Iterative Learning Control
—a critical review—

This thesis has been completed in partial fulfillment of the requirements of the Dutch Institute of Systems and Control (DISC) for graduate study.

The research described in this thesis was undertaken at the departments of Electrical Engineering and Applied Mathematics, University of Twente, Enschede, with financial support from the 'Duits-Nederlands Mechatronica Innovatie Centrum'.

Mechatronica
Innovatie
Centrum

Printed by Wöhrmann Print Service, The Netherlands.

ISBN: 90-365-2133-5

ITERATIVE LEARNING CONTROL
—A CRITICAL REVIEW—

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof. dr. W. H. M. Zijm,
volgens besluit van het College voor Promoties,
in het openbaar te verdedigen
op donderdag 13 januari 2005 om 15.00 uur

door

Markus Hendrik Adriaan Verwoerd
geboren op 16 November 1977
te Woerden, Nederland

Dit proefschrift is goedgekeurd door:

prof. dr. ir. J. van Amerongen, promotor

dr. ir. T.J.A. de Vries, assistent-promotor

dr. ir. G. Meinsma, assistent-promotor

Voorwoord

> "Ik had zo graag dat het tussen u en mij niet werd tot een transigeren wat er in kan blijven en wat er uit moet, maar dat u juist de grondgedachten, dus veel meer het algemene, dan het speciale, dat wat tussen de regels staat, zoudt aanvoelen en erkennen."
>
> —L. Brouwers, in een brief aan zijn promotor.

"Spijt in verband met onze slechte kunst is even ongewenst als spijt in verband met ons slecht gedrag. De slechtheid moet worden opgespoord, erkend en zo mogelijk in de toekomst worden vermeden. Tobben over de literaire tekortkomingen van twintig jaar geleden [...] is ongetwijfeld ijdel en zinloos."—Aan deze woorden van Aldous Huxley, geschreven bij de heruitgave van zijn beroemde Heerlijke Nieuwe Wereld (*Brave New World*) moest ik de laatste dagen herhaaldelijk denken. De aanleiding laat zich raden.

Nu maak ik mij wat dit proefschrift betreft overigens geen illusies: van een heruitgave zal het waarschijnlijk nooit komen. Toch was daar ook bij mij die angst, dat gevoel van potentiële schaamte. Wat nu als het allemaal 'onzin' blijkt; als blijkt dat ik, gedreven door naïef idealisme en wat al niet meer, de plank volledig misgeslagen heb?

Ik besef nu dat dit verkapte ijdelheid is. We moeten, om nogmaals Huxley aan te halen, "niet de kracht van ons leven gebruiken om de artistieke zonden te herstellen die begaan en nagelaten zijn door de ander die we in onze jeugd waren". Het is aldus met vrijmoedigheid dat ik dit proefschrift, hetwelk niettegenstaande de zorg eraan besteed vele onvolkomenheden bevat, in uw, de lezers, handen leg.

Het past mij een aantal mensen te bedanken. In de eerste plaats mijn begeleiders: prof. Job van Amerongen, voor de ruimte die hij mij liet, ook toen gaandeweg bleek dat mijn onderzoek geen antwoord ging bieden op de vraag waar ik mee van start was gegaan; dr. Theo de Vries, die als een Aäron optrad als niet spreken kon, voor de aanvankelijk dagelijkse en later wekelijkse begeleiding, alsmede voor de fijne gesprekken, de mentale ondersteuning en de soms niet malse maar altijd eerlijke behandeling; dr. Gjerrit

Meinsma, voor de wiskundige begeleiding en de voor zijn persoon zo typische positief-cynische levenshouding en zijn aan zelfverachting grenzende bescheidenheid ("ik ben een sukkel", "ik heb geen geheugen"); prof. Arjan van der Schaft, inspirator, voor zijn algemene betrokkenheid bij het project, de fascilitaire ondersteuning (totdat een brand een einde maakte aan mijn zwerftocht bij TW ben ik in twee jaar tijd vier keer verhuisd, heeft een voormalig SSB-er mijn papers bij het oud papier gezet en heb ik vele weken noodgedwongen in de bibliotheek doorgebracht) en zijn kritische suggesties, waarmee hij mij bij het schrijven voor al te veel wolligheid heeft behoed.

Thanks also to professors Kevin Moore and Yang Quan Chen for their interest in my work, and for their kind hospitality, which made my stay at Utah State a pleasant one.

Een woord van dank aan mijn collega's bij EL, voor de uitstekende sfeer binnen de vakgroep. Als ik om inspiratie verlegen zat of zomaar even een praatje wilde maken, kon ik altijd wel ergens terecht. Dat heeft mij goed gedaan, met name toen het schrijven maar niet lukken wilde. In dit verband wil ik in het bijzonder bedanken mijn beide paranimfen, Philippus Feenstra en Vincent Duindam. In Philippus vond ik een collega met wie ik meer kon delen dan een kamer alleen. Zijn open, eerlijke houding en bescheiden, nuchtere manier van doen bewaarden mij voor hemelbestormend gedrag en deden mij, als dat nodig was, de betrekkelijkheid van ons gezwoeg waarderen. Met Vincent deel ik, naast een wat ongebruikelijke wiskundige oriëntatie en een niet nader te noemen tic, de herinnering aan een drietal vakanties en een tachtigtal tennislessen. Het ontwerp voor de omslag is van zijn hand. Van mijn collega's bij TW wil ik in het bijzonder noemen Viswanath Talasila en Stefan Strubbe. Van alle mensen heeft eerstgenoemde het meest bijgedragen aan mijn ontwikkeling als wetenschapper. Buiten ons onderzoek om spraken we over uiteenlopende onderwerpen als wiskunde, fysica, filosofie en religie, maar toch nog het meest over wetenschap in het algemeen. Stefan wil ik bedanken voor zijn invloed op mijn ontwikkeling als mens. Als scholieren vonden we elkaar in onze passie voor computers. Die liefde hebben we inmiddels ten grave gedragen, maar daarmee gelukkig niet onze vriendschap. We hebben door de jaren heen veel kunnen delen en ik hoop dat we dat ook in de toekomst kunnen blijven doen.

Vele vrienden, huisgenoten en ex-huisgenoten hebben er direct of indirect, middels telefoontjes, SMSjes, emailtjes of anderszins, toe bijgedragen dat ik dit project tot een goed einde brengen kon. Ik ben hen zeer erkentelijk.

I would like to thank my international friends, for granting me, a foreigner, the privilege to play a part in their lifes. Thanks to all members and former

members of BS-Twente, for their spiritual support, prayers, friendship and encouragement in times of need. Special thanks to Mr. P, Arun, Helena, Imelda, Mirjam, Sheela, and Velma.

In dit alles dank ik God, mijn familie niet vergetend. Moeder en broer, bedankt voor alle steun; dit proefschrift zij aan jullie opgedragen.

Samenvatting

Iterative Learning Control (ILC) is een regelmethode, ontworpen om in uiteenlopende regelsituaties repetiviteit uit te buiten. Haar doel is prestatie te verbeteren middels een mechanisme van *trial and error*. De veronderstelling is dat ILC in staat zou zijn dit doel te realiseren. In dit proefschrift wordt onderzocht of dit potentieel aanwezig is en ook daadwerkelijk wordt benut. Daartoe wordt een vergelijk gemaakt tussen ILC en niet-iteratieve, conventionele methoden, in het bijzonder methoden gebaseerd op *feedback*.

De discussie concentreert zich op een klasse van lineaire, eerste-orde recurrentievergelijkingen met constante coëfficienten, geparameteriseerd door een tweetal begrensde lineaire operatoren. Het probleem van ILC wordt opgevat als een optimalisatieprobleem over de ruimte van begrensde lineaire operatorparen. Twee gevallen worden behandeld: causale ILC en niet-causale ILC.

Bij causale ILC worden beide operatoren begrensd verondersteld. Dit blijkt de maximaal haalbare prestatie te schaden. Onder deze aanname transformeert het ILC syntheseprobleem in een *compensator* ontwerpprobleem en blijken de respectievelijke methoden van causale ILC en conventionele *feedback* equivalent. Het voornaamste resultaat, dat teruggaat op het principe van *Equivalent Feedback* stelt dat er met ieder element in de verzameling van causale iteraties een causale terugkoppelafbeelding correspondeert, welke dezelfde prestatie geeft als het ILC algoritme, echter zonder gebruik te maken van iteraties.

Bij niet-causale ILC vervalt de causaliteits-eis: operatoren mogen ook niet-causaal zijn. Deze extra ontwerpvrijheid blijkt met name nuttig in situaties waarin systemen voorkomen met een hoge relatieve graad of niet-minimumfase gedrag. In zulke situaties blijkt niet-causale ILC de prestatie ten opzichte van conventionele methoden aanzienlijk te kunnen verbeteren.

Wat betreft het algemene probleem van iteratief- en lerend regelen: dit proefschrift betoogt dat de klasse van recurrentievergelijkingen met constante coëfficienten niet geschikt is om adaptatie te implementeren and dat, wanneer modelonzekerheid een probleem wordt, een overschakeling op recurrentievergelijkingen met variabele coëfficienten dient te worden overwogen.

Summary

Iterative Learning Control (ILC) is a control method designed to exploit repetitiveness in various control situations. Its purpose is to enhance performance, using a mechanism of trial and error. Supposedly, the method would have considerable potential to effect this purpose. The aim of this thesis is to investigate whether this potential is really there, and if so, whether it is indeed being capitalized on. To that end, a comparison is drawn between ILC and conventional, noniterative methods, such as feedback control.

The discussion focuses on a class of constant-coefficient linear first-order recurrences, parameterized by a pair of bounded linear operators. Accordingly, the problem of ILC is cast as an optimization problem on the space of bounded linear operator pairs. Two cases are considered: Causal ILC and Noncausal ILC.

In Causal ILC both operators are assumed causal. This assumption proves restrictive as it impairs the achievable performance. It turns the ILC synthesis problem into a compensator design problem, and renders the respective methods of Causal ILC and conventional feedback control equivalent. The main result, which pertains to the principle of Equivalent Feedback states that to each element in the set of causal iterations, there corresponds a particular causal feedback map, the Equivalent Controller, which delivers the exact same performance as the ILC scheme, yet without a need to iterate. The converse result is shown to hold true provided certain conditions on the current cycle operator are met.

In Noncausal ILC, causality constraints do not apply, and noncausal operators are allowed. This extra design freedom proves particularly useful in the context of systems with high relative degree and non-minimum phase behavior, where noncausal ILC is shown to significantly enhance performance, as compared to e.g. conventional feedback control.

As regards the general problem of learning control and the use of iteration, this thesis argues that the class of constant-coefficient recurrences is incapable of implementing adaptive behavior and that variable-coefficient recurrences, i.e. trial-dependent update laws ought be considered would model uncertainty become an issue.

Contents

1

Learning Control: an introduction

Overview – *We provide a historical perspective on the evolution of (Intelligent) Control. Starting from the very first self-regulating mechanism, we track the developments leading to the birth of cybernetics, into the era of intelligent machines. Within this context we introduce and motivate the concept of Learning Control. We outline the central problem, scope and aim of the thesis.*

1.1. Introduction

> "The extent to which we regard something as behaving in an intelligent manner is determined as much by our own state of mind and training as by the properties of the object under consideration."
>
> —Alan Turing

Expressed in the most humble of terms, Intelligent Control (IC) stands for a variety of biologically motivated techniques for solving (complex) control problems. Situated within the IC paradigm, *Learning Control* draws its particular inspiration from man's ability to *learn*. More precisely, it is based on the idea that machines can, to some extent, *emulate* this behavior.

Over the years, the question whether machines really can learn has attracted quite some debate. Basically, one can take either of three views: that of the believer, who attests to the possibility; that of the agnost, who holds it not impossible; and that of the skeptic, who endeavors to prove the opposite.

This issue, however important, falls outside the scope of this thesis. For what it is worth, we do believe that at least some aspects of learning can

be mimicked by engineering artifacts. But the important question, as far as this thesis is concerned, is not whether machines can *learn*, but whether they can exploit repetitiveness in a useful way. In this respect, our perception of learning is relevant only in as far as it helps us in deciding what is useful and what is not.

There is no doubt the use of iteration holds considerable potential, if only as a means to improve control performance. It is the aim of this thesis to find out how, in the context of Learning Control, iteration *is* being deployed as well as how it *could* be deployed; in other words, to determine whether the aforementioned potential is being capitalized on. To that end, we draw a comparison between Iterative Learning Control (ILC), and its non-iterative counterpart, Conventional Feedback Control (CFC).

1.2. From Ktesibios to Turing: on the evolution of the autonomous machine

It needs no argument that Learning Control and Conventional Feedback Control differ in many respects. Yet, one thing links them together, and that is their role in the development of autonomous mechanisms. Before we zoom in on their distinctive features, let us presently consider their collective past.

1.2.1. Feedback Control: the first generation of autonomous machines

> "The 'trick' of using negative feedback to increase the precision of an inherently imprecise device was subtle and brilliant."
>
> —Dennis S. Bernstein

Ktesibios and the water clock

Ever since antiquity, and possibly long before, man has been intrigued by the idea of autonomous machines. A good example, and one of the first at that, is the self-regulating water clock—Uber (2004). The water clock (*Clepsydra*) is an ancient Egyptian invention dating back to around 1500 B.C. It consists of two vessels. The first vessel, having a small aperture at the bottom, drops water into a second, lower vessel, at a supposedly constant rate. The indication of time is effected by marks that correspond

to either the diminuition of water in the supplying vessel, or the increase of water in the receiving vessel.

First-generation water clocks were not very effective—not as a means for timekeeping, that is. For as it turns out, the water flow is much more rapid when the supplying vessel is full, than when it is nearly empty (owing to the difference in water pressure).

It was Ktesibios from Alexandria, who perfected the design by adding a crucial ingredient. Ktesibios, a barber by profession who eventually became an engineer under Ptolomy II, lived in the third century B.C. A contemporary of Euclid and Archimedes, he is credited with the invention of the pump, the water organ and several kinds of catapults. Confronted with the water clock problem, Ktesibios developed a self-regulating valve. The valve is comprised of a "cone-shaped float" and a "mating inverted funnel"—Kelly (1994). In a normal situation, water flows through the valve, down from the funnel, over the cone, into a bowl. As water comes in and fills the bowl, the cone floats up into the funnel, blocking the passage. As the water diminishes, the float sinks, allowing more water to enter.

The self-regulating valve served its purpose well. And with that, the water clock became a milestone in the history of Control Theory, even "the first nonliving object to self-regulate, self-govern, and self-control", "the first self to be born outside of biology"—Kelly (1994).

Cornelis J. Drebbel and the thermostat

Cornelis Drebbel was a Dutch alchemist and self-made engineer. He is known for his work on submarines (first prototype in 1620), optics, and dyeing, not to mention his perpetual mobile. His more modest (but no less useful) inventions include the thermostat, which, it is said, he thought out while trying to forge gold from lead, identifying temperature fluctuations as the main cause for failure. To counteract these fluctuations, he came up with the idea to manipulate the combustion process by regulating the air (oxygen) supply; onto one side of the furnace, he mounted "a glass tube [. . .] filled with alcohol. When heated, the liquid would expand, pushing mercury into a connecting, second tube, which in turn would push a rod that would close an air draft on the stove. The hotter the furnace, the further the draft would close, decreasing the fire. The cooling tube retracted the rod, thus opening the draft and increasing the fire"—Kelly (1994).

Like Ktesibios' valve, Drebbel's thermostat is completely autonomous. Strikingly simple and yet very effective, it provides an excellent example of what feedback control can do.

James Watt and the self-regulating governor

James Watt (1736-1819), scottish inventor and mechanical engineer, is often credited with the invention of the steam engine. In actual fact, the steam engine was invented by Thomas Savery and Thomas Newcomen. But like Ktesibios before him, Watt greatly improved on an existing design. Most notably, he came up with the idea of using a separate condensing chamber, which greatly enhanced both the engine's power and its efficiency—Bernstein (2002). But that was not his only contribution: the rotary motion of existing steam engines suffered from variations in speed, which limited their applicability. Adapting a design by Mead, Watt developed a mechanism we now know as the *Watt governor*. The Watt governer is basically a double conical pendulum which, through various levers and linkages, connects to a throttle valve. The shaft of the governor is spun by the steam engine. As it spins, centrifugal force pushes the weights outward, moving linkages that slow the machine down. As the shaft slows down, the weights fall, engaging the throttle that speeds the engine up. Thus the governor forces the engine to operate at a constant and consistent speed.

Feedback as a universal principle

The central notion embodied in the work of Ktesibios, Drebbel and Watt is that of *feedback*. Roughly speaking, the purpose of feedback, as far as regulators are concerned, is to *counteract change*, whether this be change in rate of flow, change in temperature, or change in rotational velocity.

An invisible thread through history, feedback has had a tremendous impact on the advancement of technology, a fact to which also the 20th century testifies. In the late 1920s Harold Black, an engineer at Bell Labs, introduced (negative) feedback to suppress the effect of gain variations and nonlinear distortions in amplifiers—Bennett (1979); Bernstein (2002). A few years later, when World War II necessitated a rapid technological advance, feedback was at the heart of it.

By that time, feedback had become a universal concept and was recognized as one of the most powerful ideas in the general science of systems. Researchers in psychology, (neuro)physiology, mathematics, and computer science combined forces to conclude that there must be some common mechanism of communication and control in man and machine. It was time for cybernetics.

