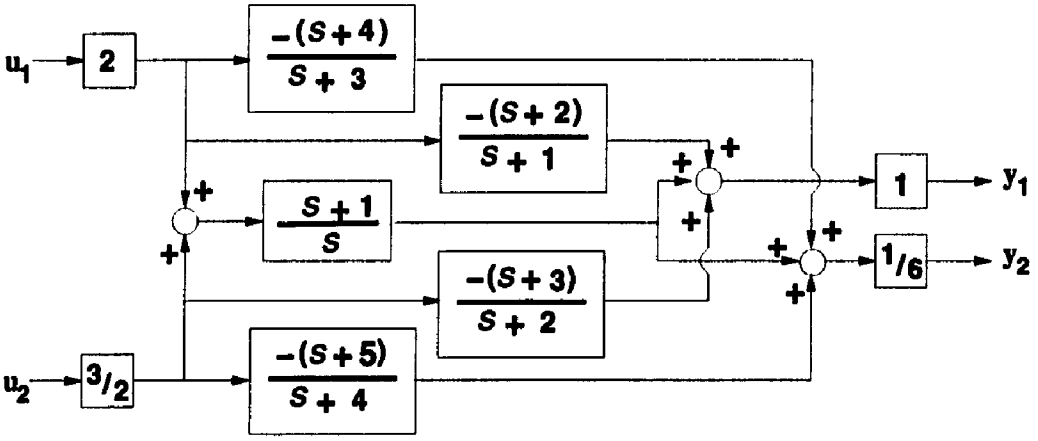


Figure 33: Particular solution, pathological case.

if the entries are both strictly proper they have a common zero at infinity. Consider the transfer function matrix

$$G = \begin{bmatrix} \frac{(s+a)(s+b)}{(s+1)(s+2)} & \frac{(s+c)(s+d)}{(s+1)(s+3)} \end{bmatrix}, \tag{C.30}$$

where we assume that the zeros are *not* common and do not cancel with any of the poles. Now we simply introduce zeros at infinity by decomposing the entries into the sums of constants and strictly proper transfer functions. Specifically, we can write G

as

$$G = \begin{bmatrix} f & h \end{bmatrix} + \begin{bmatrix} \frac{(s+e)}{(s+1)(s+2)} & \frac{(s+g)}{(s+1)(s+3)} \end{bmatrix}. \tag{C.31}$$

Now we simply factor out the common pole and draw, by inspection, the minimal block diagram shown in Figure 34. In this way common zeros can be obtained whenever needed by the addition of constant gain paths. While such situations require a little more attention, they are still completely straight forward and algorithm development should not be difficult.

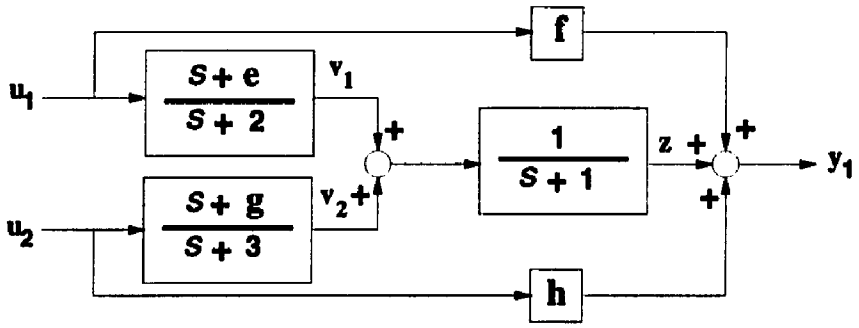


Figure 34: Minimal block diagram, non-common zeros.

C.2.2 Practical example

The details of obtaining the state space realization required for the numerical example in Chapter III are now given. While the transfer function matrix to be considered here does not represent a physical plant, it does come from physical considerations and thus has common zeros corresponding to its common poles and contains no pathologies. The only exception to this is that the zeros W_1 have been purposely chosen to cancel with poles of N, see Section 3.4. Due to the sparse structure of G, this does not cause

any complications. The transfer function matrix is given by

$$G := \begin{bmatrix} W_1 N & 0 \\ W_1 W_2 & -W_1 W_2 D \\ 0 & -W_2 D \end{bmatrix}. \tag{C.32}$$

Denote the numerator of a transfer function X by X^n and the denominator by X^d. After pole/zero cancellations, we have

$$G := \begin{bmatrix} \frac{N^n}{W_1^d D^d} & 0 \\ \frac{W_1^n W_2^n}{W_1^d W_2^d} & \frac{W_1^n W_2^n D^n}{W_1^d W_2^d D^d} \\ 0 & \frac{W_2^n D^n}{W_2^d D^d} \end{bmatrix}, \quad \text{and let} \quad \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} := G \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \tag{C.33}$$

Even without considering the minors, we can see that the the transfer function $W_1 W_2$ acts on both the inputs in generating the middle output, y_2. Thus, it seems reasonable to put this term down as a block connected to y_2 (see Figure 35). It is also clear that the second input, u_2, always acts through the transfer function D. So, we place a block with transfer function D at the input u_2. For this example that is all that is required to come up with the minimal block diagram. It can be readily verified that the characteristic polynomial is $W_1^d W_1^d W_2^d W_2^d D^d D^d$, which is in accordance with the block diagram in Figure 35.

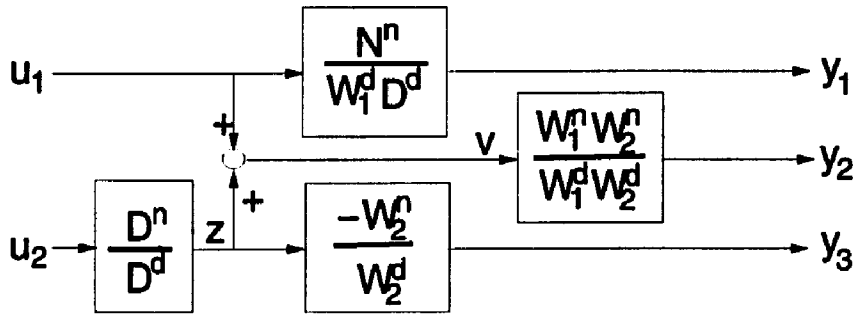


Figure 35: Minimal block diagram for the example.

Remark C.2.1 *We could at this point simply write a MATLAB "m-file" describing the block diagram in Figure 35 and use the μ-tools command "sysic.m". However, for illustrative purposes we present the abstract composite system in terms of the state space representation of the individual blocks.*

Let the transfer functions, T_i, of the individual blocks be given by

$$T_1 = \frac{N^n}{W_1^d D^d}, \quad T_2 = \frac{W_1^n W_2^n}{W_1^d W_2^d}, \tag{C.34}$$

$$T_3 = \frac{-W_2^n}{W_2^d}, \quad \text{and} \quad T_4 = \frac{D^n}{D^d}. \tag{C.35}$$

Denote the state space realizations by $T_1 := (A_1, B_1, C_1, D_1)$, $T_2 =: (A_2, B_v, C_2, D_2)$, $T_3 =: (A_3, B_z, C_3, D_3)$, and $T_4 =: (A_z, B_2, C_z, D_z)$, where the subscripts of the A, C, and D matrices correspond to the output variable, and the subscripts of the B matrices correspond to the input variable. With a little algebra it can be seen that the resulting minimal multiple input multiple output ($MIMO$) state space realization $G =: (A, B, C, D)$ is given by

$$A = \begin{bmatrix} A_1 & 0 & 0 & 0 \\ 0 & A_2 & 0 & -B_v C_z \\ 0 & 0 & A_3 & B_z C_z \\ 0 & 0 & 0 & A_z \end{bmatrix}, \quad B = \begin{bmatrix} B_1 & 0 \\ B_v & -B_v D_z \\ 0 & B_z D_z \\ 0 & B_2 \end{bmatrix},$$

$$C = \begin{bmatrix} C_1 & 0 & 0 & 0 \\ 0 & C_2 & 0 & -D_2 C_z \\ 0 & 0 & C_3 & D_3 C_z \end{bmatrix}, \quad \text{and} \quad D = \begin{bmatrix} D_1 & 0 \\ D_2 & -D_2 D_z \\ 0 & D_3 D_z \end{bmatrix}. \tag{C.36}$$

An algorithm was implemented in MATLAB that generates the state space realization automatically given the parameters that constitute G.

C.3 Conclusions

The inaccuracy of Markov parameter based methods clearly illustrates the need for a new approach. The above approach appears to be a viable solution. The next step is to develop detailed algorithms suitable for implementation in software packages such as MATLAB. This should not be difficult for the non-pathological cases. Another issue that should be addressed is balanced realizations. A nice feature of the Markov based methods is that they provide balanced realizations, however this advantage is completely inadequate to compensate for the inaccuracy of the zeros. More work needs to be done with respect to incorporation of a certain level of balancing in the direct method. It seams reasonable to expect that with a combination of balancing of the individual $SISO$ realizations and appropriate scalings in the block diagrams that "nearly" balanced realizations could be obtained. An interesting investigation would be to compare the results of an SVD based balancing of the direct realization with the balanced realization resulting from the Markov methods. It is expected that for most, if not all, cases the direct method will still prove superior. Finally, the extension of this direct method to descriptor systems should be straight forward.

Appendix D

Proof of Theorem 5.5.1, Sampled-data Extension

In this appendix we give a detailed outline of the proof of Theorem 5.5.1. We start out by presenting several useful identities and making a couple more definitions. Then we derive explicit matrix formulas for the various finite rank operators given in Theorem 5.5.1 in terms of the sampling period, τ.

D.1 Mathematical Preliminaries

Let $C_Q := \begin{bmatrix} (B_1^* + D_{11}^* D_Q D_{11} B_1^*) & D_{11}^* D_Q C_1 \end{bmatrix}$ and $s \bigvee t := max(s,t)$. The following identities, which can be readily verified using elementary mathematics, will be used freely in the sequel, without comment:

$$\frac{d}{ds}\left\{ e^{-A_R s} \begin{bmatrix} I \\ 0 \end{bmatrix} e^{-A^* s} \right\} \equiv e^{-A_R s} B_R B_1^* e^{-A^* s}, \tag{D.1}$$

$$\frac{d}{ds}\left\{ e^{-A_Q s} \begin{bmatrix} 0 \\ I \end{bmatrix} e^{A s} \right\} \equiv e^{-A_Q s} B_Q C_1 e^{A s}, \tag{D.2}$$

$$\frac{d}{ds}\left\{ e^{-A s} \begin{bmatrix} 0 & I \end{bmatrix} e^{A_R s} \right\} \equiv -e^{-A s} B_1 C_R e^{A_R s}, \tag{D.3}$$

$$\frac{d}{ds}\left\{ e^{A^* s} \begin{bmatrix} I & 0 \end{bmatrix} e^{A_Q s} \right\} \equiv e^{A^* s} C_1^* C_Q e^{A_Q s}, \tag{D.4}$$

$$\frac{d}{ds}\left\{ e^{-A s} \begin{bmatrix} 0 & I \end{bmatrix} e^{A_Q s} \right\} \equiv e^{-A s} B_1 C_Q e^{A_Q s}, \tag{D.5}$$

$$\frac{d}{ds}\left\{e^{As}\begin{bmatrix} 0 & I \end{bmatrix} e^{-A_Q s}\right\} \equiv -e^{As}B_1 C_Q e^{-A_Q s}, \quad \text{and} \tag{D.6}$$

$$\int_0^\tau f(s \bigvee t)ds \equiv \int_0^t f(t)ds + \int_t^\tau f(s)ds. \tag{D.7}$$

The kernels of the adjoints ($\hat{B}_1^*, \hat{C}_1^*, \hat{D}_{11}^*, \hat{D}_{12}^*$) of the operators $(\hat{B}_1, \hat{C}_1, \hat{D}_{11}, \hat{D}_{12})$ are:

$$b_1^*(s) := B_1^* e^{A^*(\tau - s)}, \quad c_1^*(s) := e^{A^* s} C_1^*, \quad d_{12}^*(s) := B_2^* \Psi(s)^* C_1^* + D_{12}^*,$$

$$\text{and} \quad d_{11}^*(t,s) := B_1^* e^{A^*(s-t)} 1(s-t) C_1^* + D_{11}^* \delta(s-t). \tag{D.8}$$

Note that depending on whether an operator, $\hat{H}$, maps to time functions ($h(t,s)$ is the kernel) or to constants ($h(s)$ is the kernel) we have either

$$\hat{H}f(t) = \int_0^\tau h(t,s)f(s)ds \quad \text{or} \quad \hat{H}f(t) = \int_0^\tau h(s)f(s)ds. \tag{D.9}$$

D.2 Explicit Formulas for the Various Operators

By inspection of (5.10) the composite operators we must find explicit formulas for are: $\hat{B}_1\hat{R}\hat{B}_1^*$, $\hat{C}_1^*\hat{Q}\hat{C}_1$, $\hat{B}_1\hat{D}_{11}^*\hat{Q}\hat{C}_1$, $\hat{C}_1^*\hat{Q}\hat{D}_{12}$, $\hat{D}_{12}^*\hat{Q}\hat{D}_{12}$, and $\hat{B}_1\hat{D}_{11}^*\hat{Q}\hat{D}_{12}$.

D.2.1 Explicit formula for $\hat{B}_1\hat{R}\hat{B}_1^*$

By definition we have

$$\begin{aligned}(\hat{R}\hat{B}_1^*)(t) = C_R\Bigg\{ & -e^{A_R t}\begin{bmatrix} \Gamma_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix}\int_0^\tau e^{A_R(\tau-s)}B_R B_1^* e^{A^*(\tau-s)}ds \\ & + \int_0^t e^{A_R(t-s)}B_R B_1^* e^{A^*(\tau-s)}ds\Bigg\} + D_R B_1^* e^{A^*(\tau-t)} \\ = C_R\Bigg\{ & -e^{A_R t}\begin{bmatrix} \Gamma_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix} e^{A_R\tau}\left(e^{-A_R s}\begin{bmatrix} I \\ 0 \end{bmatrix} e^{-A^* s}\right)\Bigg|_0^\tau e^{A^*\tau}\end{aligned}$$

$$+e^{A_R t}\left(e^{-A_R s}\begin{bmatrix} I \\ 0 \end{bmatrix} e^{-A^* s}\right)\Big|_0^t e^{A^* \tau}\Bigg\} + D_R B_1^* e^{A^*(\tau - t)}. \quad \text{(D.10)}$$