1.2.2. AI and the second generation of autonomous machines

> "Let us then boldly conclude that man is a machine, and that in the whole universe there is but a single substance differently modified."
>
> —Julien Offray de La Mettrie

Cybernetics

The period after the war saw the birth of cybernetics. One of the pioneers in this area was Norbert Wiener, who, apart from his contributions to mathematics proper, is best known for his book entitled 'Cybernetics, or Control and Communication in the Animal and the Machine'. On the website of *Principia Cybernetica* (2004), it is related how Wiener developed some of his ideas. While working on a design of an automatic range finder (a servomechanism for predicting the trajectory of an airplane on the basis of past measurements) Wiener was struck by two facts: (a) the seemingly intelligent behavior of such machines, i.e. their ability to deal with experience and to anticipate the future; (b) the 'diseases' (certain defects in performance) that could affect them. Conferring with his friend Rosenbluth, who was a neurophysiologist, he learned that similar behavior was to be found in man. From this he concluded that in order to allow for purposeful behavior, the path of information inside the human body must form "a closed loop allowing the evaluation of the effects of one's actions and the adaptation of future conduct based on past performances".

In other words, Wiener saw a clear parallel between the self-regulating action in machines and that in man. Purposeful behavior was to be explained in terms of (negative) feedback. The same idea took shape in the work of Warren McCullogh a psychologist perhaps best known for his work on neural networks. In a paper entitled 'machines that want and think', McCullogh uses the *governor* (see Section 1.2.1) as a metaphor for self-regulation in the human body. He writes: "Purposive acts cease when they reach their ends. Only negative feedback so behaves and only it can set the link of a governor to any purpose."

This unifying view on man and machine is not unique to Cybernetics. Throughout history, mankind has attempted to explain itself in terms of metaphors derived from once current technology. Examples include clocks, steam engines, and switchboards—Maessen (1990). But it was not until the advent of analog and digital computing (the age of cybernetics) that these

metaphors became sophisticated enough to threaten man's superiority and his exclusive rights to intelligence.

Intelligent machinery—Alan Turing

Long before Darwin stepped up to challenge man's position among other living beings, Julien de La Mettrie boldly concluded that "man is a machine, and that in the whole universe there is but a single substance differently modified". During the 19th century, the prevailing thought that man held special position in the whole of creation, slowly began to lose ground. In a 1948 article entitled 'intelligent machinery' (Turing (1968)), Alan Turing takes up the question whether it is possible for machinery to show intelligent behavior. In those days, the common opinion was that it is not ("It is usually assumed without argument that it is not possible"). Turing explains that this negative attitude is due to a number of reasons, varying from the unwillingness of man to give up their intellectual superiority, to Gödel's Theorem, to religious belief, and from the limitations of the then current machinery to the viewpoint that every intelligence is to be regarded as a reflection of that of its creator. He goes on to refute each one of them. Most notably, he likens the view that intelligence in machinery is merely a reflection of that of its creator to the view that the credits for a discovery of a pupil should be given to his teacher. "In such a case the teacher would be pleased with the success of his methods of education, but would not claim the results themselves, unless he had actually communicated them to his pupil."

Constrasting the common opinion, Turing argues that there is good reason to believe in the possibility of making thinking machinery. Among other things, he points to the fact that it is possible to make machinery to imitate "any small part of man". He mentions examples of machine vision (camera), machine hearing (microphone), and machine motion (remotely controlled robots that perform certain balancing tasks). Rejecting the most sure way of building a thinking machine (taking a man as a whole and replacing all parts of him by machinery) on grounds of impractability, Turing proposes a research program, slightly less ambitious, and arguably more feasible, namely to build a "brain, which is more or less without a body, providing at most organs of sight, speech, and hearing".

This very program formed the basis for a new branch of science, which today is called by the name Artificial Intelligence (AI). Turing himself did not live to witness the full impact of his ideas. He died in 1954, leaving his work as a legacy for future generations of researchers.

1.3. Iterative Learning Control: a first acquaintance

As we picture a team of robots lined up alongside some conveyor belt in a manufacturing plant, performing the exact same operation over and over again, it occurs to us that if we could find a way to enhance each robot's performance, the plant could operate at higher speed, which would effect a higher throughput. One way or another, we ought to be able to exploit repetitiveness.

1.3.1. Iterative Learning Control

Learning Control comes in many flavors; flavors, which differ mainly in the way they represent knowledge (e.g. using neural networks or other data structures) as well as in how they update the stored information. In this thesis we focus on the particular methodology that goes by the name of *Iterative Learning Control* (ILC). For a detailed overview of the ILC literature, we refer to the next chapter. Here, a brief sketch will suffice.

Where Learning Control would merely assume successive control tasks to be *related*, ILC assumes them to be *identical*. In addition, ILC assumes a perfect reset after each trial (iteration, execution, run), so as to ensure that the system can repeatedly operate on the same task under the same conditions. This latter assumption can be relaxed, for several studies (see for instance Arimoto et al. (1986); Lee and Bien (1996)) have shown that, although resetting errors generally deteriorate performance, they do not rule out the possibility of 'learning' altogether.

ILC is concerned with a single control task. Consequently, all information is typically encoded as a function of a single parameter: *time*. The desired behavior (output) is defined accordingly. Extensions based on more elaborate data structures have been considered. Most notably, *Learning Feed-Forward Control* (LFFC, Velthuis (2000); de Kruif (2004)) encodes all information as a function of a virtual state vector (which may have position, velocity, and acceleration among its components). The obvious advantage of LFFC (over ILC) is that it renders the condition on strict trial-to-trial repetition obsolete. A disadvantage is that the method requires less transparent data structures, and significantly more complex techniques for storage and retrieval. Another, related, approach is that of *Direct Learning Control* (DLC)—Xu (1996); Xu and Zhu (1999). Given two output trajectories, identical up to an arbitrary scaling in time and/or magnitude, the problem of DLC is to construct the input corresponding to the one using the input

corresponding to the other, without resorting to repetition (hence, *direct*).

Iterative Learning Control has been applied in various industrial settings, usually as an *add-on* to existing control algorithms. Examples include, among others, wafer stages—de Roover and Bosgra (2000), welding processes—Schrijver (2002); Naidu et al. (2003), and robot manipulators—Kavli (1992). For more information on ILC applications, we refer to Bien and Xu (1998) and references therein.

1.3.2. The problem of ILC

> "The real problem is not whether machines think, but whether man do"
>
> —B.F. Skinner

We learned that ILC is about enhancing a system's performance by means of *repetition*, but we did not learn *how* it is done. This bring us to the core activity in ILC research, which is the construction and subsequent analysis of *algorithms*.

ILC starts from a qualitative description of a learning behavior . The problem is to find an algorithm which implements it. As this is not an easy problem, let us outline what features such an algorithm should have. Indeed, let us take a step back, and ask: "*What is the need for (Iterative) Learning Control?*" Admitted, certain applications simply seem to ask for Learning Control, particularly those with a pronouncedly cyclic operational component. But as a motivation, this does not suffice, since it does not explicate a need. Part of this need becomes apparent when we concentrate on situations in which conventional control (e.g. feedback control) is not likely to yield adequate performance, for instance because the *a priori* knowledge does not allow for a competitive design. Whatever ILC is believed to do, this much at least is true: the use of iterations opens a *possibility* to improve performance. Needless to say, the real difficulty is to convert this possibility into an actuality.

With the early work of Arimoto as a possible exception, the vast body of literature contains few contributions which put serious effort in delineating ILC's distinctive features against the spectrum of other control methods. Also, there is no single, widely accepted, formal problem definition, nor any format we know of that stands out from the rest. In contrast, there is a clear *intuitive agreement* on what ILC (as opposed to other, conventional methods) should effect:

> "The Learning Control concept stands for the repeatability of operating a given objective system and the possibility of improving the control input on the basis of previous actual operation data."
>
> —Arimoto et al. (1986)

> "Learning Control is a name attributed to a class of self-tuning processes whereby the systems performance of a specified task improves, based on the previous performance of identical tasks."
>
> —Heinzinger et al. (1992)

> "Learning Control is a technique in which the input signal required to achieve a given behavior as output of a dynamical system is built iteratively from successive experiments."
>
> —Luca et al. (1992)

> "Learning Control is an iterative approach to the problem of improving transient behavior for processes that are repetitive in nature."
>
> —Moore (1993)

> "The main strategy of the Iterative Learning Control is to improve the quality of control iteratively by using information obtained from previous trials, and finally to obtain the control input that causes the desired output."
>
> —Jang et al. (1995)

> "The goal of Iterative Learning Control is to improve the accuracy of a system that repeatedly follows a reference trajectory."
>
> —Goldsmith (2002)

Note that in their original context, these fragments did not necessarily serve as (formal) problem *definitions*. Also, they may not be representative of the authors' current view. Yet they do reflect how various people, in past and present, have come to *view* the problem of ILC. The common theme is that ILC is an *iterative* method as opposed to, for instance, conventional feedback control, or feedforward control (FFC) (to which we distinctively refer as *noniterative*, or *a priori* methods). Iteration is thought of as a means for contructing the control input corresponding to some fixed desired output. In view of this, various authors have referred to ILC as an iterative inversion process. This view however does not account, at least not explicitly, for the role of *uncertainty*,; others have suggested (and we tend to agree with them) that uncertainty is where learning comes, or *should come*, into play.

> "One way to describe the learning is as a process where the objective of achieving a desired result is obtained by experience *when only partial knowledge about the plant is available.*" (emphasis added, MV)
>
> —Chen and Wen (1999)

In another place, the same authors as above cited proclaim: "Learning is a bridge between knowledge and experience". The way we see it, the problem of ILC is to make such notions precise.

1.4. Problem Statement: Iterative Learning vs. Conventional Feedback Control

According to Passino (1993), much of the controversy about Intelligent Control stems from the name itself—no two people would agree on any definition of intelligence and "even if at one point in time a group of experts would agree that a system exhibits intelligence, over the years, as the system is better understood, the experts often begin to classify the exhibited behavior as 'algorithmic' and 'unintelligent' "—as well as from the hype that it generates (the conception that since it is 'intelligent' it must automatically be better than other, conventional approaches). Most conventional control engineers, Passino argues, are not at all concerned with the question whether their controller is intelligent; They simply seek to develop a controller that will enhance their system's performance. "...They prefer to leave the 'intelligence' issue to persons such as psychologists, philosophers, persons in medical professions, and the computer scientists in AI that try to emulate it."

Passino argues that the general negative attitude towards methods of Intelligent Control is due to an unfortunate emphasis on attempts to define *intelligence*; much of the controversy could be overcome by focusing on the *methodology* instead. In effect, he proposes to first define the methodology and to base the definition of an intelligent controller on that. His definition of an intelligent control methodology would read: "A control methodology is an intelligent control methodology if it uses human/animal/biologically motivated techniques and procedures (e.g., forms of representation and/or decision making) to develop and/or implement a controller for a dynamical system". Based on this definition "a controller is an intelligent controller if it is developed/implemented with (a) an intelligent control methodology, or (b) conventional systems/control techniques to emulate/perform control

functions that are normally performed by humans/animals/biological systems".

1.4.1. Problem Statement

> "The greatest challenge to any thinker is stating the problem in a way that will allow a solution."
>
> —Bertrand Russell

Paraphrasing Passino, we could say that much of the negative attitude towards Learning Control is due to an unfortunate emphasis on attempts to define 'learning'. Rather than asking: what *is* Learning Control, why not take a pragmatic approach and ask: "What can we do with it?".

We see no reason to differ; that is, not as long as there is some room for critical questions to be asked. One such question concerns the following. We have seen that Learning Control ought to capitalize on repetitiveness; that much is clear. But what does that really mean? To us, it means that Learning Control ought to outperform any conventional, i.e. noniterative method of control. In addition we expect it to have an enhanced ability to deal with uncertainty (alternatively: an ability to adapt). Both aspects are covered by different connotations of the word 'learning' and it is only in that sense that we care to ask whether our system implements a learning behavior.

Our intent with this thesis is not to question, but to underline ILC's distinctiveness. And for that reason we thought it good to compare it against as well established a method as Conventional Feedback Control. In doing so we try to keep the discussion as general as possible. Yet, for the most part, our analysis does not extend beyond the realm of linear, time-invariant systems, does not incorporate sampling effects, is confined to systems defined over infinite time, assumes all operators to be bounded, and restricts attention to the specific class of first-order recurrences (and though we do believe this class to be generic in a sense to be made precise, it certainly does not encompass each and every algorithm).

Having said that, let us expound on what this thesis is about. Recall that the purpose of the thesis is to bring to light the fundamental dinstinction between Iterarive Learning Control and Conventional Feedback Control. Our investigation hinges on two central issues. These issues are:

1. How does ILC overcome the performance limitations in Conventional Feedback Control?

2. How does it handle uncertainty in the plant model?

Dealing with the first issue, we ask such questions as: What are the main determinants for performance, what are the limitations? How does the information feedback in ILC differ from that in Conventional Feedback Control?

Dealing with the second, we focus on design uncertainty. The issues we address are: How does ILC counteract a lack of a priori knowledge, if at all? How do the convergence conditions in ILC relate to the stability conditions in Feedback Control? Does ILC, like Conventional Feedback Control, exhibit a tradeoff between robust stability and performance? What is the use of iteration with regard to model uncertainty, what with respect to noise? How many iterations are necessary, and how does this number relate to the structural complexity of the plant?

In answering these questions we will learn that, notwithstanding the apparent differences in evolution and appearance, ILC and Conventional Feedback Control have more in common then expected, but are by no means interchangeable.

1.5. Outline

The outline of the thesis is as follows. Chapter 2 provides an overview of Iterative Learning Control. This includes a brief historical outline, as well as a review of commonly deployed design methods. Opening with a recap of relevant background material, the chapter ends with a formal problem statement.

Chapter 3 deals with a class of causal update equations and its relation with conventional feedback. It is shown that the basic synthesis problem (that of finding the minimizing argument of some cost functional defined over the space of all causal bounded linear operator pairs) is essentially a *compensator design problem*.

Basically an extension of Chapter 3, Chapter 4 considers the same family of update laws, with the exception that the operators involved are no longer constrained to be causal. The idea of using noncausal update laws instead of causal update laws is shown to have a solid foundation in the solution to certain optimization problems. It is shown that noncausal update laws do not subject to the same limitations as causal update laws.

Integrating results from previous chapters, Chapter 5 provides a detailed discussion on the subject of Learning Control proper. The word 'learning' carrying a variety of meanings, the question is posed in what sense, and to what extent the algorithms discussed in this thesis implement a 'learning

behavior'. Higher-level update laws are introduced as mathematical objects potentially capable of modelling rudimentary forms of learning (in a sense to be made precise). Particular instances of such update laws are shown to have interesting 'adaptive' abilities.

The final chapter, Chapter 6, contains a discussion of the main results in relation to the specific assumptions on the basis of which they were derived. In addition, it presents the thesis' main conclusions regarding the distinctive nature of ILC vis-à-vis conventional methods of control.

2

Iterative Learning Control: an overview

Overview – *Starting off with a review of preliminary results, this chapter provides an introduction to the field of Iterative Learning Control. It contains: (a) a brief historical sketch (covering both birth and evolution, but with a focus on the former); (b) a digest of frequently encountered algorithms, and (c) an overview of well-established design procedures. Awaiting detailed treatment (due in the next chapter), the final section introduces the class of algorithms to be looked at, and poses the problem of ILC as a synthesis problem on the space of causal bounded linear operator pairs.*

2.1. Basic framework: preliminaries, notation and terminology

We review some preliminaries, and introduce a notation.

2.1.1. Signals and systems: $\mathcal{H}_2$ and $\mathcal{H}_\infty$

Viewed as an abstract mathematical object, a *signal* is a vector function $u : \mathbb{C} \mapsto \mathbb{C}^{\mathfrak{u}}$. The set of signals thus defined forms a vector space under pointwise addition and complex scalar multiplication (over the complex field $\mathbb{C}$). Of all signals, those that are in some sense bounded are of particular interest. This brings us to the concept of a *normed vector space*. A normed vector space is a vector space whose elements are bounded with respect to

some associated *norm*. In the present text we will be concerned with $\mathcal{L}_2$-spaces only, and with $\mathcal{L}_2(\mathbb{R}_+)$ in particular. The latter space is the space of all vector functions $u : \mathbb{R}_+ \mapsto \mathbb{R}^{\mathtt{u}}$, such that

$$\left\|u\right\|^2_{\mathcal{L}_2(\mathbb{R}_+)} \;:=\; \int_0^\infty \left\|u(t)\right\|_2^2 \mathrm{d}t, \tag{2.1}$$

is finite. Here, $\left\|.\right\|_p$ denotes the euclidean p-norm, defined as $\left\|u(t)\right\|_p^p := \sum_{i=1}^{\mathtt{u}} |u_i(t)|^p$. More generally, for any interval $I \subset \mathbb{R}_+$, and for $1 \leq p < \infty$, we say that u is in $\mathcal{L}_p(I)$ if

$$\int_I \left\|u(t)\right\|_p^p \mathrm{d}t \;<\; \infty. \tag{2.2}$$

In addition, we define $\mathcal{L}_\infty(I) := \{u : I \subset \mathbb{R}_+ \mapsto \mathbb{R}^{\mathtt{u}} \mid \operatorname{ess\,sup}_{t\in I} \max_i \left|u_i(t)\right| < \infty\}$. The time-domain space $\mathcal{L}_2(\mathbb{R}_+)$ is isometric (with respect to the norm defined by (2.1)) to the frequency-domain space $\mathcal{H}_2$. The latter space, $\mathcal{H}_2$, is the space of all matrix functions $F(s)$, analytic in $\mathrm{Re}(s) > 0$, such that

$$\left\|F\right\|^2_{\mathcal{H}_2} \;:=\; \sup_{\sigma>0} \frac{1}{2\pi} \int_{-\infty}^{\infty} \mathrm{Trace}\left[F^*(\sigma + j\omega)F(\sigma + j\omega)\right] \mathrm{d}\omega \tag{2.3}$$

is finite. Here, $(\cdot)^*$ denotes the complex conjugate transpose. It can be shown, see for instance Zhou et al. (1996); Francis (1987), that

$$\left\|F\right\|^2_{\mathcal{H}_2} \;=\; \frac{1}{2\pi} \int_{-\infty}^{\infty} \mathrm{Trace}\left[F^*(j\omega)F(j\omega)\right] \mathrm{d}\omega. \tag{2.4}$$

The isomorphism between $\mathcal{L}_2(\mathbb{R}_+)$ and $\mathcal{H}_2$ is given by the Laplace transform. The Laplace Transform allows us to identify every $u(t) \in \mathcal{L}_2(\mathbb{R}_+)$ with an element $u(s) \in \mathcal{H}_2$ and vice versa.

Linear Systems

Having defined the notion of a signal and the associated concept of a (normed) signal space, we are now in position to define what we mean by a *system*. The following material is largely based on Kwakernaak and Sivan (1991). A system (more precisely, an input-output, or *IO system*) is defined by a signal set U, called the *input set*, a signal set Y, called the *output set*, and a subset R on the product set $U \times Y$, called the *rule*, or *relation* of the system. In this thesis we will be solely concerned with so called *input-output mapping* (IOM) systems. An element of the larger class of IO systems, an IOM system is characterized by a (single-valued) input-output *map* P that assigns a *unique* output $y \in Y$ to each input $u \in U$. An

IOM system can have many additional properties. For instance, it may be linear or nonlinear, time-varying or time-invariant, anticipating (noncausal) or non-anticipating (causal), etc.

Let U, Y be vector spaces and let $P : U \mapsto Y$ define an IOM system with input set U and output set Y. A pair $(u, y) \in U \times Y$ with $y = Pu$ is said to be an *input-output pair* (IO pair) of the system P. Let (u_1, y_1) and (u_2, y_2) be any two IO pairs of the system P. Then P is *linear* if, for any $\alpha, \beta \in \mathbb{R}$, $(\alpha u_1 + \beta u_2, \alpha y_1 + \beta y_2)$ is also an IO pair. A system that is not linear is said to be *nonlinear*. Of all nonlinear systems, the *affine* systems form a particular subset. An IOM system $P : U \mapsto Y$ is said to be affine if the associated operator $P_\emptyset : U \mapsto Y,\ P_\emptyset(u) := Pu - P\emptyset$ is linear (here, $\emptyset$ denotes the zero element of U). The system P is said to be *time-invariant* if for every IO pair (u, y) also the time-shifted pair $(u(t-\tau), y(t-\tau))$ is an IO-pair for any admissible time shift τ (with $\mathbb{T}$ denoting the *time axis*, τ is admissible if $\mathbb{T} + \tau = \mathbb{T}$). A system that is not time-invariant is said to be *time-varying*.

Now let U, Y be *normed* vector spaces and suppose $P : U \mapsto Y$ is linear. Then P is said to be *bounded* (with respect to vector norms $\left\|\cdot\right\|_U$ and $\left\|\cdot\right\|_Y$) if there exists a real number c such that $\left\|Pu\right\|_Y \leq c \left\|u\right\|_U$ for all $u \in U$. Let P be any linear system, we define its *induced norm* as $\left\|P\right\| := \sup\left\{\left\|Pu\right\|_Y :\ u \in U, \left\|u\right\|_U = 1\right\}$. Provided U and Y are (normed) vector spaces, the set of all bounded linear IOM systems with input set U and output set Y, is again a vector space. This space is generally denoted as $\mathcal{L}(U, Y)$, with the convention that $\mathcal{L}(U) := \mathcal{L}(U, U)$. Its associated norm is that induced by U and Y.

Finite-Dimensional LTI systems

Most of our analysis pertains to *finite-dimensional*, linear, time-invariant (LTI) IOM systems. Recall that an LTI system is finite dimensional if it has a realization on a finite-dimensional state space. It is well known (see for instance Kwakernaak and Sivan (1991)) that essentially every LTI IOM system can be cast into the form

$$\begin{aligned}(Pu)(t) &= (h_P * u)(t) \\ &:= \int_{-\infty}^{\infty} h_P(t-\tau)u(\tau)\mathrm{d}\tau. \end{aligned} \tag{2.5}$$

Systems as defined in (2.5) are known as *convolution systems*. The function $h_P : \mathbb{R} \mapsto \mathbb{R}$ is called the kernel or *impulse response* of P. For causal or non-anticipating systems, the impulse response is identically zero for

negative time. An LTI operator is *BIBO-stable* (or simply *bounded*) if every bounded input yields a bounded output. A sufficient condition for the convolution system (2.5) to be BIBO-stable in the sense of $\mathcal{L}_2(\mathbb{R}) \mapsto \mathcal{L}_2(\mathbb{R})$ is $h_P \in \mathcal{L}_1(\mathbb{R})$.

Recall that under the (one-sided) Laplace Transform the space of $\mathcal{L}_2$-stable causal finite-dimensional LTI operators transforms into the space of *real-rational* proper transfer functions with no poles in the closed right half plane, i.e. into the frequency domain space $\mathcal{RH}_\infty$. As a consequence, the $\mathcal{L}_2(\mathbb{R}_+)$-induced norm for stable causal finite dimensional LTI operators coincides with the $\mathcal{H}_2$-induced norm for operators in $\mathcal{RH}_\infty$,

$$\left\|P\right\|_\infty \;:=\; \sup_{\omega\in\mathbb{R}} \bar{\sigma}\left(P(j\omega)\right). \tag{2.6}$$

Here $\bar{\sigma}(\cdot)$ denotes the largest singular value.

2.1.2. Convergence

Iterative schemes generate sequences of inputs, and *convergence analysis* is to tell whether, and under what conditions these sequences converge. The mathematical literature contains a number of different notions of convergence. As far as *sequences of functions* are concerned, we may distinguish between (a) *pointwise convergence*: for each $t \in [0, T]$ and every $\varepsilon > 0$ there exists $K \in \mathbb{N}$ (possibly depending on t), such that $|u_k(t) - \bar{u}(t)| < \varepsilon$ for all $k > K$; (b) *uniform convergence*: for each $\varepsilon \in \mathbb{R}$ there exists $K \in \mathbb{N}$ (*not* depending on t) such that $|u_k(t) - \bar{u}(t)| < \varepsilon$ for all $k > K$ and all $t \in [0, T]$. In this thesis, we adopt the following definition.

Definition 2.1.1 (Convergence). *Let U be a normed vector space. A sequence $\{u_k\}$ is said to converge to a limit $\bar{u} \in U$ if for every $\varepsilon > 0$ there exists K such that $\left\|u_k - \bar{u}\right\|_U < \varepsilon$ for all $k > K$. Sequences that do not converge are said to diverge.*

Convergence of input, and output sequences

In the context of Iterative Learning Control, it is convenient to distinguish between input sequences and output sequences. Let $P : U \mapsto Y$ be an IOM system. An *input sequence* is any sequence $\{u_0, u_1, \ldots\}$ such that $u_i \in U$ for all i. Correspondingly, a sequence $\{y_0, y_1, \ldots\}$ is an *output sequence* if there exists an input sequence $\{u_i\}$ such that $y_i := Pu_i$ for all i. Input, and output sequences are related (by the IO map P) and so are, to some extent, their convergence properties. Indeed, the following lemma shows that if P

is a bounded linear (IOM) system then convergence of the input sequence *implies* that of the output sequence.

Lemma 2.1.2. *Let $P : U \mapsto Y$ be a bounded linear IOM system and let $\{u_0, u_1, \ldots\}$ be an input sequence. If the input sequence converges to a limit $\bar{u} \in U$, then likewise, the output sequence $\{Pu_0, Pu_1, \ldots\}$ converges to a limit $\bar{y} \in Y$. Moreover we have that $\bar{y} = P\bar{u}$.*

Proof. To prove: $Pu_k \to P\bar{u}$ as $k \to \infty$. By linearity, $\|Pu_k - P\bar{u}\|_Y = \|P(u_k - \bar{u})\|_Y$. And by definition of the induced norm,

$$\|P(u_k - \bar{u})\|_Y \quad \leq \quad \|P\| \, \|u_k - \bar{u}\|_U . \tag{2.7}$$

Thus the right hand side of (2.7) converges to zero and ergo the left hand side. This completes the proof. ∎

The converse of Lemma 2.1.2 is generally not true, as the following example illustrates.

Example 2.1.3 (Diverging input). *Let $P \in \mathcal{RH}_\infty$ be such that $|1 - P(j\omega)| < 1$ for all finite frequencies. Consider the update law*

$$u_{k+1} \quad = \quad u_k + (y_d - Pu_k), \qquad k = 0, 1, \ldots \tag{2.8}$$

Assume $u_0, y_d \in \mathcal{H}_2$. We show that the sequence $\{Pu_0, Pu_1, \ldots\}$ converges to a limit $\bar{y} = y_d$ and does so for all y_d and all u_0. Define $e_k := y_d - Pu_k$ and rewrite (2.8) to obtain

$$e_{k+1} \quad = \quad (1 - P)\, e_k. \tag{2.9}$$

To prove: for every $\varepsilon > 0$ there exist $K \in \mathbb{N}$ such that $\|e_k\| < \varepsilon$ for all $k > K$. We proceed as follows. Take any $\varepsilon > 0$ and pick $a > 0$ such that $\int_a^\infty |e_0|^2 d\omega < \varepsilon/4$ (one may verify that such a number always exists by virtue of the fact that e_0 is in $\mathcal{H}_2$). Define $\gamma_a := \max_{\omega \in [0,a]} |1 - P(j\omega)|$. Note that $\gamma_a < 1$. We select K such that $\gamma_a^{2K} \int_0^a |e_0(j\omega)|^2 < \varepsilon/4$. Suppose $k > K$. It follows that

$$\begin{aligned} \|e_k\|^2/2 \quad &= \quad \int_0^a |1 - P(j\omega)|^{2k} \, |e_0(j\omega)|^2 d\omega + \int_a^\infty |1 - P(j\omega)| \, |e_0(j\omega)|^2 d\omega \\ &\leq \quad \max_{\omega \in [0,a]} |1 - P(j\omega)|^{2k} \int_0^a |e_0(j\omega)|^2 d\omega + \int_a^\infty |e_0(j\omega)|^2 d\omega \\ &\leq \quad \gamma_a^{2K} \int_0^a |e_0(j\omega)|^2 d\omega + \varepsilon/2 \quad < \quad \varepsilon/4 + \varepsilon/4 = \varepsilon/2. \end{aligned} \tag{2.10}$$

Which shows that $\|e_k\| < \varepsilon$. This concludes the proof.

Next consider the input sequence $\{u_0, u_1, \ldots\}$. *Suppose this sequence converges to some limit* $\bar{u}(y_d) \in \mathcal{H}_2$, *and does so for every* $y_d \in \mathcal{H}_2$. *Then it follows from (2.8) that* $(I - P)^{-1} \in \mathcal{RH}_\infty$. *Clearly, with* P *as in*

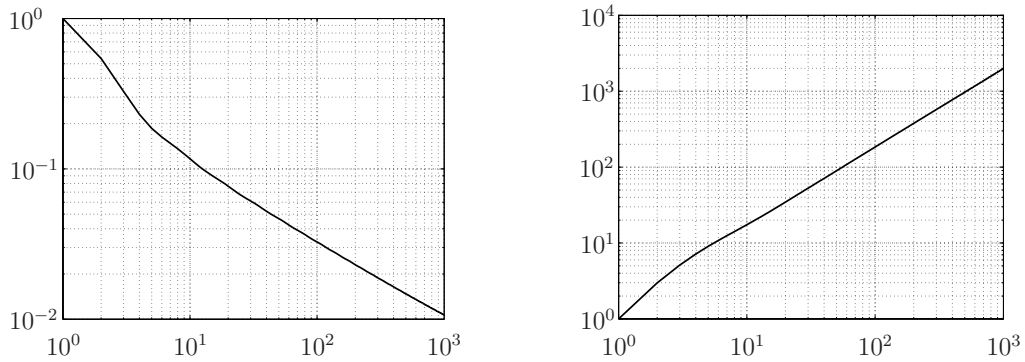
$$P(s) \;:=\; \frac{1}{s+1}, \tag{2.11}$$

this condition is not satisfied, which implies that the input sequence diverges for at least one $y_d \in \mathcal{H}_2$. *We conclude that convergence of the output sequence* does not *imply convergence of the input sequence.*

We close with a numerical example. Let $P(s)$ *be as above and define* y_d *to be*

$$y_d(t) \;:=\; \begin{cases} 0 & t \in [0, 2) \\ 1 & t \in [2, 4] \end{cases} \tag{2.12}$$

We apply update law (2.8) with initial input $u_0 = 0$. *Figure 2.1 shows the results. We observe that the mean squared error* $\|e_k\|$ *slowly tends to zero, whereas the mean-squared input* $\|u_k\|$ *grows without bound.*



(a) $\|e_k\|/\|e_0\|$ (b) $\|u_k\|/\|u_0\|$

Figure 2.1.: Simulation results corresponding to Example 2.1.3. The figure on the left and the right respectively show the normalized mean-squared error $\|e_k\|/\|e_0\|$, and the normalized mean squared input $\|u_k\|/\|u_0\|$. On the horizontal axis is the trial number k.

Example 2.1.3 may appear rather contrived. For it does not require a great deal of insight to see that the desired output (2.12) is not *feasible* for the given system to track, in the sense that no bounded input exists that will produce the output asked for. In general (e.g. when dealing with MIMO

systems) this may not be so easy to see. In this respect, information about the system's relative degree is essential. Without such information, asking for perfect tracking may well be asking too much.

Convergence over finite time

In the literature, the problem of ILC is often cast as a tracking problem over *finite time*. This is done for a good reason, since most signals of interest (in particular the reference signal itself) are typically not defined outside some finite window $[0, T]$. In many analyses, signals of finite duration are extended over the entire semi-infinite axis $[0, \infty)$, to allow, for instance, for certain forms of transform analysis.

In view of this practice, it is important to note that conclusions derived from an infinite time analysis, need not (equally) apply in a finite-time context. To give but one example: the *system gain* (the $\mathcal{L}_2[0, T]$-induced norm) is known to be a nondecreasing function of T. If we were to estimate this particular system property using infinite-time data, the estimate would be conservative.

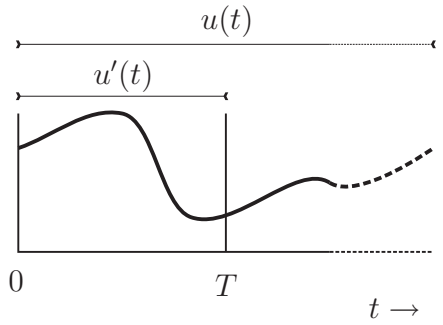


Figure 2.2.: With any $u \in \mathcal{L}_2[0, \infty)$ we associate $u' \in \mathcal{L}_2[0, T]$.

The next observation is particularly relevant in relation to ILC. Let $\{u_k(t)\}$ be a sequence of functions defined on the entire right semi-infinite interval, $[0, \infty)$. If $u_k \in \mathcal{L}_2[0, \infty)$ for all $k \geq 0$ and, in addition, $\lim_{k\to\infty} u_k =: \bar{u} \in \mathcal{L}_2[0, \infty)$ then for any $T < \infty$ we have that $\{u'_k(t)\}$, with $u'_k(t) := u_k(t)$, $0 \leq t \leq T$ (see Figure 2.2), has a limit point $\bar{u}' \in \mathcal{L}_2[0, T]$. In other words, for any sequence of functions, and any $T < \infty$, convergence on $[0, \infty)$ (in the sense of $\mathcal{L}_2[0, \infty)$) *implies* convergence on $[0, T]$ (in the sense of $\mathcal{L}_2[0, T]$). Obviously, the converse is not true.

The next example may be thought of as a typical ILC analysis problem. Let $G : \mathcal{L}_2[0, \infty) \mapsto \mathcal{L}_2[0, \infty)$ be some LTI system, and let $\{u_0, u_1, \ldots\}$ be

a sequence of functions (which we may assume to be defined on $\mathbb{R}_+$), such that $u_{k+1} = Gu_k$ for all $k \geq 1$, and $u_0 \in \mathcal{L}_2[0,\infty)$. Simultaneously, consider a sequence $\{u'_0, u'_1, \ldots\}$, with $u'_k(t) := u_k(t)$ for all $k \geq 0$, and $0 \leq t \leq T$ (assuming $T < \infty$).