Noting that

$$\begin{bmatrix} \Gamma_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \Gamma_{11}(\tau) & \Gamma_{12}(\tau) \\ \Gamma_{21}(\tau) & \Gamma_{22}(\tau) \end{bmatrix}\begin{bmatrix} I \\ 0 \end{bmatrix} = \begin{bmatrix} I \\ 0 \end{bmatrix} \text{ and } C_R\begin{bmatrix} I \\ 0 \end{bmatrix} = -D_R B_1^*, \quad \text{(D.11)}$$

we have

$$(\hat{R}\hat{B}_1^*)(t) = -C_R e^{A_R t}\begin{bmatrix} \Gamma_{11}^{-1}(\tau) \\ 0 \end{bmatrix}. \quad \text{(D.12)}$$

Now we can compute the desired composite operator

$$\begin{aligned} \hat{B}_1\hat{R}\hat{B}_1^* &= -\int_0^\tau e^{A(\tau - s)} B_1 C_R e^{A_R s} ds \begin{bmatrix} \Gamma_{11}^{-1}(\tau) \\ 0 \end{bmatrix} \\ &= -e^{A\tau}\left(e^{-As}\begin{bmatrix} 0 & I \end{bmatrix} e^{A_R s}\right)\Big|_0^\tau \begin{bmatrix} \Gamma_{11}^{-1}(\tau) \\ 0 \end{bmatrix}. \end{aligned} \quad \text{(D.13)}$$

Expanding and simplifying, we have

$$\hat{B}_1\hat{R}\hat{B}_1^* = \Gamma_{21}(\tau)\Gamma_{11}^{-1}(\tau). \quad \text{(D.14)}$$

D.2.2 Explicit formula for $\hat{C}_1^*\hat{Q}\hat{C}_1$

$$\begin{aligned} (\hat{Q}\hat{C}_1)(t) &= C_Q\Bigg\{-e^{A_Q t}\begin{bmatrix} \Upsilon_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix}\int_0^\tau e^{A_Q(\tau - s)} B_Q C_1 e^{As} ds \\ &\qquad + \int_0^t e^{A_Q(t-s)} B_Q C_1 e^{As} ds\Bigg\} + D_Q C_1 e^{At} \\ &= C_Q\Bigg\{-e^{A_Q t}\begin{bmatrix} \Upsilon_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix} e^{A_Q\tau}\left(e^{-A_Q s}\begin{bmatrix} 0 \\ I \end{bmatrix} e^{As}\right)\Big|_0^\tau \\ &\qquad + e^{A_Q t}\left(e^{-A_Q s}\begin{bmatrix} 0 \\ I \end{bmatrix} e^{As}\right)\Big|_0^t\Bigg\} + D_Q C_1 e^{At} \\ &= C_Q\Bigg\{-e^{A_Q t}\begin{bmatrix} \Upsilon_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix}\left(\begin{bmatrix} 0 \\ I \end{bmatrix} e^{A\tau} - e^{A_Q\tau}\begin{bmatrix} 0 \\ I \end{bmatrix}\right) \end{aligned}$$

$$+\left[\begin{array}{c}0\\I\end{array}\right]e^{At}-e^{A_Qt}\left[\begin{array}{c}0\\I\end{array}\right]\Big\}+D_QC_1e^{At}, \quad \text{(D.15)}$$

which, noting that $C_Q\left[\begin{array}{c}0\\I\end{array}\right]=-D_QC_1$, simplifies to

$$(\hat{Q}\hat{C}_1)(t)=C_Q\ e^{A_Qt}\left[\begin{array}{c}\Upsilon_{11}^{-1}(\tau)\Upsilon_{12}(\tau)\\-I\end{array}\right]. \quad \text{(D.16)}$$

Now, we have

$$\hat{C}_1^*\hat{Q}\hat{C}_1=\int_0^\tau e^{A^*s}C_1^*C_Q\ e^{A_Qs}ds\left[\begin{array}{c}\Upsilon_{11}^{-1}(\tau)\Upsilon_{12}(\tau)\\-I\end{array}\right]$$

$$=\left(e^{A^*\tau}\left[\begin{array}{cc}I&0\end{array}\right]\left[\begin{array}{cc}\Upsilon_{11}(\tau)&\Upsilon_{12}(\tau)\\\Upsilon_{21}(\tau)&\Upsilon_{22}(\tau)\end{array}\right]-\left[\begin{array}{cc}I&0\end{array}\right]\right)\left[\begin{array}{c}\Upsilon_{11}^{-1}(\tau)\Upsilon_{12}(\tau)\\-I\end{array}\right]. \quad \text{(D.17)}$$

Noting that $\left[\begin{array}{cc}I&0\end{array}\right]\left[\begin{array}{cc}\Upsilon_{11}(\tau)&\Upsilon_{12}(\tau)\\\Upsilon_{21}(\tau)&\Upsilon_{22}(\tau)\end{array}\right]\left[\begin{array}{c}\Upsilon_{11}^{-1}(\tau)\Upsilon_{12}(\tau)\\-I\end{array}\right]\equiv 0$, we have

$$\hat{C}_1^*\hat{Q}\hat{C}_1=-\Upsilon_{11}^{-1}(\tau)\Upsilon_{12}(\tau). \quad \text{(D.18)}$$

D.2.3 Explicit formula for $\hat{B}_1\hat{D}_{11}^*\hat{Q}\hat{C}_1$

Using (D.16), we have

$$(\hat{D}_{11}^*\hat{Q}\hat{C}_1)(t)=\left\{\int_t^\tau B_1^*e^{A^*(s-t)}C_1^*C_Qe^{A_Qs}ds+D_{11}^*C_Qe^{A_Qt}\right\}\left[\begin{array}{c}\Upsilon_{11}^{-1}(\tau)\Upsilon_{12}(\tau)\\-I\end{array}\right]$$

$$=\left\{B_1^*e^{-A^*t}\left(e^{A^*s}\left[\begin{array}{cc}I&0\end{array}\right]e^{A_Qs}\right)\Big|_t^\tau+D_{11}^*C_Qe^{A_Qt}\right\}\left[\begin{array}{c}\Upsilon_{11}^{-1}(\tau)\Upsilon_{12}(\tau)\\-I\end{array}\right], \quad \text{(D.19)}$$

which, using the fact that $D_{11}^*C_Q-B_1^*\left[\begin{array}{cc}I&0\end{array}\right]=-C_Q$, gives

$$(\hat{D}_{11}^*\hat{Q}\hat{C}_1)(t)=-C_Qe^{A_Qt}\left[\begin{array}{c}\Upsilon_{11}^{-1}(\tau)\Upsilon_{12}(\tau)\\-I\end{array}\right]. \quad \text{(D.20)}$$

Thus, the desired result is given by

$$\hat{B}_1\hat{D}_{11}^*\hat{Q}\hat{C}_1=-\int_0^\tau e^{A(\tau-s)}B_1C_Qe^{A_Qs}ds\left[\begin{array}{c}\Upsilon_{11}^{-1}(\tau)\Upsilon_{12}(\tau)\\-I\end{array}\right]$$

$$= -\left(\begin{bmatrix} 0 & I \end{bmatrix} e^{A_Q\tau} - e^{A\tau}\begin{bmatrix} 0 & I \end{bmatrix}\right)\begin{bmatrix} \Upsilon_{11}^{-1}(\tau)\Upsilon_{12}(\tau) \\ -I \end{bmatrix}, \tag{D.21}$$

which gives

$$\hat{B}_1\hat{D}_{11}^*\hat{Q}\hat{C}_1 = \Upsilon_{22}(\tau) - \Upsilon_{21}(\tau)\Upsilon_{11}^{-1}(\tau)\Upsilon_{12}(\tau) - e^{A\tau}. \tag{D.22}$$