It is well-known that the first sequence (the one defined over infinite time) converges for all initial inputs u_0 if and only if $\left\|G\right\|_\infty < 1$. With regard to the second sequence (the one defined over finite time), this same condition is also sufficient, but not necessary. Indeed, this latter sequence converges under much weaker conditions. To see this, let us introduce the notion of an *extended $\mathcal{L}_2$-space*. For any interval $I \subset \mathbb{R}$ and any $\lambda > 0$ we define the space $\mathcal{L}_2^\lambda(I)$,

$$\mathcal{L}_2^\lambda(I) := \left\{u : I \subset \mathbb{R} \mapsto \mathbb{R} \mid u(t)e^{-\lambda t} \in \mathcal{L}_2(I)\right\}, \tag{2.13}$$

and its associated norm

$$\left\|u\right\|^2_{\mathcal{L}_2(I)} := \int_{t\in I} [u(t)e^{-\lambda t}]^2 \mathrm{d}t. \tag{2.14}$$

We have the following result. If, for any $\lambda > 0$, the sequence $\{u_0, u_1, \ldots\}$ has a limit point $\bar{u} \in \mathcal{L}_2^\lambda[0,\infty)$, then the associated sequence $\{u'_0, u'_1, \ldots\}$, with $u'_k(t) := u_k(t)$, for all $k \geq 0$ and all $t \in [0,T]$, has a limit point $\bar{u}' \in \mathcal{L}_2^\lambda[0,T]$. Moreover, $\bar{u}'(t) = \bar{u}(t)$ for all $t \in [0,T]$. Given that, for any $T < \infty$ and all $\lambda > 0$, the spaces $\mathcal{L}_2^\lambda[0,T]$ and $\mathcal{L}_2[0,T]$ coincide, we conclude that, in order to prove convergence of the sequence $\{u'_k\}$, it suffices to show that there exist $\lambda > 0$ and $\bar{u} \in \mathcal{L}_2^\lambda[0,\infty)$ such that $\lim_{k\to\infty} u_k = \bar{u}$. We next derive conditions on G under which, for any given λ, the existence of a limit point is guaranteed.

For all $k \geq 0$ and all $t \in \mathbb{R}$, define $\tilde{u}_k(t;\lambda) := u_k(t)e^{-\lambda t}$. Application of the Laplace Transform yields the next identity

$$\begin{aligned}\tilde{u}_k(s;\lambda) &:= \int_0^\infty u_k(t)e^{-\lambda t}e^{-st}\mathrm{d}t \\ &= \int_0^\infty u_k(t)e^{-(\lambda+s)t}\mathrm{d}t \quad =: \quad u_k(s+\lambda),\end{aligned} \tag{2.15}$$

which holds for all $s > -\lambda$, since $u_k \in \mathcal{L}_2[0,\infty)$ for all $k \geq 0$. Using this identity it is easy to show that $\tilde{u}_k(s;\lambda)$ satisfies the next recurrence relation:

$$\tilde{u}_{k+1}(s;\lambda) = G(s+\lambda)\tilde{u}_k(s;\lambda). \tag{2.16}$$

We conclude that a sufficient condition for the sequence $\{\tilde{u}_0, \tilde{u}_1, \ldots\}$ to converge to a limit point in $\mathcal{L}_2^\lambda[0,\infty)$, and as such a sufficient condition for the sequence $\{u'_0, u'_1, \ldots\}$ to converge to a limit point in $\mathcal{L}_2[0,T]$ is given by

$$\sup_{Re(s)>0} \bar{\sigma}\left(G(s+\lambda)\right) < 1. \tag{2.17}$$

Since this same condition applies for all $\lambda > 0$, convergence is ensured if $\bar{\sigma}\left(G(\infty)\right) < 1$, a condition much less restrictive than its 'infinite-time' counterpart $\left\|G\right\| < 1$.

Having found a less restrictive condition is gain, but we would be negligent, not to mention the following. Our result states that under certain weak conditions on G, there exists some $\lambda > 0$ such that the sequence of functions defined on $\mathbb{R}_+$ converges monotonically in the sense of $\mathcal{L}_2^{\lambda}[0,\infty)$. The problem is that this kind of convergence is not the kind that is often asked for in practical applications. For we may note that, because of the weighting term involved, the functions $\{u_k\}$, though *bounded* on any finite time interval in both $\mathcal{L}_2$, and $\mathcal{L}_2^{\lambda}$-sense, may yet take *unacceptably large* values, even on $[0,T]$. Which is why we may prefer to work with the more restrictive 'infinite time' condition after all, since that condition at least guarantees monotone convergence in the usual, $\mathcal{L}_2[0,\infty)$-sense.

Finally, let us remark that in a *discrete-time* setting with $G(z)$ mapping $u_k(z)$ onto $u_{k+1}(z)$, the finite-time condition translates to $\bar{\sigma}\left(G(0)\right) < 1$.

Series

A *series* is a sum of terms. Let the terms in the series be denoted a_i. The k-th *partial sum* associated with the series is given by

$$S_k \;=\; \sum_{i=0}^{k} a_i. \tag{2.18}$$

We say that a series is *convergent* if the sequence of partial sums $S_0, S_1, \ldots$ converges. Of special interest to us (to ILC that is) is the *geometric series*. A geometric series $\sum_i a_i$ is a series in which the ratio between consecutive terms a_{i+1}/a_i is constant (alternatively: in which $a_{i+1} = ra_i$ for some r). Let r denote this ratio, then the partial sum S_k is given by

$$S_k \;=\; \left(1 + r + r^2 + \ldots + r^k\right) a_0. \tag{2.19}$$

It is not hard to show that in case $|r| < 1$, the sequence S_k converges to a limit

$$\lim_{k\to\infty} S_k \;=\; \left(\frac{1}{1-r}\right) a_0. \tag{2.20}$$

Series pop up in solutions to recurrence relations. Consider the general update law $u_{k+1} := u_k + \Delta u_k$. The solution to this first-order recurrence relation may be expressed in terms of a series involving the increment Δu_k:

$$u_{k+1} \;=\; u_0 + \sum_{i=0}^{k} \Delta u_i. \tag{2.21}$$

We can often prove convergence of a sequence $\{u_0, u_1, \ldots\}$, by investigating the series $\sum_i \Delta u_i$. Suppose for instance that the increment Δu_k satisfies a relation of the type $\Delta u_{i+1} = F(\Delta u_i)$, then it suffices to show that $\|F\| < 1$, where $\|\cdot\|$ is the induced operator norm.

2.1.3. Recurrence relations and iterations

Reccurence Relations

A *recurrence relation* (or simply recurrence) is a mathematical contruction, commonly deployed to express the dynamics of an algorithm. It relates current and past instances of certain (sets of) variables; it is the general term for what, in the context of ILC, is sometimes refered to as an *update law* or *learning rule*.

There are many types of such relations. Sedgewick and Fjajolet (1996) list a few: first, and higher-order; linear and nonlinear; constant, and variable coefficient; full-history, etc. Of particular interest to us is the class of *linear* recurrences.

Definition 2.1.4 (Linear recurrence). *Let U be a normed vector space. Given a set of initial conditions $\{u_i \in U : i = 0, 1, \ldots, N-1\}$ and some constant term $w \in U$. For all $k \geq 0$, we define the class of linear recurrences on U as*

$$u_{k+N} = F_1 u_{k+N-1} + F_2 u_{k+N-2} + \cdots + F_N u_k + w, \tag{2.22}$$

where the F_i's range over all bounded linear operators on U, i.e. $F_i \in \mathcal{L}(U)$, $i = 1, 2, \ldots, N$.

Linear recurrences have made frequent appearance in, and are characteristic of, the literature on 'classical ILC'—Norrlöf (2000).

Example 2.1.5 (A first-order, linear recurrence relation). *Let U, Y be normed vector spaces, and let $Q : U \mapsto U$, $P : U \mapsto Y$, and $L : Y \mapsto U$ be linear operators. Let $y_d \in Y$, and consider the following update law:*

$$\begin{aligned} u_{k+1} &= Qu_k + Le_k \\ &= (Q - LP)\, u_k + Ly_d. \end{aligned} \tag{2.23}$$

Though immaterial at this point, let us remark that, in an ILC setting, y_d would denote the desired output*; $y_k = Pu_k$ the* current output *(as opposed to y_{k+1} being the* future output*); u_k the* current input*; and $e_k := y_d - y_k$ the* current error*.*

Recurrence (2.23) is first-order*, as u does not depend on any but the previous instance of itself; has* constant coefficients*, because $(Q - LP)$ does not depend on k; is* linear *since its first (and only) coefficient, $(Q - LP)$, is a* linear *operator—see Definition 2.1.4.*

Higher-order ($N > 1$, Definition 2.1.4), as well as nonlinear relations with variable coefficients have been studied, but are not considered in this thesis. We focus on a class of first-order linear recurrences, like the one in (2.23). The general class of first-order linear recurrences (see Definition 2.1.4) is given as

$$u_{k+1} \;=\; Fu_k + d, \tag{2.24}$$

with $F \in \mathcal{L}(U)$ and $d \in U$. For relations of type (2.24), an explicit solution is easily obtained. Indeed, repeated substitution reveals that

$$u_k \;=\; F^k u_0 + \left(F^{k-1} + F^{k-2} + \cdots + F + I\right) d \;=\; F^k u_0 + S_{k-1}, \tag{2.25}$$

where the *partial sum* S_k is defined as $S_k := \sum_{i=0}^{k} F^i d$. After some manipulation, we arrive at

$$(I - F)\, S_k \;=\; \left(I - F^{k+1}\right) d. \tag{2.26}$$

Assuming $(I - F)^{-1}$ exists (as a bounded linear operator on U), we solve for S_k

$$S_k \;=\; (I - F)^{-1} \left(I - F^{k+1}\right) d, \tag{2.27}$$

and substitute (2.27) into (2.25), to obtain

$$u_k \;=\; F^k u_0 + (I - F)^{-1} \left(I - F^k\right) d. \tag{2.28}$$

Iteration

The mathematical notion of an *iteration* is very close to that of a recurrence relation. Note that the word 'iteration' is used both as a verb ("the process of iteration"), and as a noun ("an iteration"). We adopt the following definition.

Definition 2.1.6 (Iteration). *Let U be some vector space and let F be some operator that maps U into U. Iteration is the repeated application of F to itself. More precisely, given any u_0, iteration is the process of generating a sequence $\{u[0], u[1], u[2] \ldots\}$ such that*

$$u[k] \;=\; \begin{cases} u_0 & \text{if } k = 0, \\ Fu[k-1] & \text{if } k = 1, 2, \ldots \end{cases} \tag{2.29}$$

We say that Equation (2.29) defines an *iteration (noun). The operator F is sometimes refered to as the* transition map*.*

Definition 2.1.7 (Fixed point). *Let $F : U \mapsto U$ be any map. A point $u \in U$ is a* fixed point *of F if it satisfies $F(u) = u$.*

We say that an iteration *converges* if, for every initial condition $u_0 \in U$, there exists a fixed point $\bar{u}(u_0) \in U$ such that $\lim_{k\to\infty} u_k = \bar{u}(u_0)$. Usually we would want the iteration to converge to a *unique* fixed point $\bar{u}$ for all u_0. For specific spaces and certain classes of iterations, results on the existence and uniqueness of fixed points are available. One of the more commonly known results is *Banach's Fixed Point Theorem*, which we will review in Section 2.1.4.

Definition 2.1.6 suggests that an iteration is really a first-order recurrence relation, and one with constant coefficients at that. However, since U can be any (normed) space, this does not impose severe restriction on its applicability. The next example for instance, shows that within this framework 'trial-dependent' behavior is easily accommodated.

Example 2.1.8 (Accommodating trial-dependent behavior). *Let U be a normed vector space. Given $u_0 \in U$ and $F_0 \in \mathcal{L}(U)$. We consider the following update law:*

$$u_{k+1} \quad = \quad F_{k+1}(u_k) \quad ; \quad F_{k+1} \quad = \quad G(F_k, u_k) \qquad k = 0, 1, \ldots. \tag{2.30}$$

The map $G : \mathcal{L}(U) \times U \mapsto \mathcal{L}(U)$ assigns to each pair $(F, u) \in \mathcal{L}(U) \times U$ an element $G(F, u) \in \mathcal{L}(U)$. Define $z_k := (u_k, F_k)$, and denote its respective components as z_k^1, z_k^2. We rewrite (2.30) to obtain

$$z_{k+1} \quad = \quad (z_k^2(z_k^1), G(z_k^2, z_k^1)) \quad =: \quad F'(z_k). \tag{2.31}$$

Note that F' is a constant map (does not depend on k). Eqn. (2.31) defines an iteration *on the space $U \times \mathcal{L}(U)$.*

2.1.4. Fixed points and contraction operators

The next material is largely based on Agarwal et al. (2001). Let (U, d) be a metric space. A map $F : U \to U$ is said to be *Lipschitzian* if there exists a constant $\alpha \geq 0$ such that

$$d(F(u), F(u')) \quad \leq \quad \alpha\, d(u, u') \quad \text{ for all } u, u' \in U. \tag{2.32}$$

The smallest α for which (2.32) holds is said to be the *Lipschitz constant* for F. We denote it by ρ. If $\rho < 1$ we say that F is a *contraction*, and if

$\rho \leq 1$, we say that F is *nonexpansive*. In the context of a Banach Space (such as $\mathcal{L}_2$), the metric may be replaced by a norm. In that case we say that a map F is a contraction if there exists $\alpha < 1$ such that

$$\|F(u) - F(u')\| \;\leq\; \alpha \|u - u'\| \quad \text{for all } u, u' \in U. \tag{2.33}$$

For (bounded) linear operators, this is equivalent with saying that $\|F\| < 1$.

A map $F : U \to U$ may have one fixed point, more than one, or none at all. The following theorem gives a sufficient condition for F to have a *unique* fixed point in U, along with an iterative procedure to compute it.

Theorem 2.1.9 (Banach's Fixed Point Theorem). *Let (U, d) be a complete metric space and let $F : U \to U$ be a contraction. Then F has a unique fixed point $\bar{u} \in U$. Moreover, for any $u \in U$ we have*

$$\lim_{k \to \infty} F^k(u) \;=\; \bar{u}. \tag{2.34}$$

2.2. Literature overview

We review some results from the ILC literature most relevant to us. First we present a brief history and timeline. Then we discuss some conceptual frameworks. Additional references can be found in e.g. Moore (1993, 1999); Bien and Xu (1998); Chen and Wen (1999); Xu and Tan (2003).

2.2.1. Brief history

Origins

Looking at ILC from an evolutionary perspective, the original work of Arimoto et al. (1984) can certainly be identified as *a*, if not *the* Big Bang. As it appears, the key ingredients of what we now know as ILC were already present in the 'primal soup' of their first paper.

In that 1984 paper, entitled "Bettering Operation of robots by learning", the authors set out to propose "a practical approach to the problem of bettering the present operation of mechanical robots". The introduction draws a lively picture of human beings, engaged in a process of learning. Upon reflection, the authors are led to ask:

> "Is it possible to think of a way to implement such a learning ability in the automatic operation of dynamic systems?"

Although the paper does not provide an answer, the results suggest a 'yes'.

Arimoto and co-workers were by no means the first to write on the subject of Learning Control. We mention the early work of the Russian author Tsypkin (1971, 1973), who two books, respectively entitled "Foundations of the Theory of Learning Systems" and "Adaptation and Learning in Automatic Systems", both date back to the early 1970's. We can even go back as far as the late 1940's through the mid 1960's, when related ideas appear in the Cybernetics literature (see Chapter 1). Of more recent nature, but still predating the work of Arimoto *et al.*, is the theory of *multipass processes*, which was developed by Edwards and Owens (1982) in the late 70's and early 80's.

Given all these contributions, one wonders: what is so special about the aforementioned paper? We believe Arimoto's approach to Learning Control stands out because of its *transparancy* and *simplicity*. Stripped to the essential, it evades the common pitfalls of useless generality and needless complexity. As such, it is found to be very effective and has inspired a great many researchers.

Intermezzo: The original algorithm of Arimoto et al.

Let us interrupt our current pursuit for a while, and have a closer look at Arimoto's original algorithm. Our aim is to highlight main ideas, and we will discuss but few results in detail. The interested reader is refered to Arimoto et al. (1984). Picking up the paper where we left it earlier, we are presented with an introductory example in which a simple voltage-controlled dc motor is to track a given angular velocity profile $\omega_{\mathrm{d}}(t)$ ('d' from 'desired') during some interval of time $[0, T]$. The objective is to find the corresponding input voltage $V_{\mathrm{d}}(t)$. This would not be a hard problem, if not for the following assumption:

> "...we implicitly assume that the description of the system dynamics is unknown..."

The use of the word 'implicit' may be a bit confusing at first. Presumably, what the authors meant to say is that, though the paper uses an *exact* description of the system dynamics, it does not actually assume to know the values of the various parameters. Whatever it is, the problem is clear: how does one determine the required supply to an unknown system?

The authors propose an *iterative* scheme ("betterment process") in which the current (as in 'present') supply $V_k(t)$ —note the index k, indicating the iteration number—is computed from the previous supply $V_{k-1}(t)$ and the associated error in the resulting angular velocity $\omega_{\mathrm{d}}(t) - \omega_{k-1}(t)$. It is shown

that under certain conditions, the actual supply converges to the desired supply.

Later on, a more general class of linear, time-varying systems is considered,

$$y(t) \;=\; g(t) + \int_0^t H(t,\tau)u(\tau)\mathrm{d}\tau \tag{2.35}$$

along with the following iterative scheme:

$$u_{k+1}(t) \;:=\; u_k(t) + \Gamma(t)\dot{e}_k(t). \tag{2.36}$$

Here, $e_k(t) := y_\mathrm{d}(t) - y_k(t)$ denotes the current tracking error and $\Gamma(t)$ is some time-varying 'learning parameter'. This scheme is sometimes refered to as a D-type learning rule, because of its derivative action.