D.2.4 Explicit formula for $\hat{C}_1^*\hat{Q}\hat{D}_{12}$

First, we must compute the first composite operator

$$\begin{aligned}
(\hat{Q}\hat{D}_{12})(t) = C_Q\Bigg\{&-e^{A_Qt}\begin{bmatrix} \Upsilon_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix}\int_0^\tau e^{A_Q(\tau-s)}B_QC_1\int_0^s e^{A(s-r)}drB_2ds \\
&+\int_0^t e^{A_Q(t-s)}B_QC_1\int_0^s e^{A(s-r)}drB_2ds\Bigg\} \\
&+D_QC_1\Psi(t)B_2 + \int_0^\tau q(t,s)dsD_{12} \\
= C_Q\Bigg\{&-e^{A_Qt}\begin{bmatrix} \Upsilon_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix}\int_0^\tau\int_0^\tau e^{A_Q(\tau-s)}B_QC_1e^{A(s-r)}\mathbf{1}(s-r)B_2dsdr \\
&+\int_0^\tau\int_0^\tau e^{A_Q(t-s)}B_QC_1e^{A(s-r)}B_2\mathbf{1}(t-s)\mathbf{1}(s-r)dsdr\Bigg\} \\
&-C_Q\left\{e^{A_Qt}\begin{bmatrix} \Upsilon_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix}\int_0^\tau e^{A_Q(\tau-r)}dr - \int_0^t e^{A_Q(t-r)}dr\right\}B_QD_{12} \\
&+D_QD_{12} + D_QC_1\Psi(t)B_2 \\
= C_Q\Bigg\{&-e^{A_Qt}\begin{bmatrix} \Upsilon_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix}\int_0^\tau e^{A_Q\tau}\left(e^{-A_Qs}\begin{bmatrix} 0 \\ I \end{bmatrix}e^{As}\right)\Bigg|_r^\tau e^{-Ar}B_2dr \\
&+\int_0^t e^{A_Qt}\left(e^{-A_Qs}\begin{bmatrix} 0 \\ I \end{bmatrix}e^{As}\right)\Bigg|_r^t e^{-Ar}B_2dr\Bigg\} \\
&-C_Q\left\{e^{A_Qt}\begin{bmatrix} \Upsilon_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix}\int_0^\tau e^{A_Q(\tau-r)}dr - \int_0^t e^{A_Q(t-r)}dr\right\}B_QD_{12} \\
&+D_QD_{12} + D_QC_1\Psi(t)B_2,
\end{aligned} \tag{D.23}$$

which, after some algebra, gives

$$(\hat{Q}\hat{D}_{12})(t) = C_Q \left\{ e^{A_Q t} \begin{bmatrix} \Upsilon_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix} \int_0^\tau e^{A_Q(\tau - r)} dr - \int_0^t e^{A_Q(t-r)} dr \right\} \mathcal{B}_Q + D_Q D_{12}. \quad \text{(D.24)}$$

Now, we can write

$$\begin{aligned} \hat{C}_1^* \hat{Q} \hat{D}_{12} &= \int_0^\tau e^{A^* t} dt C_1^* D_Q D_{12} + \left\{ \int_0^\tau e^{A^* t} C_1^* C_Q e^{A_Q t} dt \begin{bmatrix} \Upsilon_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix} \Phi(\tau) \right. \\ &\qquad \left. - \int_0^\tau \int_0^\tau e^{A^* t} C_1^* C_Q e^{A_Q(t-r)} \mathbf{1}(t-r) dt dr \right\} \mathcal{B}_Q \\ &= \Psi^*(\tau) C_1^* D_Q D_{12} + \left\{ \left(e^{A^* s} \begin{bmatrix} I & 0 \end{bmatrix} e^{A_Q s} \right) \Big|_0^\tau \begin{bmatrix} \Upsilon_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix} \Phi(\tau) \right. \\ &\qquad \left. - \int_0^\tau \int_r^\tau e^{A^* t} C_1^* C_Q e^{A_Q t} dt e^{-A_Q r} dr \right\} \mathcal{B}_Q \\ &= \Psi^*(\tau) C_1^* D_Q D_{12} + \left\{ e^{A^* \tau} \begin{bmatrix} I & 0 \end{bmatrix} \Phi(\tau) - \begin{bmatrix} \Upsilon_{11}^{-1}(\tau) & 0 \end{bmatrix} \Phi(\tau) \right. \\ &\qquad \left. - e^{A^* \tau} \begin{bmatrix} I & 0 \end{bmatrix} \int_0^\tau e^{A_Q(\tau - r)} dr \right\} \mathcal{B}_Q. \quad \text{(D.25)} \end{aligned}$$

So, we have

$$\hat{C}_1^* \hat{Q} \hat{D}_{12} = -\Upsilon_{11}^{-1}(\tau) \begin{bmatrix} \Phi_{11}(\tau) & \Phi_{12}(\tau) \end{bmatrix} \mathcal{B}_Q. \quad \text{(D.26)}$$

D.2.5 Explicit formula for $\hat{D}_{12}^* \hat{Q} \hat{D}_{12}$

The derivation of this one is a bit more subtle and involved than the preceding ones, so we will give a little more detail at the outset. By definition we have

$$\begin{aligned} \hat{D}_{12}^* \hat{Q} \hat{D}_{12} &:= \int_0^\tau d_{12}^*(t) (\hat{Q} \hat{D}_{12})(t) dt \\ &= \int_0^\tau \left\{ B_2^* \int_0^t e^{A^*(t-s)} ds C_1^* + D_{12}^* \right\} (\hat{Q} \hat{D}_{12})(t) dt. \end{aligned} \quad \text{(D.27)}$$

Let $\mathcal{D}_{12} := \int_0^\tau D_{12}^* (\hat{Q} \hat{D}_{12})(t) dt$. Then, using (D.24), we have

$$\mathcal{D}_{12} = D_{12}^* C_Q \left\{ \int_0^\tau e^{A_Q t} dt \begin{bmatrix} \Upsilon_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix} \Phi(\tau) - \int_0^\tau \int_0^t e^{A_Q(t-r)} dr dt \right\} \mathcal{B}_Q + \int_0^\tau D_{12}^* D_Q D_{12} dt$$

$$= D_{12}^* C_Q \left\{ \Phi(\tau) \begin{bmatrix} \Upsilon_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix} \Phi(\tau) - \Omega(\tau) \right\} \mathcal{B}_Q + \tau D_{12}^* D_Q D_{12}. \qquad \text{(D.28)}$$

Now, using (D.24), we can write

$$\begin{aligned}
\hat{D}_{12}^* \hat{Q} \hat{D}_{12} &= \mathcal{D}_{12} + B_2^* \int_0^\tau \Psi^*(t) dt C_1^* D_Q D_{12} \\
&\quad + B_2^* \int_0^\tau e^{-A^* s} \int_s^\tau e^{A^* t} C_1^* C_Q e^{A_Q t} dt ds \begin{bmatrix} \Upsilon_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix} \Phi(\tau) \mathcal{B}_Q \\
&\quad - B_2^* \int_0^\tau \int_0^\tau e^{-A^* s} \int_0^\tau e^{A^* t} C_1^* C_Q e^{A_Q t} 1(t-s) 1(t-r) dt e^{-A_Q r} dr ds \mathcal{B}_Q \\
&= \mathcal{D}_{12} + B_2^* \int_0^\tau \Psi^*(t) dt C_1^* D_Q D_{12} \\
&\quad + B_2^* \int_0^\tau e^{-A^* s} \left(e^{A^* t} \begin{bmatrix} I & 0 \end{bmatrix} e^{A_Q t} \right) \Big|_s^\tau ds \begin{bmatrix} \Upsilon_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix} \Phi(\tau) \mathcal{B}_Q \\
&\quad - B_2^* \int_0^\tau \int_0^\tau e^{-A^* s} \left(e^{A^* t} \begin{bmatrix} I & 0 \end{bmatrix} e^{A_Q t} \right) \Big|_{s \vee r}^\tau e^{-A_Q r} dr ds \mathcal{B}_Q \\
&= \mathcal{D}_{12} + B_2^* \int_0^\tau \Psi^*(t) dt C_1^* D_Q D_{12} \\
&\quad + B_2^* \int_0^\tau e^{A^*(\tau - s)} ds \begin{bmatrix} I & 0 \end{bmatrix} \Upsilon(\tau) \begin{bmatrix} \Upsilon_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix} \Phi(\tau) \mathcal{B}_Q \\
&\quad - B_2^* \begin{bmatrix} I & 0 \end{bmatrix} \Phi(\tau) \begin{bmatrix} \Upsilon_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix} \Phi(\tau) \mathcal{B}_Q \\
&\quad - B_2^* \int_0^\tau e^{A^*(\tau - s)} ds \begin{bmatrix} I & 0 \end{bmatrix} \int_0^\tau e^{A_Q(\tau - r)} dr \mathcal{B}_Q \\
&\quad + B_2^* \int_0^\tau \int_0^\tau e^{-A^* s} e^{A^*(s \vee r)} \begin{bmatrix} I & 0 \end{bmatrix} e^{A_Q(s \vee r)} e^{-A_Q r} dr ds \mathcal{B}_Q. \qquad \text{(D.29)}
\end{aligned}$$

Noting that $\begin{bmatrix} I, & 0 \end{bmatrix} \Upsilon(\tau) \begin{bmatrix} \Upsilon_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} I & 0 \end{bmatrix}$, we have

$$\hat{D}_{12}^{*}\hat{Q}\hat{D}_{12} = \mathcal{D}_{12} - B_2^{*}\left[\begin{array}{cc}\Phi_{11}(\tau) & \Phi_{12}(\tau)\end{array}\right]\left[\begin{array}{cc}\Upsilon_{11}^{-1}(\tau) & 0\\ 0 & 0\end{array}\right]\Phi(\tau)\mathcal{B}_Q$$

$$+B_2^{*}\left[\begin{array}{cc}I & 0\end{array}\right]\int_0^{\tau}\int_0^{s} e^{A^{*}(s-r)}drds\mathcal{B}_Q + B_2^{*}\int_0^{\tau}\Psi^{*}(t)dtC_1^{*}D_Q D_{12}. \quad \text{(D.30)}$$

Thus, after some algebra, we have

$$\hat{D}_{12}^{*}\hat{Q}\hat{D}_{12} = \left[\begin{array}{cc}\mathcal{B}_Q^{L} & \mathcal{B}_Q^{R}\end{array}\right]\left\{\Phi(\tau)\left[\begin{array}{cc}\Upsilon_{11}^{-1}(\tau) & 0\\ 0 & 0\end{array}\right]\Phi(\tau) - \Omega(\tau)\right\}\mathcal{B}_Q$$

$$+\tau D_{12}^{*}D_Q D_{12}. \quad \text{(D.31)}$$

D.2.6 Explicit formula for $\hat{B}_1\hat{D}_{11}^{*}\hat{Q}\hat{D}_{12}$

The derivation of the last composite operator is even more involved.

Define $\mathcal{D}_{11}(t) := \int_0^{\tau} D_{11}^{*}(\hat{Q}\hat{D}_{12})(t)\delta(s-t)ds = D_{11}^{*}(\hat{Q}\hat{D}_{12})(t)$, which gives

$$\mathcal{D}_{11}(t) = D_{11}^{*}C_Q\left\{e^{A_Q t}\left[\begin{array}{cc}\Upsilon_{11}^{-1}(\tau) & 0\\ 0 & 0\end{array}\right]\Phi(\tau) - \int_0^{t} e^{A_Q(t-r)}dr\right\}\mathcal{B}_Q + D_{11}D_Q D_{12}. \quad \text{(D.32)}$$

Using this definition, we have

$$(\hat{D}_{11}^{*}\hat{Q}\hat{D}_{12})(t) = \mathcal{D}_{11}(t) + B_1^{*}\int_t^{\tau} e^{A^{*}(s-t)}dsC_1^{*}D_Q D_{12}$$

$$+B_1^{*}\left\{e^{-A^{*}t}\int_t^{\tau} e^{A^{*}s}C_1^{*}C_Q e^{A_Q s}ds\left[\begin{array}{cc}\Upsilon_{11}^{-1}(\tau) & 0\\ 0 & 0\end{array}\right]\Phi(\tau)\right.$$

$$\left.-\int_0^{\tau} e^{-A^{*}t}\int_{t\vee r}^{\tau} e^{A^{*}s}C_1^{*}C_Q e^{A_Q s}dse^{-A_Q r}dr\right\}\mathcal{B}_Q$$

$$= \mathcal{D}_{11}(t) + B_1^{*}\int_t^{\tau} e^{A^{*}(s-t)}dsC_1^{*}D_Q D_{12}$$

$$+B_1^{*}e^{A^{*}(\tau-t)}\left[\begin{array}{cc}I & 0\end{array}\right]\Upsilon(\tau)\left[\begin{array}{cc}\Upsilon_{11}^{-1}(\tau) & 0\\ 0 & 0\end{array}\right]\Phi(\tau)\mathcal{B}_Q$$

$$-B_1^{*}\left[\begin{array}{cc}I & 0\end{array}\right]e^{A_Q t}\left[\begin{array}{cc}\Upsilon_{11}^{-1}(\tau) & 0\\ 0 & 0\end{array}\right]\Phi(\tau)\mathcal{B}_Q$$

$$-B_1^{*}e^{A^{*}(\tau-t)}\left[\begin{array}{cc}I & 0\end{array}\right]\Phi(\tau)\mathcal{B}_Q$$

$$+B_1^* e^{-A^* t} \int_0^\tau e^{A^*(t \vee \tau)} \begin{bmatrix} I & 0 \end{bmatrix} e^{A_Q(t \vee \tau)} e^{-A_Q \tau} d\tau \mathcal{B}_Q. \quad \text{(D.33)}$$

After making cancellations, as in the previous subsection, we have

$$(\hat{D}_{11}^* \hat{Q} \hat{D}_{12})(t) = \mathcal{D}_{11}(t) - B_1^* \begin{bmatrix} I & 0 \end{bmatrix} e^{A_Q t} \begin{bmatrix} \Upsilon_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix} \Phi(\tau) \mathcal{B}_Q$$
$$+B_1^* \begin{bmatrix} I & 0 \end{bmatrix} \int_0^t e^{A_Q(t-\tau)} d\tau \mathcal{B}_Q. \quad \text{(D.34)}$$