Under what conditions does the given update scheme (2.36) converge? The answer is as follows (omitting details). Suppose $y_\mathrm{d}(0) = g(0)$. If

$$\sup_{t\in[0,T]} \left\| I - \Gamma(t)H(t,t) \right\| \;<\; 1, \tag{2.37}$$

where $\left\|\cdot\right\| = \max_i \sum_j \left|(\cdot)_{ij}\right|$ is the 'maximum absolute row sum'-norm, then $\lim_{k\to\infty} y_k = y_\mathrm{d}$, the convergence being uniform in t on $[0,T]$. Condition (2.37) ensures *monotone* convergence in the λ-norm,

$$\|w\|_\lambda \;:=\; \sup_{t\in[0,T]} |w(t)e^{-\lambda t}| \tag{2.38}$$

and, in general, does *not* ensure monotone convergence in $\mathcal{L}_\infty$, or $\mathcal{L}_2$-sense.

The remainder of the paper is about the application of the "betterment process" to the control of a manipulator. In conclusion, the authors write:

> "With the goal of applying 'learning' control to robot manipulators, we proposed an iterative betterment process for mechanical sytems that improves their present performance of motion by use of updated data of the previous operation."

Progress—a brief history continued

Since its early appearence in Arimoto et al. (1984), the original algorithm for Learning Control has been modified and extended in, we dare say, all possible ways. By now, there are so many different algorithms that it has become extremely hard to come up with an adequate classification key. For a detailed overview, the interested reader is referred to Moore (1999); We just take the bird eye's view.

Inspection of the literature shows that in the early years (mid to late 1980's), research was primarily devoted to such questions as: "Under what conditions does this algorithm converge?", "To what class of systems is it applicable?", "Is it robust with respect to resetting errors", etc. Moore (1993) lists some early contributions.

Though bold to say, it appears that, at the time, performance was not really an issue, for either an algorithm would converge to zero error, or it would not converge at all. There was the issue of monotone convergence (rather, the lack thereof), as well as issues related to imperfect resetting, but these do not compare with the synthesis problems that came up in the 90's. In terms of ILC research, the 90's are characterized by a growing awareness of a tradeoff between performance and robustness. This itself may have been a result of the paradigm shift that was taking place in the general field of control. Following the mainstream control community, researchers in ILC adopted what Zhou et al. (1996) have described as a 'postmodern control' point of view, along with its terminology, and associated analysis techniques. We refer to Padieu and Su (1990); Kavli (1992); Moore et al. (1992); de Roover (1996); Amann et al. (1996c). In these (and other) papers, the problem of ILC is recast as an $\mathcal{H}_\infty$ synthesis problem in one or two parameters (typically *causal* operators). Owens et al. (1995) detail the *status quo* in the early 90's, covering, among others, a 2D Systems Theory approach to ILC, as well as Norm-optimal ILC (both of which will be discussed in the next section). All in all, we concur with the suggestion made by Norrlöf (2000) that in the past decade, the focus in ILC research has shifted from typical analysis questions to questions of synthesis and design.

2.2.2. Conceptual frameworks—paradigms for learning control

By now, we have a rough idea of the sorts of algorithms (linear, nonlinear; first-order, higher-order) that are out there. Next, we want to look at some of the conceptual frameworks behind these algorithms. The basic question we ask is: why do we look at certain algorithms of a particular kind?

It will become clear that different authors have different ideas of what ILC really is about. No less our choice of presentation reflects a bias towards certain paradigms.

Arimoto et al—a betterment process revisited

As mentioned in Section 2.2.1, Arimoto's original idea was to propose a betterment process for robotic manipulators. And that is what he did: he

proposed a learning rule, a D-type learning rule, to be precise. At first sight, there is a certain arbitrariness to this learning rule. Why is the increment chosen proportional to the derivative of the error? Arimoto et al. (1984) provide ample motivation for looking at betterment processes, but do not explain why a betterment process should take the particular form that it does. Let the following serve as a partial answer. For any SISO real-rational system $G(s)$ with relative degree one ($\lim_{s\to\infty} sG(s)$ exists and is unequal to zero), there exists $\gamma \in \mathbb{R}$ such that

$$\lim_{s\to\infty} \left|1 - \gamma s G(s)\right| \;<\; 1. \tag{2.39}$$

Now consider the update law

$$u_{k+1} \;=\; u_k + \gamma \dot{e}_k. \tag{2.40}$$

After applying the Laplace Transform on both sides we arrive at

$$u_{k+1}(s) \;=\; u_k(s) + \gamma s e_k(s), \tag{2.41}$$

which we rewrite to obtain

$$e_{k+1}(s) \;=\; (1 - \gamma s G(s))\, e_k(s). \tag{2.42}$$

Using a result from Section 2.1.2 we conclude that the sequence $\{e_k(t)\}$ converges to zero over *any* finite time interval, provided γ is such that Condition (2.39) is satisfied. We conclude that the rationale behind this algorithm is based on two assumptions:

1. that the system's relative degree is unity;

2. that at least the sign (and preferably the magnitude) of $\lim_{s\to\infty} sG(s)$ is known (with $(A, b, c, 0)$ being a realization of G, this amounts to knowing the sign of the first Markov parameter cb).

Vector space optimization—gradient type algorithms

Given that the larger part of the ILC literature is dedicated to the study of first-order linear recurrences, we wonder: what is it about these algorithms that makes them so special? One particularly appealing answer is that we can view them as *gradient-type algorithms* associated with the minimization of some cost functional. Or at least as their offspring, we can.

Let U and Y be Hilbert spaces and let P (the plant) be a bounded linear map from U to Y. Given $y_{\mathrm{d}} \in Y$ (the desired output), the objective of ILC

(as viewed within this framework) is to find the input u^{opt} that minimizes the associated cost functional

$$J(u) \;=\; \|y_{\text{d}} - Pu\|_Y^2. \tag{2.43}$$

If, for whatever reason, we are unable to determine the minimum analytically, we may consider iterative methods of optimization. We could then ask: for a given u, what is the optimal increment Δu? With the latter optimization problem we associate the functional

$$J'(\Delta u) \;=\; \|y_{\text{d}} - P(u + \Delta u)\|^2. \tag{2.44}$$

What is the minimizing argument? Define $e := y_{\text{d}} - Pu$. From Luenberger (1969) we know that if a solution Δu^{opt} exists, it satisfies

$$P^*P\Delta u^{\text{opt}} \;=\; P^*e, \tag{2.45}$$

where P^* denotes the adjoint of P. If P has a bounded inverse, then so does P^*P (and vice versa). Let us suppose it does, and solve for Δu^{opt}

$$\begin{aligned} \Delta u^{\text{opt}} \;&=\; (P^*P)^{-1} P^*e \\ &=\; P^{-1}e. \end{aligned} \tag{2.46}$$

In explicit form, the update law reads as

$$u_{k+1} \;=\; u_k + P^{-1}e_k. \tag{2.47}$$

It is not hard to see that it takes just one step for this algorithm to reach its fixed point

$$\bar{u} \;=\; P^{-1}y_{\text{d}}. \tag{2.48}$$

This fixed point is in fact the solution to the original optimization problem $\min_{u\in U} J(u)$.

The idea to associate the problem of ILC with the minimization of some cost functional was particularly advocated in a series of publications by Amann *et al.* on *norm-optimal ILC*. Their approach aims at reducing the error $\|y_{\text{d}} - Pu\|$ whilst bounding the change in control input. To that end, they consider the problem of minimizing the functional

$$J(\Delta u;\gamma) \;=\; \left\| \begin{matrix} y_{\text{d}} - P\,(u + \Delta u) \\ \gamma\Delta u \end{matrix} \right\|^2, \tag{2.49}$$

where γ is some positive weight. A solution can be obtained as follows. First we rewrite (2.49) in the form "$y - Ax$"

$$J(\Delta u;\gamma) \;=\; \left\| \begin{bmatrix} e \\ 0 \end{bmatrix} - \begin{bmatrix} P \\ \gamma I \end{bmatrix} \Delta u \right\|^2. \tag{2.50}$$

Then we apply the same arguments as before. So again, if a minimizing solution Δu^{opt} exists, it satisfies

$$\begin{aligned} \begin{bmatrix} P^* & \gamma I \end{bmatrix} \begin{bmatrix} P \\ \gamma I \end{bmatrix} \Delta u^{\text{opt}} &= \begin{bmatrix} P^* & \gamma I \end{bmatrix} \begin{bmatrix} e \\ 0 \end{bmatrix} \\ &\Leftrightarrow \\ \left(P^*P + \gamma^2 I\right) \Delta u^{\text{opt}} &= P^* e. \end{aligned} \tag{2.51}$$

Let us suppose that the operator $(P^*P + \gamma^2 I)$ is boundedly invertible on U. We obtain the update law

$$u_{k+1} = u_k + \left(P^*P + \gamma^2 I\right)^{-1} P^* e_k. \tag{2.52}$$

In case P itself is boundedly invertible, Equation (2.52) takes the interesting form

$$u_{k+1} = u_k + \gamma^{-1} \left(\gamma^{-1} P + \gamma \left(P^*\right)^{-1}\right)^{-1} e_k. \tag{2.53}$$

If we substitute $\gamma = 0$ into (2.52), we obtain (2.46) as a special case. Interestingly enough, Amann et al. (1996a) arrive at a different expression:

$$u_{k+1} = u_k + \gamma^{-2} P^* e_{k+1}. \tag{2.54}$$

In some references, such as Owens et al. (1995) and Amann et al. (1996b), the γ-parameter is missing. Presumably, these references implicitly assume that $\gamma = 1$. Now, how do we account for the fact that in minimizing the exact same functional we arrive at different expressions? The answer is that these expressions are not so different as they appear to be. To see this, note that

$$e_{k+1} = e_k - P\Delta u_k. \tag{2.55}$$

Hence

$$P^* e_k = P^*P\Delta u_k + P^* e_{k+1}. \tag{2.56}$$

Substituting (2.56) into (2.52) and solving for Δu_k gives us update law (2.54). Note that, in (2.54), the adjoint operator is feeding back future errors. Thus it seems that, to run this scheme, we would need to know the error ahead of time (which we do not). In Owens et al. (1995) however, it is shown that, with considerable effort, the algorithm may be recast in causal form. Even so, in practice we would prefer to use the alternative scheme (2.52) as it is already in causal form (the noncausal operator, $\left(P^*P + \gamma^2 I\right)^{-1} P^*$ acts on the *past* error, e_k only) and has a relatively straightforward implementation.

Algorithms such as the ones above are examples of so called gradient-descent methods. Gradient methods are commonly used to compute the minimum of a function or functional $J : U \mapsto \mathbb{R}$. The general idea is as follows. Starting from some point u_0, we compute the gradient $\nabla J(u_0)$. Then we take a step Δu in the direction of the negative gradient

$$\Delta u \propto -\big[\nabla J(u_0)\big] \tag{2.57}$$

in the hope to descent along the cost surface. The negative gradient of the functional $J(u)$ defined in (2.43) is given as

$$-\nabla J(u_0) \;=\; 2P^*e. \tag{2.58}$$

We could use (2.58) to build a whole *class* of update laws:

$$u_{k+1} \;=\; u_k + \gamma P^* e_k, \tag{2.59}$$

with γ some fixed (positive) scalar. More general update laws can be obtained by replacing γ with some arbitrary (positive) linear operator. A particular instance of such a learning rule is given by (2.52).

One may note that each of the update laws considered requires precise knowledge of the plant dynamics. Of course, in engineering practice, such extensive knowledge is rare. For that reason people frequently consider generalized update laws, similar in structure, yet not confined to any plant in particular, i.e. update laws of the form:

$$u_{k+1} \;=\; u_k + Le_k. \tag{2.60}$$

By introducing this *free* parameter L, the problem of Iterative Learning Control is reduced to a one-parameter synthesis problem, where the objective is to pick L in such a way as to make (2.60) converge. In this respect it matters whether we allow L to anticipate or not. Would we insist on causality, convergence conditions would become restrictive. Indeed, for convergence in the sense of $\mathcal{L}_2[0,\infty)$ we would need L to satisfy

$$\|I - LP\|_\infty \;<\; 1; \tag{2.61}$$

This condition can be relaxed to

$$\|I - L(\infty)P(\infty)\| \;<\; 1 \tag{2.62}$$

in case we settle for convergence in some extended space $\mathcal{L}_2^\lambda$. In order for (2.61) to be satisfied for some $L \in \mathcal{RH}_\infty$, we need $P^{-1} \in \mathcal{RH}_\infty$. Condition (2.62), though less restrictive, still requires $P(\infty)$ to be full column rank

and is thus never satisfied for strictly proper plants. This does not appear a good starting point for a synthesis procedure.

Basically, there are two ways out. One is to allow for noncausal operators (as we will do in Chapter 4). The other is to introduce yet another parameter:

$$u_{k+1} = Qu_k + Le_k. \tag{2.63}$$

The problem of Learning Control now becomes a *two*-parameter synthesis problem. Again, we would need to decide on a suitable range, for both L and Q. But this time, even if we contrain both parameters to be causal bounded linear operators, the problem remains well-posed. That is to say, if P is a bounded linear operator, there is always a parameter pair (Q, L), such that (2.63) converges to a bounded solution $\bar{u}$.

Note however that by introducing this extra parameter Q, we somehow give up perfect tracking; In the one-parameter problem, the error converges to zero by construction (if it converges at all). Not so in the two-parameter problem.

In Section 2.3 we return to this family of update laws and the associated synthesis problem.

2D system theory

An nD system is a system in which the information propagates in n seperate directions. In ILC we can recognize two such directions: *time* t and *iteration* k. The following example is based on Kurek and Zaremba (1993).

Example 2.2.1. *Consider a discrete-time system*

$$\begin{cases} x(t+1,k) = Ax(t,k) + Bu(t,k) \\ y(t,k) = Cx(t,k) \end{cases} \tag{2.64}$$

The boundary conditions for this system are given as $x(0,k) = x_0$ *for* $k = 0, 1, \ldots$ *and* $u(t,0) = 0$ *for* $t = 0, 1, \ldots, N$. *The objective is to track a certain output* y_d. *More precisely, we want to design a 2D controller (an update law) such that* $y(t,k) \to y_d(t)$ *for all* $t \in [1, N]$. *With* $\Delta u(t,k)$ *yet to be defined, we consider a general update equation of the form*

$$u(t,k+1) = u(t,k) + \Delta u(t,k) \tag{2.65}$$

Define $e(t,k) := y_d(t) - y(t,k)$. *Combining (2.65) with (2.64), we arrive at*

$$e(t,k+1) - e(t,k) = -CA\eta(t,k) - CB\Delta u, \tag{2.66}$$

where

$$\eta(t,k) \;:=\; x(t-1,k+1) - x(t-1,k). \tag{2.67}$$

Combining (2.67) and (2.64), we obtain

$$\eta(t+1,k) \;=\; A\eta(t,k) + B\Delta u(t-1,k). \tag{2.68}$$

Equations (2.66) and (2.68) may be rewritten in compact form

$$\begin{bmatrix}\eta(t+1,k)\\ e(t,k+1)\end{bmatrix} \;=\; \begin{bmatrix}A & 0\\ -CA & I\end{bmatrix}\begin{bmatrix}\eta(t,k)\\ e(t,k)\end{bmatrix} + \begin{bmatrix}B\\ -CB\end{bmatrix}\Delta u(t-1,k) \tag{2.69}$$

The objective is to drive this system to zero. We deploy the following 2D compensator.

$$\Delta u(t,k) = Ke(t+1,k) \tag{2.70}$$

The closed-loop system dynamics are given as:

$$\begin{bmatrix}\eta(t+1,k)\\ e(t,k+1)\end{bmatrix} \;=\; \begin{bmatrix}A & BK\\ -CA & I-CBK\end{bmatrix}\begin{bmatrix}\eta(t,k)\\ e(t,k)\end{bmatrix}. \tag{2.71}$$

In the theory of 2D systems, the above model is known as a Roesser model. Using specific tools developed for this class of models, it can be shown that the overall system is stable (convergent) if and only if $I-CBK$ is asymptotically stable (has all its eigenvalue in the open unit circle).

The above example shows that the problem of ILC can be cast as a particular 2D stabilization problem. In general, 2D systems theory provides a solid framework for Iterative Learning Control. But then again, to solve a typical ILC problem, we may not need the *complete* 2D system toolbox.

> "It is not necessary to bring the full power of 2D systems to the problem, because ILC is really a special case in which one of the dimensions (time) is a finite, fixed interval. As a result, stability in that direction is [...] always assured. Thus the direction in which stability needs to be studied is in repetition."
>
> —Moore (2000)

For more details on the 2D system theory approach to ILC we refer to e.g. Amann et al. (1994); Rogers et al. (2002).

Internal Model Control

The Internal Model Principle (IMP), see Francis and Wonham (1975), refers to an important result in the theory of controller synthesis. Informally, the IMP states that, for asymptotic tracking of a signal, the controller must contain a model of that signal. In particular, if the tracking signal so happens to be *periodic*, the controller must contain a model of that periodicity. If we then agree that ILC—and even more so repetitive control—is about tracking periodic signals, we got ourselves a new paradigm for Learning Control. This point of view is most clearly expressed in (and probably due to) de Roover et al. (2000). In that paper, it is shown how the design of an ILC scheme can be incorporated into a general compensator design problem. The main idea is to append the closed loop with the appropriate filters, and a delay block z^{-N} which implements a storage element (memory block). The paper offers a synthesis procedure for solving the *robust periodic control problem*, which is: to asymptotically reject all periodic disturbances. The resulting compensator is a high-order controller that includes a model of the disturbance.

In a slightly different context, Moore (2000) deployed the IMP to reason about first- and higher-order update laws. He showed that, by introducing a *shift variable* w, acting on the iteration variable k,

$$(w^{-1}u_k)(t) \;=\; u_{k-1}(t) \tag{2.72}$$

the problem of ILC boils down to that of designing a compensator $C(w)$. Expressed as a rational transfer function in w, it relates past and current instances of the input and the error.

$$u_k(t) \;=\; C(w)e_k(t) \tag{2.73}$$

Note that the plant P is static in terms of w. In this case, the IMP informs us that if ILC is to reject a constant (again, in terms of w) disturbance, the compensator $C(w)$ must contain a pure integrator.

2.3. ILC: a two-parameter synthesis problem

As announced in Subsection 2.2.2, this final section poses the problem of ILC as a synthesis problem on the space of causal bounded linear operator pairs.

2.3.1. The problem statement

Let U, Y be vector spaces. Assume U is *normed*, and let $Y_{\rm n} \subset Y$ be a *normed subset* of Y. The problem of ILC may be cast as follows (see Figure 2.3). Given a plant $P : U \mapsto Y$, along with some desired output $y_{\rm d} \in Y_{\rm n}$: the objective of ILC is to construct a sequence of inputs $\{u_0, u_1, \ldots\}$, such that, with $y_k := Pu_k$,

1. $u_k \in U$ for all $k \geq 0$, and there exists $\bar{u} \in U$ such that $\lim_{k\to\infty} u_k = \bar{u}$.

2. $y_k \in Y_{\rm n}$ for all $k \geq 0$ and there exists $\bar{y} \in Y_{\rm n}$ such that $\lim_{k\to\infty} y_k = \bar{y}$. In addition, $\left\|y_{\rm d} - \bar{y}\right\|_{Y_{\rm n}}$ is 'small'.

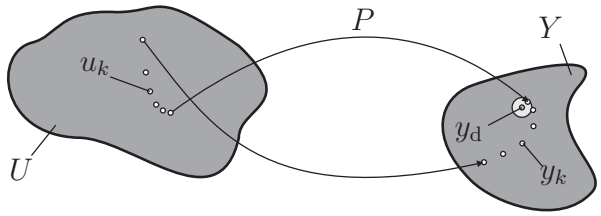


Figure 2.3.: Given $P : U \mapsto Y$ and $y_{\rm d} \in Y$. The problem of ILC is to construct a sequence of inputs $\{u_k\}$ such that the corresponding sequence of outputs $\{y_k\}$ converges to some immediate neighborhood of $y_{\rm d}$.

More specifically, the problem of ILC is to construct an *iteration*, or simply a map $F : U \to U$ such that: (a) There exist one, and just one $\bar{u} \in U$ such that $F(\bar{u}) = \bar{u}$; (b) for all $u_0 \in U$, $\lim_{k\to\infty} u_k = \bar{u}$; (c) there exist $\bar{y} \in Y_{\rm n}$ such that $\lim_{k\to\infty} Pu_k = \bar{y}$; (d) The 'error' $\left\|y_{\rm d} - \bar{y}\right\|_{Y_{\rm n}}$ is small. In this thesis, we consider one particular family of iterations $F(Q, L; C) : \mathcal{L}_2(\mathbb{R}_+) \mapsto \mathcal{L}_2(\mathbb{R}_+)$,

$$\begin{aligned} u_{k+1} &= F(Q, L; C)u_k \\ &= Qu_k + Le_k + Ce_{k+1}. \end{aligned} \tag{2.74}$$

As before, e_k denotes the current tracking error. With $U = Y_{\rm n} = \mathcal{L}_2(\mathbb{R}_+)$, the free parameters $Q : U \mapsto Y_{\rm n}$ and $L : Y_{\rm n} \mapsto U$ range over all causal, bounded finite-dimensional LTI operators. That is, we assume $Q(s), L(s) \in \mathcal{RH}_\infty$. The parameter C is fixed. It represents a feedback compensator and is assumed to internally stabilize the closed loop. Which is to say that we

assume the next condition to hold:

$$\begin{bmatrix} I & C \\ -P & I \end{bmatrix}^{-1} = \begin{bmatrix} (I+CP)^{-1} & -C\,(I+PC)^{-1} \\ P\,(I+CP)^{-1} & (I+PC)^{-1} \end{bmatrix} \tag{2.75}$$
$$= \begin{bmatrix} I - C\,(I+PC)^{-1}\,P & -C\,(I+PC)^{-1} \\ (I+PC)^{-1}\,P & (I+PC)^{-1} \end{bmatrix} \in \mathcal{RH}_\infty.$$

The plant P need not be stable. Neither do we require C to be stable. In Chapter 4, we extend the range of Q and L to all bounded LTI operators, not necessarily causal. If asked for, we could even drop the assumption of boundedness. That is to say, well-posedness does not require Q and L to be bounded on $[0,\infty)$ (at least not in the sense of $\mathcal{L}_2[0,\infty)$). We will nonetheless demand that both operators be bounded on $[0,\infty)$. The reason is twofold: (a) Depending on the window size T, an operator bounded on $[0,T]$, but unbounded on $[0,\infty)$, may have unacceptably large gain (unacceptable for practical applications that is); (b) There may be no need to resort to unbounded (causal) operators, since the same control action can be effected by bounded *non*causal operators. For instance, for all $s \in \mathbb{C}$ with $\mathrm{Re}(s) < 0$, let $G(s)$ be given as $G(s) = 1/(s-1)$. The map G corresponds to a system

$$z(t) = (Gw)(t) := \int_{-\infty}^{\infty} g(t-\tau)w(\tau)\mathrm{d}\tau,$$

where $g(t)$ is defined as

$$g(t) := \begin{cases} -e^t & t < 0 \\ 0 & t \geq 0 \end{cases} \tag{2.76}$$

Note that $g \in \mathcal{L}_1$, which implies that G is BIBO stable—in the sense of $\mathcal{L}_2(\mathbb{R}) \mapsto \mathcal{L}_2(\mathbb{R})$. In a similar fashion, every real-rational transfer function with no poles on the imaginary axis (that is, every $G \in \mathcal{RL}_\infty$) can be implemented as a bounded operator, not necessarily causal.

The next lemma shows that the class of recurrences defined by (2.74) is well-defined in the sense that each member maps bounded input-output pairs onto bounded input-output pairs.

Lemma 2.3.1. *Consider the class of recurrences (2.74). Let $y_d \in \mathcal{H}_2$. Suppose (P,C) satisfies (2.75). Then for all $Q, L \in \mathcal{RH}_\infty$, $(u_k, y_k) \in \mathcal{H}_2 \times \mathcal{H}_2 \Rightarrow (u_{k+1}, y_{k+1}) \in \mathcal{H}_2 \times \mathcal{H}_2$.*

Proof. Suppose $(u_k, y_k) \in \mathcal{H}_2$. Define $z_k := Qu_k + Le_k$. With $Q, L \in \mathcal{RH}_\infty$, it follows that $z_k \in \mathcal{H}_2$. To compute (u_{k+1}, y_{k+1}), we solve

$$\begin{aligned} u_{k+1} &= z_k + C(y_d - y_{k+1}) \\ y_{k+1} &= Pu_{k+1}, \end{aligned} \tag{2.77}$$

the solution of which is given by

$$\begin{aligned} u_{k+1} &= (I+CP)^{-1} z_k + (I+CP)^{-1} y_d \\ y_{k+1} &= P(I+CP)^{-1} z_k + P(I+CP)^{-1} y_d. \end{aligned} \tag{2.78}$$

By assumption, (P, C) satisfies (2.75), and with $z_k, y_d \in \mathcal{H}_2$, it follows that $u_{k+1}, y_{k+1} \in \mathcal{H}_2$. This concludes the proof. ■

2.3.2. The Standard ILC Problem

We now assume $P(s) \in \mathcal{RH}_\infty$ and turn our attention to a subclass of iterations $F(Q, L; 0)$, compare (2.74):

$$u_{k+1} = Qu_k + Le_k. \tag{2.79}$$

This class of iterations is representative of what we call Standard ILC, and what Norrlöf (2000) refers to as Classical ILC. Figure 2.4 shows the corresponding schematic. Note that the future control action u_{k+1} is solely determined by the current control input u_k and the current error e_k; The *direct feedback term* Ce_{k+1} has been eliminated.

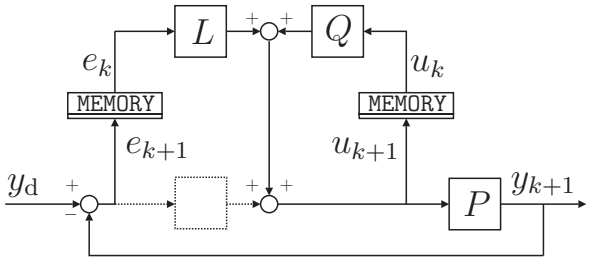


Figure 2.4.: Schematic of Standard ILC. The future input u_{k+1} is constructed from the current input u_k and the current error e_k. In standard ILC, the current cycle term, indicated by the empty dashed box, is absent. Compare with Figure 2.5.

We define the Standard ILC synthesis problem as the problem of selecting $Q, L \in \mathcal{RH}_\infty$ such that

1. the iteration $F(Q, L; 0)$, defined in (2.79), converges to a fixed point $\bar{u} := \lim_{k\to\infty} u_k$.

2. the asymptotic error $\bar{e} := y_d - P\bar{u}$ is small.

Literature on the Standard ILC Problem

The family of iterations (2.79) and the associated synthesis problem have received considerable attention. Early contributions are due to, among others, Padieu and Su (1990); Liang and Looze (1993); Amann et al. (1996c). Moore et al. (1992) studied its convergence properties and derived a lower bound on the maximally achievable performance by relating it to the problem of determining a best *approximate inverse*. Kavli (1992) discussed robustness and derived a quasi-heuristic synthesis procedure, on which de Roover (1996) and de Roover and Bosgra (2000) improved, using $\mathcal{H}_\infty$ optimization techniques.

All in all, there is a good degree of consensus on how to choose the respective design parameters: The parameter L should somehow resemble the approximate inverse of P, whereas Q is typically taken to be a lowpass filter, understood to improve robustness by suppressing the effect of model uncertainty at high frequencies.

Yet, the interplay between the two parameters is not fully understood. One can argue that Q should be 'close to unity' in some sense. By choosing Q to have lowpass characteristics, good tracking can be ensured in the low frequency range (where Q is close to unity). Then, for a given Q, the parameter L can be 'matched' to the inverse of the given plant by solving what is essentially a model matching problem—de Roover (1996). This kind of *two-step approach* is common to all existing procedures. Though very effective, it lacks rigorous justification. By fixing one of the parameters, the design problem becomes easier, but generality is lost.

In the next chapter we show that the problem of ILC (taken to be the above) is *overparameterized* and can be reduced to a 1-parameter problem without loss of generality. Finally, we remark that some references work with the slightly modified update law

$$u_{k+1} \;=\; Q\,(u_k + Le_k)\,. \tag{2.80}$$

It is not hard to see that, under the conditions stated ($Q, L \in \mathcal{RH}_\infty$), the original family (2.79) includes the latter (2.80), but not vice versa.

2.3.3. ILC with current-cycle feedback

We now return to the full class, as defined in (2.74). This class is representative of *current-cycle feedback ILC* (CCF-ILC) or *current-iteration tracking error* ILC (CITE-ILC). The basic schematic is shown in Figure 2.5. The idea of including a direct feedback term (Ce_{k+1}) has a long history. Instances can be found in the work of Chen, e.g. Chen et al. (1996a,b, 1997a,b). We

also mention the work of Owens (1992) who considered ILC in conjuction with high-gain feedback.

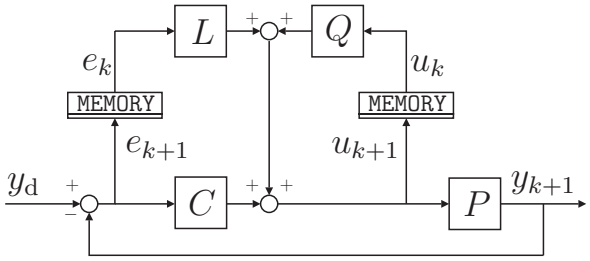


Figure 2.5.: Schematic of CCF-ILC. Compare with Figure 2.4. Note the current-cycle term Ce_{k+1}.

In the CCF-ILC problem, the parameter C is assumed to be fixed. This assumption is motivated by the fact that in many applications, an existing controller can be appended, but not altogether replaced.

We define the problem of CCF-ILC as the problem of finding $Q, L \in \mathcal{RH}_\infty$ such that:

1. The sequence of IO pairs $\{(u_k, y_k)\}$ converges to some limit point $(\bar{u}, \bar{y}) \in \mathcal{H}_2 \times \mathcal{H}_2$;

2. The asymptotic error $\bar{e} := y_d - \bar{y}$ is small.

2.4. Summary and preview

We provided relevant background material and reviewed some of the literature on Iterative Learning Control (ILC). In addition, we formally posed the problem of ILC as a two-parameter synthesis problem on the space of (causal) bounded linear operator pairs. In the next two chapters, the synthesis problem is studied in greater detail, under various assumptions. Then, in Chapter 5, we ask in what possible sense the algorithms we study show a 'learning behavior'.

3

Causal ILC and Equivalent Feedback

Overview – *We present a detailed analysis of the two-parameter synthesis problem (within the scope of this chapter simply 'the problem of ILC'). We require that both parameters (operators) be causal, and discuss how this particular constraint affects the synthesis problem. One of our findings states that, under the conditions indicated, the problem of ILC, and that of conventional feedback control are very much akin, not to say essentially equivalent.*

3.1. Introduction

In the preceding chapter we outlined various approaches to the problem of Learning Control, expressing particular interest in a family of first-order recurrences (Section 2.3). This same class of recurrences forms the backbone of the present chapter. We address the problem of parameterizing its stable members and consider how, from among these members, we may select the ones that give best performance.

The question of optimality ("What parameter combination yields the best performance?") depends on possible constraints on the parameters involved. Apart from boundedness, the one constraint we impose on each of the parameters (which are really linear operators) is that they be *causal*. We will see that this assumption has far-reaching consequences. For it turns out that under this very constraint, the problem of ILC becomes equivalent with that of conventional feedback control. How we may understand this equivalence, and what the implications are, will become clear as we proceed.

The outline of this chapter is as follows. In Section 3.2 we review some aspects of fixed point analysis so as to motivate the main ideas. Then, in Section 3.3, we define the notion of an admissible pair, which we deploy in Sections 3.4 and 3.5 to address the respective problems of Standard ILC and CCF-ILC. Section 3.6 reports on the results obtained in an experimental study. Section 3.7 closes with a discussion of the main results. The chapter is based on Verwoerd et al. (2003, 2004a,c).

3.2. Fixed Point Analysis

Fixed point analysis, or equilibrium analysis, applied to a 'system of change' (e.g. a set of difference or differential equations, a recurrence relation or iteration) provides insight into certain aspects of that system's behavior. For instance, it may tell whether the system is stable, where its solutions will tend to, etc.

ILC deals with systems that are typically expressed as recurrence relations on some (infinite-dimensional) input space. In the previous chapter we emphasized the algorithmic aspects of such relations. That is to say, we viewed them as *computational procedures*. But when formulated as an equation to be solved for, a recurrence relation turns into a *model*. This model implicitly defines a behavior. For an *explicit* description of the system's behavior, we would need to *solve* the reccurence equation with respect to the particular variable of interest (in our case, the control input).

But that may not be necessary. For perhaps all we want to know is whether this solution, whatever it be, is 'acceptable'. This, at least, seems to be the general philosophy in ILC: people rarely compute complete solutions, even though, for first, and even higher order *linear* recurrences, it would not be all that hard. This suggests that *the focus of the synthesis problem is on the asymptotic behavior*, rather than the temporal behavior. In a later stage, this will explain why we insist on saying that two solutions converging to the same fixed point are equivalent, even as their temporal behavior is radically different.

3.2.1. Fixed point stability

Let F map U into U. Recall that any $u \in U$ satisfying

$$F(u) \;=\; u, \tag{3.1}$$

is a fixed point of F. The adjective 'fixed' indicates a situation of 'no change'. The rationale behind this terminology becomes clear upon consid-

ering the iteration

$$u_{k+1} = F(u_k). \tag{3.2}$$

The dynamics of this algorithm express that if, at some point in time, the sequence $\{u_0, u_1, \ldots\}$ reaches a fixed point (which it may never actually do), it is bound to remain there. Fixed points are important as they determine, in part or in full, how an algorithm behaves, asymptotically.

Fixed points may be classified as follows. Let $\bar{u} = F(\bar{u})$ be a fixed point of F; $\bar{u}$ is said to be *stable* if, for all $\varepsilon > 0$, there exists $\delta > 0$ such that $\|u_0 - \bar{u}\|_U < \delta$ implies $\|u_k - \bar{u}\|_U < \varepsilon$ for all $k \geq 0$. That is, if all trajectories starting in a small enough neighborhood of $\bar{u}$ remain in some specified neighborhood of $\bar{u}$. Moreover, $\bar{u}$ is said to be *asymptotically stable* if it is stable and, in addition, there exists $\delta > 0$ such that $\|u_0 - \bar{u}\|_U < \delta$ implies $u_k \to \bar{u}$ as $k \to \infty$. Finally, $\bar{u}$ is said to be *globally asymptotically stable* if asymptotic stability is independent of the initial condition u_0, i.e. if $\bar{u}$ is stable and, in addition, $\lim_{k\to\infty} u_k = \bar{u}$ for all $u_0 \in U$. In the context of linear systems, global asymptotic stability and asymptotic stability are equivalent.