Once again omitting the algebra, we have

$$(\hat{D}_{11}^* \hat{Q} \hat{D}_{12})(t) = \mathcal{C}_Q \left\{ \int_0^t e^{A_Q(t-\tau)} d\tau - e^{A_Q t} \begin{bmatrix} \Upsilon_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix} \Phi(\tau) \right\} \mathcal{B}_Q$$
$$+D_{11}^* D_Q D_{12}. \quad \text{(D.35)}$$

Now, we can compute the final desired composite operator

$$\begin{aligned} \hat{B}_1 \hat{D}_{11}^* \hat{Q} \hat{D}_{12} &= \int_0^\tau e^{A\tau} \int_\tau^\tau e^{-As} B_1 C_Q e^{A_Q s} ds e^{-A_Q \tau} d\tau \mathcal{B}_Q \\ &\quad -e^{A\tau} \int_0^\tau e^{-As} B_1 C_Q e^{A_Q s} ds \begin{bmatrix} \Upsilon_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix} \Phi(\tau) \mathcal{B}_Q \\ &\quad + \int_0^\tau e^{As} ds B_1 D_{11}^* D_Q D_{12} \\ &= \begin{bmatrix} 0 & I \end{bmatrix} \Phi(\tau) \mathcal{B}_Q - \Psi(\tau) \begin{bmatrix} 0 & I \end{bmatrix} \mathcal{B}_Q \\ &\quad - \begin{bmatrix} 0 & I \end{bmatrix} \Upsilon(\tau) \begin{bmatrix} \Upsilon_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix} \Phi(\tau) \mathcal{B}_Q + \Psi(\tau) B_1 D_{11}^* D_Q D_{12}. \quad \text{(D.36)} \end{aligned}$$

Thus, we have

$$\hat{B}_1 \hat{D}_{11}^* \hat{Q} \hat{D}_{12} = \begin{bmatrix} (\Phi_{21}(\tau) - \Upsilon_{21}(\tau) \Upsilon_{11}^{-1}(\tau) \Phi_{11}(\tau)) \\ (\Phi_{22}(\tau) - \Upsilon_{21}(\tau) \Upsilon_{11}^{-1}(\tau) \Phi_{12}(\tau)) \end{bmatrix}^T \mathcal{B}_Q - \Psi(\tau) B_2. \quad \text{(D.37)}$$

D.3 Summary of proof

The proof is essentially complete. All that remains is to plug the expressions for the above operators into the discrete-time equivalent system representation given in Theorem 5.5.1 and do a little arithmetic.

BIBLIOGRAPHY

[1] K. J. Åstrom, P. Hagander, and J. Sternby. Zeros of sampled-data systems. *IEEE Transactions on Automatic Control*, 20(1):31–38, 1984.

[2] Karl Åstrom and Bjorn Wittenmark. *Adaptive Control.* Addison-Wesley, New York, NY, 1989.

[3] Gary J. Balas, John C. Doyle, Keith Glover, Packard Andy, and Roy Smith. *μ-Analysis and Synthesis Toolbox: User's Guide.* The Mathworks Inc., Natick, MA, 1991.

[4] Bassam Bamieh, J. Boyd Person, Bruce A. Francis, and Allen Tannenbaum. A lifting technique for linear periodic systems with applications to sample-data control. *Systems & Control Letters*, 17:79–88, 1991.

[5] Bassam A. Bamieh and J. Boyd Pearson Jr. A general framework for linear periodic systems with applications to H^∞ sampled-data control. *IEEE Transactions on Automatic Control*, 37(4):418–435, 1992.

[6] Tamer Basar. Optimal H^∞ designs under sampled state measurements. *Systems & Control Letters*, 16:399–409, 1991.

[7] Pierre Bernhard. Application of the min-max certainty equivalence principle to the sampled-data output feedback H^∞ control problem. *Systems & Control Letters*, 16:229–234, 1991.

[8] Stephen P. Boyd and Craig H. Barratt. *Linear Control Design: Limits of Performance.* Information and Systems Sciences Series. Prentice Hall, Englewood Cliffs, NJ, 1991.

[9] Stephen P. Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan. *Linear Matrix Inequalities in Systems and Control Theory.* Studies in Applied Mathematics. SIAM, Philidelphia, PA, 1994.

[10] D. Bugajski, D. Enns, and M. Elgersma. A dynamic inversion based control law with application to the high angle of attack research vehicle. In *Procedings of the AIAA Guidance Navigation and Control Conference*, Portland, OR, 1990.

[11] Ben M. Chen, Ali Saberi, and Yacov Shamash. A non-recursive method for solving the general discrete-time Riccati equations related to the H^∞ control problem. In *Proceedings of the American Control Conference*, pages 2649–2653, San Francisco, California, 1993.

[12] Chi-Tsong Chen. *Linear System Theory and Design.* Holt, Reinhart, and Winston, Orlando, FL, 1984.

[13] M. J. Chen and C. A. Desoer. Necessary and sufficient condition for robust stability of linear distributed feedback systems. *International Journal of Control*, 35:255–267, 1982.

[14] Tongwen Chen. A simple derivation of the L^2-optimal sampled-data controllers. *Systems & Control Letters*, 20:49–56, 1993.

[15] Tongwen Chen and Bruce Francis. On the L^2-induced norm of a sampled-data system. *Systems & Control Letters*, 15:211–219, 1990.

[16] Tongwen Chen and Bruce A. Francis. H^2-optimal sampled-data control. *IEEE Transactions on Automatic Control*, 36(4):387–397, 1991.

[17] Tongwen Chen and Bruce A. Francis. Input-output stability of sampled-data systems. *IEEE Transactions on Automatic Control*, 36(1):50–58, 1991.

[18] Tongwen Chen and Bruce A. Francis. Linear time-varying H_2-optimal control of sampled-data systems. *Automatica*, 27(6):963–974, 1991.

[19] Tongwen Chen and Bruce A. Francis. Sample-data optimal design and robust stabilization. *ASME Transactions, Journal of Dynamic Systems, Measurement, and Control*, 114:538–543, 1992.

[20] K. K. Chew and M. Tomizuka. Digital control of repetitive errors in disk drive system. *IEEE Control Systems Magazine*, 10(1):16–20, 1990.

[21] Cheng-Chih Chu. On discrete inner-outer and spectral factorizations. In *Proceedings of the American Control Conference*, pages 1699–1700, Atlanta, GA, 1988.

[22] J. Doyle, Francis B., and A. Tannenbaum. *Feedback Control Theory.* McMillan, New York, NY, 1992.

[23] John C. Doyle. Analysis of feedback systems with structured uncertainties. *IEE Proceedings*, 129(6):242–250, 1982. Part D.

[24] John C. Doyle, Keith Glover, Pramod P. Khargonekar, and Bruce A. Francis. State-space solutions to standard H_2 and H^∞ control problems. *IEEE Transactions on Automatic Control*, 34(8):831–847, 1989.

[25] Geir Dullerud and Keith Glover. Robust stabilization of sampled-data systems to structured LTI perturbations. *IEEE Transactions on Automatic Control*, 38(10):1497–1508, 1993.

[26] M. R. Elgersma. *Control of Nonlinear Systems Using Partial Dynamic Inversion.* PhD thesis, University of Minnesota, Minneapolis, MN, 1988.