3.2.2. Fixed points and the problem of ILC

Let us recursively define $F^k(u)$ as $F^k(u) := F(F^{k-1}(u))$, $F^0(u) := u$. In Section 2.3 we posed the problem of ILC as that of finding an iteration $F : U \mapsto U, u_{k+1} = F(u_k)$ such that certain properties hold. Let us review these properties. First of all, we would like the iteration to *converge*, which implies that F must have *at least one* asymptotically stable fixed point. Secondly, we want the asymptotic behavior to be independent of the choice of initial input, which implies that F can have *only one* fixed point, which must therefore be globally asymptotically stable. Combining the two properties, we may state our first requirement on F as follows:

(R1) There must exist $\bar{u} \in U$ such that for all $u_0 \in U$ we have

$$\lim_{k\to\infty} F^k(u_0) \;=\; \bar{u}. \tag{3.3}$$

But this is not all. We also require that the sequence $\{Pu_k\}$ converge to a neighborhood of some desired output y_d. Let us define $\mathcal{P}(y_\mathrm{d}) \subset U$ as the set of all $u \in U$ that satisfy this performance criterion, e.g.

$$\mathcal{P}(y_\mathrm{d}; \varepsilon) \;=\; \{u \in U : \|y_\mathrm{d} - Pu\| \leq \varepsilon\}. \tag{3.4}$$

The second requirement on F is thus given as:

(R2) $\bar{u}$ must be contained in $\mathcal{P}(y_\mathrm{d})$.

In the two-parameter synthesis problem, we parameterize F as $F(Q, L)$ and ask: how to choose Q, L such that (in accordance with **R1** and **R2**) the equation

$$u \;=\; F(Q,L)u \tag{3.5}$$

has a unique solution $\bar{u} \in \mathcal{P}(y_\mathrm{d})$? In *Standard ILC*, we define $F(Q, L)$ as

$$F(Q,L)u \;:=\; (Q - LP)\,u + Ly_\mathrm{d}. \tag{3.6}$$

To determine the fixed points of the map $F(Q, L)$, we solve the fixed point equation

$$\begin{aligned} u &= (Q - LP)\,u + Ly_\mathrm{d} \\ &\Leftrightarrow \\ (I - Q + LP)\,u &= Ly_\mathrm{d}. \end{aligned} \tag{3.7}$$

Depending on Q and L, the above equation may have one or more bounded solutions, or none whatsoever. We must make sure that it has one and only one for every y_d. This is surely the case if $(I - Q + LP)$ is boundedly invertible on U (its inverse exists and is bounded), in which case we have

$$\bar{u} \;=\; (I - Q + LP)^{-1}\,Ly_\mathrm{d}. \tag{3.8}$$

Our synthesis objective is to tune Q and L so as to position the fixed point at an optimal location, e.g. to minimize the asymptotic error. To that end, we take the set of all pairs $(Q, L) \in \mathcal{L}(U) \times \mathcal{L}(Y, U)$, and divide it into a set of pairs that do satisfy **R1** (plus some additional constraint, yet to be discussed) and a set of pairs that do not. The pairs that do satisfy **R1**, we call *admissible*. Naturally, we restrict attention to these admissible pairs. As a next step we recognize that different admissible pairs may define the same fixed point. This redundancy is removed by introducing an equivalence relation on the set of admissible pairs. Among equivalent pairs, we appoint representatives, and we redefine the synthesis problem in terms of these representatives. In short, that is what the rest of this chapter is about.

3.3. The set of admissible pairs

The controller synthesis literature distinguishes between controllers that internally stabilize, and those that do not. Likewise, in ILC, we can distinguish between certain parameter combinations that are 'good', and others

that are 'bad'. In this section we define what is 'good' and what is 'bad'. To that end, we perturb the class of CCF-ILC iterations, previously defined in Section (2.3.1)

$$u^w_{k+1} \;=\; Qu^w_k + Le^w_k + Ce^w_{k+1} + w_k \qquad (k = 0, 1, \ldots). \tag{3.9}$$

We use the superscript 'w' in u^w_k and e^w_k to indicate the effect of the perturbation $\{w_k\}$ on the solution $\{u^w_k\}$. The unperturbed solution (corresponding to $w \equiv 0$) is simply denoted as $\{u_k\}$. The standard assumptions on Q, L, C and P apply (see Section 2.3 for details).

Let us note that the initial input u^w_0 is constrained by the feedback term Ce^w_0; For this reason it cannot be considered an initial condition (as it can not be chosen freely). In its stead, we introduce the auxilary variable $\hat{u}_0$ (see Figure 3.1). Note that setting $\hat{u}_0$ suffices to initialize (3.9). The variable $\hat{u}_0$ will be our 'true' initial condition.

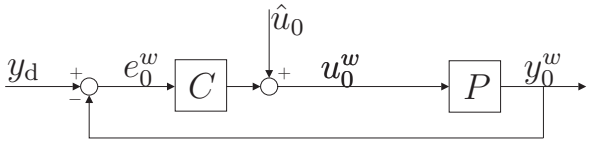


Figure 3.1.: Setting the initial condition in current cycle feedback ILC.

Definition 3.3.1 (Admissible pair). *Consider the class of perturbed iterations (3.9). Assume $u^w_0 = u_0$. We say that the pair $(Q, L) \in \mathcal{RH}^2_\infty$ is admissible (with respect to the said class of iterations) if:*

(C1) *For every $y_d \in \mathcal{H}_2$, there exists $\bar{u}, \bar{y} \in \mathcal{H}_2$, depending on y_d, but not on $\hat{u}_0$, such that for all $\hat{u}_0$, $\lim_{k\to\infty} u_k = \bar{u}$, and $\lim_{k\to\infty} y_k = \bar{y}$;*

(C2) *For every $\varepsilon > 0$ there exists $\delta > 0$ s.t. $\sup_k \|w_k\| < \delta$ implies $\sup_k \|u_k - u^w_k\| < \varepsilon$.*

The set of all admissible pairs is denoted as $\mathcal{A}$.

Condition **C1** is basically a restatement of Requirement **R1** (Section 3.2.2). It says that the respective parameters should be chosen such that the sequence of IO pairs (u_k, y_k) converges to some limit point $(\bar{u}, \bar{y}) \in \mathcal{H}^2_2$. Condition **C2** imposes the additional constraint that the perturbed solution u^w_k must at all times be contained in a neigborhood of the unperturbed solution and should approach the latter continuously when the pertubation terms subsides.

The notion of admissibility is not to be confused with that of 'internal stability for ILC', see (Goldsmith, 2002, Definition 1). In the spirit of that definition (which does not entirely match our terminology), the iteration (3.9) is said to be *internally stable* if both y_k and u_k are bounded on each trial $k < \infty$ in the presence of bounded disturbances and noise. In our setup, internal stability for ILC is an immediate consequence of the assumption that C is internally stabilizing and Q, L stable. Internal stability for ILC guarantees well-posedness (see Lemma 2.3.1), but the notion of admissibility is stronger: starting from a family of well-posed (internally stable) iterations, operators (and the corresponding family members) are classified into one of two subsets, depending on whether they have or lack certain properties.

3.4. Standard ILC and Equivalent Feedback

In this section we apply the ideas outlined in Section 3.2 to the problem of Standard ILC. We derive a necessary and sufficient condition for the pair (Q, L) to be admissible. Then we show that the synthesis problem is overparameterized, and we define an equivalence relation so as to reduce the 2-parameter problem to a 1-parameter problem. Along the way we introduce the concept of equivalent feedback, which, as it turns out, puts all things in a whole new perspective.

3.4.1. Necessary and sufficient conditions for admissibility

Consider the family of iterations (3.9). In accordance with the problem definition of Standard ILC in Section 2.3.2, we assume $P \in \mathcal{RH}_\infty$ and $C = 0$,

$$u^w_{k+1} \;=\; Qu^w_k + Le^w_k + w_k. \tag{3.10}$$

As there is no direct feedback term ($C = 0$), it follows that $\hat{u}_0 = u_0$ (see Figure 3.1).

Lemma 3.4.1. *Given the triple $(P, Q, L) \in \mathcal{RH}^3_\infty$ and suppose $C = 0$. The pair (Q, L) is admissible in the sense of Definition 3.3.1 if and only if $\|Q - LP\|_\infty < 1$.*

Proof. The proof of sufficiency follows standard contraction arguments and can also be found in e.g. Moore et al. (1992); Padieu and Su (1990); Norrlöf

(2000). We rewrite (3.10) to obtain

$$u^w_{k+1} = \underbrace{(Q - LP)}_{F} u^w_k + \underbrace{Ly_{\mathrm{d}}}_{d} + w_k. \tag{3.11}$$

By introducing the shorthand notation $F := (Q - LP)$, $d = Ly_{\mathrm{d}}$ we compact Equation (3.11) to the familiar form (compare (2.24))

$$u^w_{k+1} = Fu^w_k + d + w_k. \tag{3.12}$$

Repeated substitution gives

$$\begin{aligned} u^w_k &= F^k u^w_0 + \sum_{j=0}^{k-1} F^j \left(d + w_{k-1-j}\right) \\ &= F^k u^w_0 + \sum_{j=0}^{k-1} F^j w_{k-1-j} + (I - F)^{-1} \left(I - F^{k+1}\right) d. \end{aligned} \tag{3.13}$$

To prove convergence of the unperturbed solution (**C1**), we set w_j equal to zero. The result follows from Theorem 2.1.9, noting that $F : \mathcal{H}_2 \mapsto \mathcal{H}_2$, $z = Fw + d$, is a contraction. To verify continuity of the perturbed solution (**C2**), assume $\sup_k \|w_k\|_{\mathcal{H}_2} < \delta$, and observe that

$$\begin{aligned} \|u^w_k - u_k\|_{\mathcal{H}_2} &= \left\| \textstyle\sum_{j=0}^{k-1} F^j w_{k-1-j} \right\|_{\mathcal{H}_2} \leq \sum_{j=0}^{k-1} \|F\|^j_\infty \delta \\ &\leq \left(1 - \|F\|_\infty\right)^{-1} \delta. \end{aligned} \tag{3.14}$$

This shows that for any $\varepsilon > 0$, we can find $\delta > 0$ s.t. $\sup_k \|u^w_k - u_k\|_{\mathcal{H}_2} < \varepsilon$. This proves sufficiency.

Next we prove necessity. Suppose $\|F\| = 1$ (the case $\|F\| > 1$ is easy). We consider the SISO case. What we need to show is that for every $\delta > 0$ and every $\varepsilon > 0$ there exists $K > 0$ such that for at least some u_0 and some y_{d}, we can construct a sequence $\{w_j\}$ with $\|w_j\| < \delta$, $j \leq K - 1$, such that $\|u^w_K - u_K\| > \varepsilon$. Let $y_{\mathrm{d}} = u_0 = 0$, so that $u_k = 0$ for all k. Given K, we construct w such that

$$\|F^K w\| \geq \frac{1}{2}\|w\| \tag{3.15}$$

The fact that, for SISO systems, $\|F^K\| = \|F\|^K = 1$ implies that such w exists; this follows by continuity of the norm. Now define $w_k := F^k w$, $k = 0, 1, \ldots, K-1$. It follows that $u^w_k = kF^k w$ and by construction $\|u^w_K\| \geq \frac{1}{2}K\|w\|$. Choose w such that $\|w\| = \delta/2$ and take $K = \lceil 4\left(\varepsilon/\delta\right)\rceil + 1$. Then $\|u^w_K - u_K\| = \|u^w_K\| > \varepsilon$. This concludes the proof. ∎

Though the definition of admissibility appears rather involved, checking whether a given pair is admissible or not turns out to be relatively easy. This is of course no coincidence. We defined admissibility the way we did so as to arrive at a single condition that is both necessary and sufficient. That is not to say this definition is completely arbitrary either. Condition **C1** is common. Were this the only condition, satisfying the inequality $\|Q - LP\|_\infty < 1$ would still be sufficient, but no longer necessary (we illustrate this in the next example). This means that we would need to develop more elaborate tests to be able to determine whether a given pair of operators is admissible or not. Indeed, it appears that without **C2**, there is no single condition for admissibility that is both sufficient and necessary. In addition, we may point out that **C2** ensures a primitive, but not altogether artificial form of robustness.

Example 3.4.2. *This example intends to show that, in case we leave out* **C2**, *the condition* $\|Q - LP\| < 1$ *(see Lemma 3.4.1) is no longer necessary for admissibility.*

Consider the iteration

$$u_{k+1} \;=\; Fu_k + d. \tag{3.16}$$

Suppose $F \in \mathcal{RH}_\infty$ *and* $d \in \mathcal{H}_2$. *By Theorem 2.1.9 we know that if* $\|F\|_\infty < 1$, *then* $\{u_k\}$ *tends to a limit point* $\bar{u}(d) \in \mathcal{H}_2$, *independent of* u_0. *We also know that if* $\|F\|_\infty > 1$, *then there exists* u_0 *such that* $\{u_k\}$ *does not have a limit. But what if* $\|F\|_\infty = 1$*? Let* F *be given as*

$$F(s;\gamma) \;=\; \frac{\gamma s}{(s+1)^3}. \tag{3.17}$$

We select $\gamma = \frac{3}{2}\sqrt{3}$ *so that* $\|F\|_\infty = 1$. *The real-valued function* $|F(j\omega)|$ *attains its maximum value* $(= 1)$ *at* $\omega = \pm\frac{1}{2}\sqrt{2}$. *For all other* ω *(including the point at infinity),* $|F(j\omega)|$ *is less than unity.*

For $\{u_k\}$ *to converge for all* $d \in \mathcal{H}_2$, *the map* $(I - F)$ *must be boundedly invertible on* $\mathcal{RH}_\infty$ *(see also Example 2.1.3). We will show that this is indeed the case. First, let us establish that* $1 - F(s)$ *does not have any zeros in the open right half plane (RHP). Suppose on the contrary that* $1 - F(s_0) = 0$ *for some* s_0 *with* $Re(s_0) > 0$. *This implies* $|F(s_0)| = 1$. *As* $\sup_{Re(s)>0} |F(s)| = 1$ *it follows, by analyticity of* F *and the maximum modulus principle, that* $F(\cdot)$ *is a constant map. But this is clearly not true; hence we conclude that* $F(s)$ *does not have any zeros in the open RHP. But how about zeros on the imaginary axis? There are none, since* $|F(j\omega)| < 1$ *for all* $\omega \in \mathbb{R}$ *except at* $\omega = \pm\frac{1}{2}\sqrt{2}$, *and inspection shows that*

$$F(\pm\frac{1}{2}\sqrt{2}j) \;=\; \frac{5}{9}\sqrt{3} \mp \frac{1}{9}\sqrt{2}\sqrt{3}j \quad (\neq 1). \tag{3.18}$$

Note furthermore that $F(\infty) = 0$, which implies that $1 - F(s)$ is biproper. We conclude that $(1 - F(s))^{-1}$ is a proper real-rational transfer function with no poles in the closed RHP, i.e. $(1 - F)^{-1} \in \mathcal{RH}_\infty$.

This condition is obviously not sufficient*. Yet, in our example, the sequence $\{u_k\}$ does converge. To prove this, we show that the increment $(u_{k+1} - u_k)$ tends to zero and that $\{u_k\}$ is bounded. Define $\Delta u_k := u_{k+1} - u_k$. By linearity, we have that*

$$\Delta u_k \;=\; F\Delta u_{k-1}. \tag{3.19}$$

Observe that $\{\|\Delta u_k\|\}$ is a nonincreasing sequence, bounded from below, and hence must have a limit point. This implies that there exists $w \in \mathcal{H}_2$ such that $\|w\| = \|Fw\|$. Note that, for any $\varepsilon > 0$, there exists $\delta > 0$ such that

$$\begin{aligned} \|Fw\|^2/2 \;&\leq\; \int_0^{\frac{1}{2}\sqrt{2}-\delta} |F(j\omega)|^2\, |w(j\omega)|^2 d\omega \cdots \\ &\quad + \int_{\frac{1}{2}\sqrt{2}+\delta}^{\infty} |F(j\omega)|^2\, |w(j\omega)|^2 d\omega + \varepsilon \end{aligned} \tag{3.20}$$

Assume $w \neq 0$. This assumption, combined with the fact that $|F(j\omega)| < 1$ for all $\omega \neq \frac{1}{2}\sqrt{2}$ implies that there exists $\bar{\delta} > 0$ and $\gamma > 0$, such that whenever $\delta < \bar{\delta}$,

$$\begin{aligned} \int_0^{\frac{1}{2}\sqrt{2}-\delta} |F(j\omega)|^2\, |w(j\omega)|^2 d\omega + \int_{\frac{1}{2}\sqrt{2}+\delta}^{\infty} |F(j\omega)|^2\, |w(j\omega)|^2 d\omega \quad \cdots \\ \leq \quad \int_0^{\frac{1}{2}\sqrt{2}-\delta} |w(j\omega)|^2 d\omega + \int_{\frac{1}{2}\sqrt{2}+\delta}^{\infty} |w(j\omega)|^2 d\omega - \gamma. \end{aligned} \tag{3.21}$$

This in turn implies that

$$\begin{aligned} \|Fw\|^2/2 \;&\leq\; \int_0^{\frac{1}{2}\sqrt{2}-\delta} |w(j\omega)|^2 d\omega + \int_{\frac{1}{2}\sqrt{2}+\delta}^{\infty} |w(j\omega)|^2 d\omega - \gamma + \varepsilon \\ &\leq\; \|w\|^2/2 - \gamma + \varepsilon. \end{aligned} \tag{3.22}$$

Note that γ is fixed and does not depend on δ; The parameter ε on the other hand, can be made arbitrarily small. Indeed for any ε with $0 < \varepsilon < \gamma$, there exists δ with $0 < \delta < \bar{\delta}$ such that (3.20) holds. It follows that $\|Fw\| < \|w\|$. But this contradicts our starting assumption. We conclude that $\|w\| =: \lim_{k\to\infty} \|\Delta u_k\| = 0$ (and this in turn implies that $\|u_{k+1}\| - \|u_k\| \to 0$).