[27] Ian J. Fialho and Tryphon T. Georgiou. On stability and performance of sampled-data systems subject to wordlength constraint. In *Proceedings of the 32nd IEEE Conference on Decision and Control*, pages 309–314, San Antonio, Texas, 1993.

[28] D. S. Flamm and H. Yang. H^∞ mixed sensitivity design for a real multivariable delay problem. In *Proceedings of the American Control Conference*, pages 2202–2207, Boston, MA, 1991.

[29] C. Foias and A. E. Frazho. *The Commutant Lifting Approach to Interpolation Problems.* Birkhäuser, Basel, 1990.

[30] C. Foias and A. Tannenbaum. On the four block problem, I. *Operator Theory: Advances and Applications*, 32:93–112, 1988.

[31] C. Foias, A. Tannenbaum, and G. Zames. Some explicit formulae for the singular values of a certain hankel operators with factorizable symbol. *SIAM Journal of Mathematical Analysis*, 19:1081–1091, 1988.

[32] B. A. Francis and G. Zames. On H^∞ optimal sensitivity theory for SISO feedback systems. *IEEE Transactions on Automatic Control*, 29:9–16, 1984.

[33] Bruce A. Francis. *A Course in H^∞ Control Theory.* Lecture Notes in Control and Information Sciences. Springer-Verlag, New York, NY, 1987.

[34] Bruce A. Francis and Tryphon T. Georgiou. Stability theory for linear time-invariant plants with periodic digital controllers. *IEEE Transactions on Automatic Control*, 33(9):820–832, 1988.

[35] Gene F. Franklin, J. David Powell, and Abbas Enami-Naeini. *Feedback Control of Dynamic Systems.* Addison Wesley, Reading, MA, 1986.

[36] P Gahinet. A convex parameterization of suboptimal H^∞ controllers. In *Proceedings of the 31st IEEE Conference on Decision and Control*, pages 937–942, Tuscon, AZ, 1992.

[37] Keith Glover. All optimal hankel-norm approximations of linear multivariable systems and their l^∞-error bounds. *International Journal of Control*, 39(6):1115–1193, 1984.

[38] David E. Goldberg. *Genetic Algoriihms in Search, Optimization, and Machine Learning.* Addison-Wesley, New York, NY, 1989.

[39] S. Hara, Y. Yamamoto, T. Omata, and M. Nakano. Repetitive control system: a new type servo system for periodic exogenous signals. *IEEE Transactions on Automatic Control*, 33:659–667, 1988.

[40] Shinji Hara, Hiroyuki Kawamura, and Hak-Kyung Sung. High-precision tracking by digital repetitive control scheme with periodic-type hold function. In *Proceedings of the American Control Conference*, pages 2676–2680, Chicago, Illinois, 1992.

[41] Kenneth Hoffman. *Banach Spaces of Analytic Functions.* Dover Press (Prentice-Hall, 1962), New York, NY (Englewood Cliffs, NJ), 1988.

[42] John Holland. *Adaptation in Natural and Artificial Systems.* The University of Michican Press, Ann Arbor, 1975.

[43] R. Horn and C. Johnson. *Matrix Analysis.* Cambridge University Press, New York, 1985.

[44] P. A. Iglesias and Keith Glover. State-space approach to discrete-time H^∞ control. *International Journal of Control*, 54(5):1031–1073, 1991.

[45] T. Iwasaki and R. E. Skelton. All controllers for the general H^∞ control problem: LMI existance conditions and state space formulas. *Automatica*, to appear, 1994.

[46] Pierre T. Kabamba and Shinji Hara. Worst-case analysis and design of sampled-data control systems. *IEEE Transactions on Automatic Control*, 38(9):1337–1357, 1993.

[47] Charles Karr and Edward Gentry. Fuzzy control of pH using genetic algorithms. *IEEE Transactions on Fuzzy Systems*, 1(1):46–53, 1993.

[48] T. Katayama. A simple state-space approach to spectral factorization for discrete-time systems. *Control Theory and Advanced Technology*, 8(3):647–657, 1992.

[49] Hassan K. Khalil. *Nonlinear Systems.* Macmillian Publishing Company, New York, NY, 1992.

[50] Pramod P. Khargonekar and N. Sivashankar. L^2 optimal control for sampled-data systems. *Systems & Control Letters*, 17:425–436, 1991.

[51] F. Kobayashi, S. Hara, and H. Tanaka. Reduction of motor speed fluctuation using repetitive control. In *Proceedings of the 29th IEEE Conference on Decision and Control*, pages 1697–1702, Honolulu, Hawaii, 1990.

[52] Kristinn Kristinsson and Guy Dumont. System identification and control using genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(5):1033–1046, 1992.

[53] Bejamin C. Kuo. *Automatic Control Systems.* Prentice-Hall, Englewood Cliffs, NJ, 1982.

[54] L. L. Porter II and K. M. Passino. Genetic model reference adaptive control. In *Proc. IEEE International Symposium on Intelligent Control*, Columbus, OH., August 16-18, 1994.

[55] Alireza Langari and Bruce A. Francis. Sampled-data repetitive control systems. In *Proceedings of the American Control Conference*, pages 3234–3235, Baltimore, MD, 1994.

[56] S. R. Lee and K. Srinivasan. On-line identification of process models in closed-loop material testing. *ASME Transactions, Journal of Dynamic Systems, Measurement, and Control*, 111:172–179, 1989.

[57] G. M. H. Leung, T. P. Perry, and B. A. Francis. Performance analysis of sampled-data control systems. *Automatica*, 27(4):699–704, 1991.

[58] Gang Li and Michel Gevers. Comparitive study of finite wordlength effects in shift and delta operator parameterizations. *IEEE Transactions on Automatic Control*, 38(5):803–807, 1993.

[59] Duncan McFarlane and Keith Glover. A loop shaping design procedure using H^∞ synthesis. *IEEE Transactions on Automatic Control*, 37(6):759–769, 1992.

[60] Cleve Moler and Charles Van Loan. Nineteen dubious ways to compute the exponential of a matrix. *SIAM Review*, 20(4):801–836, 1978.

[61] Bruce C. Moore. Principle component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, 26:17–32, 1981.

[62] Aswartha Narayana. State-space approach to the bilinear transformation and some extensions. *IEEE Transactions on Education*, 34(1):139–142, 1991.

[63] Arch W. Naylor and George R. Sell. *Linear Operator Theory in Engineering and Science*, volume 40 of *Apllied Mathematical Sciences*. Springer-Verlag, Newyork, NY, 1982.

[64] Katsuhiko Ogata. *Modern Control Engineering*. Prentice-Hall, Englewood Cliffs, NJ, 1970.

[65] Katsuhiko Ogata. *Discrete-time Control Systems*. Prentice-Hall, Englewood Cliffs, NJ, 1987.

[66] T. Omata, S. Hara, and M. Nakano. Nonlinear repetitive control with application to trajectory control of manipulators. *Journal of Robotic Systems*, 4(5):631–652, 1987.

[67] Scott L. Osburn and Dennis S. Berstein. An exact treatment of the achievable closed-loop H^2 performance of sampled-data controllers: From continuous-time to open-loop. In *Proceedings of the 32nd IEEE Conference on Decision and Control*, pages 325–330, San Antonio, Texas, 1993.

[68] H. Özbay and T. E. Peery. On fixed order controllers for delay systems: Discrete-time case. In *Proceedings of the American Control Conference*, pages 1030–1031, San Francisco, California, 1993.

[69] H. Özbay, M. C. Smith, and A. Tannenbaum. Mixed sensitivity optimization for a class of unstable infinite dimensional systems. *Linear Algebra and its Applications*, 178:43–83, 1993.

[70] H. Özbay and A. Tannenbaum. A skew toeplitz approach to the H^∞ optimal control of multivariable distributed systems. *SIAM Journal of Control and Optimization*, 28:653–670, 1990.

[71] Ü. Özgüner and E. J. Davidson. Sampling and decentralized fixed modes. In *Proceedings of the American Control Conference*, pages 257–262, Boston, MA, 1985.

[72] A. Packard and J. Doyle. The complex structured singular value. *Automatica*, 29(1):71–109, 1993.

[73] T. E. Peery and H. Özbay. On H^∞ optimal repetitive controllers. In *Proceedings of the 32nd IEEE Conference on Decision and Control*, pages 1146–1151, San Antonio, Texas, 1993.

[74] T. E. Peery and H. Özbay. H^∞ optimal sampled-data repetitive control design. In *Procedings of the Thirty-third Annual Allerton Conference on Communication, Control, and Computing*, page : to appear, Monticello, IL, October 1995.

[75] Marvin Rosenblum and James Rovnyak. *Hardy Classes and Operator Theory.* Oxford University Press, New York, NY, 1985.

[76] N. Sadegh, R. Horowitz, W. W. Kao, and M. Tomizuka. A unified approach to the design of adaptive and repetitive controllers for robotic manipulators. *ASME Transactions, Journal of Dynamic Systems, Measurement, and Control*, 112(4):618–629, 1990.

[77] D. Sarason. Generalized interpolation in H^∞. *Transactions of the AMS*, 127:179–203, 1967.

[78] Carsten Scherer. H^∞-optimization without assumptions on finite or infinite zeros. *SIAM Journal of Control and Optimization*, 30(1):143–166, 1992.

[79] F. R. Shaw and K. Srinivasan. Analysis and synthesis of discrete-time repetitive controllers. *ASME Transactions, Journal of Dynamic Systems, Measurement, and Control*, 133:216–222, 1989.

[80] F. R. Shaw and K. Srinivasan. Discrete-time repetitive control system design using the regenerative spectrum. In *Proceedings of the American Control Conference*, pages 2628–2633, Boston, MA, 1991.

[81] N. Sivashankar and Promod P. Khargonekar. Induced norms for sampled-data systems. *Automatica*, 28(6):1267–1272, 1992.

[82] N. Sivashhankar and Pramod P. Khargonekar. Robust stability and performance analysis of sampled-data systems. *IEEE Transactions on Automatic Control*, 38(1):58–69, 1993.

[83] M. Srinivas and Lalit M. Patnaik. Genetic algorithms: A survey. *IEEE Computer*, pages 17–26, 1994.

[84] K. Srinivasan and F. R. Shaw. Analysis and design of repetitive control systems using the regeneration spectrum. In *Proceedings of the American Control Conference*, pages 1150–1155, San Diego, CA, 1990.

[85] A. A. Stoorvogel. The singular H^∞ control problem with dynamic measurement feedback. *SIAM Journal of Control and Optimization*, 29(1):160–184, 1991.

[86] A. A. Stoorvogel. The discrete-time H^∞ control problem with measurement feedback. *SIAM Journal of Control and Optimization*, 30(1):182–202, 1992.

[87] Anton Stoorvogel. *The H^∞ Control Problem: A State Space Approach.* International Series in Systems and Control Engineering. Prentice Hall, New York, NY, 1992.

[88] Anton A. Stoorvogel and Arie J. T. M. Weeren. The discrete-time Riccati equation related to the H^∞ control problem. *IEEE Transactions on Automatic Control*, 39(3):686–691, 1994.

[89] Weiqian Sun, K. M. Nagpal, and Pramod P. Khargonekar. Optimal sampler for H^∞ control. In *Proceedings of the 32nd IEEE Conference on Decision and Control*, pages 777–782, San Antonio, Texas, 1993.

[90] Weiqian Sun, Krishan M. Nagpal, and Pramod P. Khargonekar. Control and filtering for sampled-data systems. *IEEE Transactions on Automatic Control*, 38(8):1162–1175, 1993.

[91] Béla Sz.-Nagy and Ciprian Foiaş. *Harmonic Analysis of Operators on Hilbert Space.* North-Holland Publishing Company (American Elsevier Publishing Company, Inc.), Amstedam (Ney York, NY), 1970.

[92] H. T. Toivonen. Sampled-data control of continuous-time systems with an H^∞ optimal criterion. *Automatica*, 28(1):45–54, 1992.

[93] H. T. Toivonen. Sampled-data H^∞ optimal control of time-varying systems. *Automatica*, 28(4):823–826, 1992.

[94] H. T. Toivonen. Sampling prefilters with an H^∞ criterion. In *Proceedings of the American Control Conference*, pages 697–701, Chicago, Illinois, 1992.

[95] O. Toker and H. Özbay. H^∞ optimal and suboptimal controllers for infinite dimensional siso plants. *IEEE Transactions on Automatic Control*, 40(4):751–755, 1995.

[96] Onur Toker and Hitay Özbay. On the structure of H^∞ controllers. In *Proceedings of the 33nd IEEE Conference on Decision and Control*, pages 3949–3954, Lake Buena Vista, FL, 1994.

[97] M. Tomizuka, T. C. Tsao, and K. K. Chew. Analysis and synthesis of discrete-time repetitive controllers. *ASME Transactions, Journal of Dynamic Systems, Measurement, and Control*, 111(3):353–358, 1989.

[98] T. C. Tsao and M. Nemani. Adaptive and repetitive digital control algorithms for noncircular machining. In *Proceedings of the American Control Conference*, pages 2696–2699, Chicago, Illinois, 1992.

[99] T. C. Tsao and M. Tomizuka. Adaptive and repetitive digital control algorithms for noncircular machining. In *Proceedings of the American Control Conference*, pages 115–120, Atlanta, GA, 1988.

[100] E. Tung, G. Anwar, and M Tomizuka. Low velocity friction compensation and feedforward solution based on repetitive control. In *Proceedings of the American Control Conference*, pages 2615–2620, Boston, MA, 1991.

[101] Bodgan Varšek, Alen and Urbančič, Tanja and Filipič. Genetic algorithms in controller design and tuning. *IEEE Transactions on Systems, Man and Cybernetics*, 23(5):1330–1339, 1993.

[102] M. Vidyasagar. *Control System Synthesis: A Factorization Approach.* MIT Press, Cambridge, MA, 1985.

[103] Petros Voulgaris and Bassam Bamieh. Control of asynchronous sampled-data systems. In *Proceedings of the 32nd IEEE Conference on Decision and Control*, pages 785–792, San Antonio, Texas, 1993.

[104] Yutaka Yamamoto. New approach to sampled-data control systems: A function space method. In *Proceedings of the 29th IEEE Conference on Decision and Control*, pages 1882–1888, Honolulu, Hawaii, 1990.

[105] Yutaka Yamamoto. On the state space and frequency domain characterization of H^∞-norm of sampled data systems. *Systems & Control Letters*, 21:163–172, 1993.

[106] L. Yao and W. A. Sethares. Nonlinear parameter estimation via the genetic algorithm. *IEEE Transactions on Signal Processing*, 42(4):927–935, 1994.