Next we show that the sequence $\{\|u_k\|\}$ is bounded. To that end, we consider the solution in explicit form

$$u_k \;=\; F^k u_0 + (I - F)^{-1} \left(I + F^k\right) d. \tag{3.23}$$

The right hand side may be bounded as follows

$$\begin{aligned}\left\|F^k u_0 + (I-F)^{-1}\left(I+F^k\right)d\right\| &\leq \left\|F\right\|_\infty^k \|u_0\| \cdots \\ &\quad + \left\|(I-F)^{-1}\right\|_\infty \left(1+\|F\|^k\right)\|d\| \\ &= \|u_0\| + 2\left\|(I-F)^{-1}\right\|_\infty \|d\|,\end{aligned}$$

and this proves that $\sup_k \|u_k\| < \infty$*. We conclude that there exists* $l \in \mathbb{R}$ *such that* $\lim_{k\to\infty} \|u_k\| = l$*. Given that* $\lim_{k\to\infty} \Delta u_k = 0$*, this implies that there exists* $\bar{u} \in \mathcal{H}_2$ *such that* $\lim_{k\to\infty} u_k = \bar{u}$*. This concludes the example.*

3.4.2. Equivalent admissible pairs

We introduce some additional notation and terminology. Let $\mathcal{S}$ denote a set and let $\mathcal{R}$ be a *relation* on $\mathcal{S}$, i.e. a collection of ordered pairs of elements of $\mathcal{S}$. Suppose $\mathcal{R}$ satisfies the following properties

1. $(a,a) \in \mathcal{R}\ \forall\ a \in \mathcal{S}$.
2. $(a,b) \in \mathcal{R} \Rightarrow (b,a) \in \mathcal{R}\ \forall\ a,b \in \mathcal{S}$.
3. $(a,b),(b,c) \in \mathcal{R} \Rightarrow (a,c) \in \mathcal{R}\ \forall\ a,b,c \in \mathcal{S}$.

then $\mathcal{R}$ is an *equivalence relation*. In that case two elements $a, b \in \mathcal{S}$, $(a,b) \in \mathcal{R}$ are said to be equivalent and we write $a \simeq b$. For each $a \in \mathcal{S}$, we define the *equivalence class* $[a]$ containing a to be the set of all elements in $\mathcal{S}$ equivalent to a, i.e.

$$[a] \;=\; \{b \in \mathcal{S} \mid b \simeq a\}$$

It is easy to see that equivalence classes constitute a *partition* of $\mathcal{S}$ that is, a collection of nonempty disjoint subsets of $\mathcal{S}$ whose union is $\mathcal{S}$. In order to refer to a particular equivalence class, one element can be selected to represent the entire class. Such an element is called a *class representative*.

Let us see how we can apply this apparatus to our synthesis problem. Recall that our current objective is to select from among a family of iterations, a particular member, or set of members, that satisfy certain performance constraints (see our discussion in Section 3.2.2). As indicated before, we may restrict attention to those members corresponding to admissible pairs. It is obvious that different admissible pairs induce different input sequences. Even so, different sequences can converge to the same fixed point. Recall that if $(I-Q+LP)$ is boundedly invertible on $\mathcal{H}_2$, then the fixed point $\bar{u}$ is given as

$$\bar{u} \;=\; (I-Q+LP)^{-1}Ly_{\mathrm{d}} \;=\; \big(I+\underbrace{(I-Q)^{-1}L}_{K}P\big)^{-1}\underbrace{(I-Q)^{-1}L}_{K}y_{\mathrm{d}}.$$

(3.24)

For the second equality to hold we assume that the inverse $(I-Q)^{-1}$ is well-defined. Shortly, we will see that this assumption is justified. What Equation (3.24) shows is that the fixed point $\bar{u}$ is completely determined by the matrix fraction $K := (I-Q)^{-1} L$. Admissible pairs for which this fraction is equal are considered equivalent.

Definition 3.4.3 (Equivalence on $\mathcal{A}$). *Suppose $P \in \mathcal{RH}_\infty$ is strictly proper. Let $\mathcal{A}$ be the set of pairs admissible to the family of iterations (3.10). Two pairs $(Q_1, L_1), (Q_2, L_2) \in \mathcal{A}$ are said to be equivalent if they satisfy the equivalence relation*

$$(I-Q_1)^{-1} L_1 \;=\; (I-Q_2)^{-1} L_2.$$

Let us verify that the matrix fraction appearing in Definition 3.4.3 is well-defined. By well-defined we mean that $(I-Q)^{-1}$ exists (as a proper real-rational, but not necessarily stable transfer function) and that $(I-Q)$ and L are coprime (so that no *unstable* pole-zero cancellations can occur). The reason for insisting on coprimeness will become clear in Section 3.4. We have the following lemma.

Lemma 3.4.4. *Given $P \in \mathcal{RH}_\infty$. For any $(Q, L) \in \mathcal{A}$, the pair $(I-Q, L)$ is left-coprime. If P is strictly proper then $(I-Q)^{-1}$ exists and is proper.*

Proof. Recall that two transfer matrices $(I-Q)$ and L in $\mathcal{RH}_\infty$ are left-coprime over $\mathcal{RH}_\infty$ iff there exist matrices X and Y in $\mathcal{RH}_\infty$ such that

$$(I-Q)\,X + LY \;=\; I \tag{3.25}$$

Define $X = (I-Q+LP)^{-1}$, $Y = PX$. Note that $X \in \mathcal{RH}_\infty$ by assumption of admissibility ($\|Q-LP\|_\infty < 1$, see Lemma 3.4.1) and hence also $Y \in \mathcal{RH}_\infty$. Clearly $(I-Q)\,X + LY = I$, which proves left-coprimeness. To prove that $(I-Q)^{-1}$ defines a (proper) real-rational transfer function, it suffices to show that $(I-Q(\infty))^{-1}$ exists. We consider the bounded real-rational transfer function $(I-Q+LP)^{-1}$. Strict properness of P and boundedness of L implies that $L(\infty)P(\infty) = 0$. Thus, at infinity, the map $(I-Q+LP)^{-1}$ reduces to $(I-Q)^{-1}$. Since $(I-Q+LP)^{-1}$ is bounded at infinity, so is $(I-Q)^{-1}$. This completes the proof. ∎

Lemma 3.4.4 states that, provided the pair (Q, L) is admissible and P strictly proper, the matrix fraction $(I-Q)^{-1} L$ defines a proper, real-rational, but not necessarily stable transfer function. The condition on

strict properness of P is not essential. We could define an equivalence relation independent of this condition. For instance, in view of Equation (3.24), we could define two admissible pairs to be equivalent if

$$(I - Q_1 + L_1 P)^{-1} L_1 \;=\; (I - Q_2 + L_2 P)^{-1} L_2. \tag{3.26}$$

Yet we choose to adopt the given definition. For one, because of its compactness; but mainly because of its interpretation in terms of Equivalent Feedback, to be discussed after the next example.

Example 3.4.5. *The purpose of this example is to illustrate what it means for two pairs to be equivalent, as well as to motivative the condition of strict properness (Definition 3.4.3). Consider a plant P, given as*

$$P(s) \;=\; \frac{1}{2}\frac{s-2}{s+1}. \tag{3.27}$$

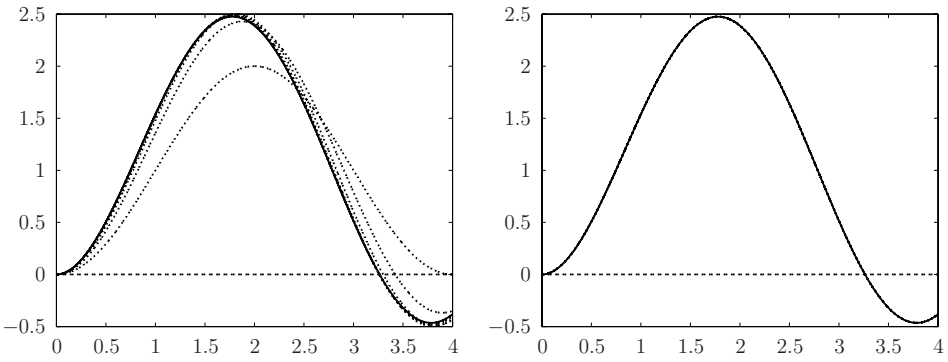
We pick Q and L such that $\|Q - LP\|_\infty < 1$, i.e. we make sure that (Q, L) is admissible (see Lemma 3.4.1). Take $L_1 = 1$; $Q_1 = (s-1)/(s+1)$ and verify that $(Q_1 - L_1 P) = s/(2(s+1))$ is a contraction. We have that

$$(I - Q_1)^{-1} L_1 \;=\; \frac{1}{2}(s+1). \tag{3.28}$$

Next we choose $L_2 = 2(s+1)/(s+2)$ and $Q_2 = (s-2)/(s+2)$. In this case $Q_2 - L_2 P = 0$, and

$$(I - Q_2)^{-1} L_2 \;=\; \frac{1}{2}(s+1). \tag{3.29}$$

We observe that the pairs (Q_1, L_1), (Q_2, L_2) are equivalent. Figure 3.2 shows how the respective iterations $\{u_{k+1} = Q_i u_k + L_i e_k : i = 1, 2\}$ converge to the same fixed point. In both cases $(I - Q)^{-1}$ is well-defined, be it not proper. For SISO systems, it is easy to see that the only case in which $(I - Q)^{-1}$ is not defined (as a rational transfer function) is when $Q = 1$. As regards this example, We may verify that there is no $L \in \mathcal{RH}_\infty$ such that $(1, L) \in \mathcal{A}$. In other words, for the given plant, the matrix fraction $(I - Q)^{-1} L$ would be well-defined for all admissible pairs. But in general, this need not be so. Consider for instance, the class of bistable plants, i.e. the set of plants P such that $P, P^{-1} \in \mathcal{RH}_\infty$. We pick $Q = I$ and $L = P^{-1}$ so that $Q - LP = 0$ and thus $(Q, L) \in \mathcal{A}$. In that case $(I - Q)^{-1}$ is clearly not defined. The fastest way around this problem is to insist on strict properness. The price we pay is that we exclude certain cases of interest, such as the one we discussed in this example.



(a) $u_k(t)$ for $t \in [0,4]$ and $k = 0 \ldots 9$; $Q = Q_1$ and $L = L_1$).

(b) $u_k(t)$ for $t \in [0,4]$ and $k = 0 \ldots 9$; $Q = Q_2$ and $L = L_2$).

Figure 3.2.: The idea of equivalent admissible pairs. The figures show the evolution of the iteration $u_{k+1} = Qu_k + Le_k$ ($u_0 \equiv 0$) for different parameter pairs (Q_1, L_1) and (Q_2, L_2). Both converge to the same fixed point. In the first case (left figure) it takes an infinite number of trials, in the second case (right figure) only one.

3.4.3. Equivalent Feedback

In a number of recent papers Goldsmith (2001, 2002); Verwoerd et al. (2002, 2003) the notion of Equivalent Feedback for ILC has been studied in some detail. The main thesis communicated in these papers is that (causal) ILC and conventional feedback control are essentially equivalent. This is best understood as follows. Consider a family of Standard ILC iterations, previously defined in (2.79),

$$u_{k+1} = Qu_k + Le_k.$$

We know that, provided the pair (Q, L) is admissible, the sequence of inputs $\{u_k\}$ and the sequence of outputs $\{y_k\}$ jointly converge to their respective limits $\bar{u}$ and $\bar{y}$. Define $\bar{e} := y_{\mathrm{d}} - \bar{y}$. It follows that $(\bar{u}, \bar{e})$ satisfies the next equality:

$$(I - Q)\,\bar{u} \;=\; L\bar{e}, \tag{3.30}$$

which we rewrite as

$$\bar{u} \;=\; (I - Q)^{-1} L\bar{e}. \tag{3.31}$$

Equation (3.31) expresses a relation between a control input $\bar{u}$ and a tracking error $\bar{e}$ and can thus be interpreted as a defining equation for the feedback controller $K = (I - Q)^{-1} L$ (see Figure 3.3). Note that this controller is defined in terms of the parameters Q and L only. Hence, its realization requires no prior knowledge (other than what is required for the design of the ILC parameters). What is more, in feedback interconnection with the plant P, this compensator would give out the exact same control input $\bar{u}$ (thus delivering the same performance), without the need to engage in a process of iteration. Goldsmith (2002).

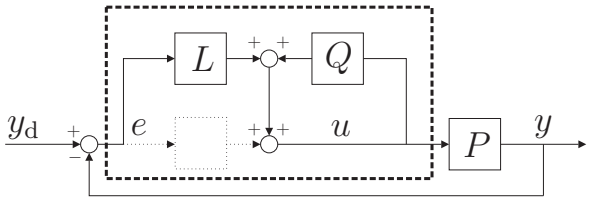


Figure 3.3.: Equivalent Feedback Controller (dashed) for Standard ILC.

Definition 3.4.6 (Equivalent Controller). *Let $P \in \mathcal{RH}_\infty$ be strictly proper. For every $(Q, L) \in \mathcal{A}$, we define the* Equivalent Controller *as*

$$K \;=\; (I - Q)^{-1} L. \tag{3.32}$$

Note that, by Lemma 3.4.4, the Equivalent Controller is well-defined as a proper real-rational transfer function with no unstable pole-zero cancellations. Not unexpected, the Equivalent Controller is internally stabilizing (see Section 2.3.1 for a definition of internal stability).

Theorem 3.4.7. *Let $P \in \mathcal{RH}_\infty$ be strictly proper. Then the associated Equivalent Controller (Definition 3.4.6, Figure 3.3) is internally stabilizing.*

Proof. Let $\tilde{V}^{-1}\tilde{U}$ and NM^{-1} be a left, and right-coprime factorization of the controller K and the plant P respectively. Then (Zhou et al., 1996, Lemma 5.1) says that K internally stabilizes P if and only if

$$\left(\tilde{V}M + \tilde{U}N\right)^{-1} \;\in\; \mathcal{RH}_\infty. \tag{3.33}$$

If we substitute $\tilde{V} = (I - Q), \tilde{U} = L, M = I, N = P$, Condition (3.33) evaluates to

$$(I - Q + LP)^{-1} \;\in\; \mathcal{RH}_\infty, \tag{3.34}$$

which is satisfied by assumption of admissibility (see Lemma 3.4.1). Note the we used the fact that for every $(Q, L) \in \mathcal{A}$, $(I-Q)$ and L are left-coprime (Lemma 3.4.4). ∎

As a kind of converse result, the following theorem says that *every* stabilizing controller admits a factorization in terms of admissible pairs.

Theorem 3.4.8. *Suppose $P \in \mathcal{RH}_\infty$ is strictly proper and let K be any stabilizing controller. Then there exists $(Q, L) \in \mathcal{A}$ such that $(I-Q)^{-1} L = K$.*

Proof. For any $F \in \mathcal{RH}_\infty$, the unique solution (Q_F, L_F) to the set of equations

$$\left.\begin{array}{rcl} Q_F - L_F P &=& F \\ L_F &=& (I - Q_F) K \end{array}\right\} \tag{3.35}$$

is given by

$$\left.\begin{array}{rcl} Q_F &=& I - (I-F)(I+KP)^{-1} \\ L_F &=& (I-F) K (I+PK)^{-1} \end{array}\right\} \tag{3.36}$$

By assumtion of internal stability, the closed-loop functions $(I+KP)^{-1}$ and $K(I+PK)^{-1}$ are both stable. From this we deduce that $Q_F, L_F \in \mathcal{RH}_\infty$. Now let F be a contraction, i.e. $\|F\|_\infty < 1$. Then, by construction, $(Q_F, L_F) \in \mathcal{A}$ (Lemma 3.4.1). ∎

The proof of Theorem 3.4.8 shows that there are infinitely many admissible pairs that define one and the same equivalent controller. By Definition 3.4.3, these are equivalent pairs. Thus we can interpret equivalent pairs as corresponding to different left-coprime factorizations of the same (equivalent) controller.

3.4.4. The synthesis problem revisited

Let us introduce the set $\mathcal{K}$ of all stabilizing controllers corresponding to a given plant P, along with the map $\phi : \mathcal{A} \mapsto \mathcal{K}$,

$$\phi(Q, L) \;:=\; (I-Q)^{-1} L \tag{3.37}$$

By Theorems 3.4.7 and 3.4.8 we know that ϕ is many-to-one and onto, i.e. surjective but not injective (see Figure 3.4). By considering equivalence classes instead, we can uniquely identify every element $K \in \mathcal{K}$ with a class

$$[(\hat{Q}, \hat{L})] \;=\; \left\{ (Q, L) \in \mathcal{A} : (Q, L) \simeq (\hat{Q}, \hat{L}) \right\}. \tag{3.38}$$

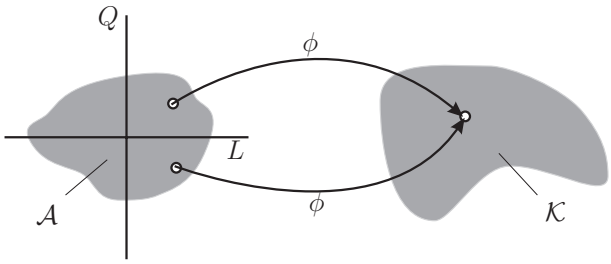


Figure 3.4.: The map $\phi : \mathcal{A} \mapsto \mathcal{K}$ is surjective but not injective.

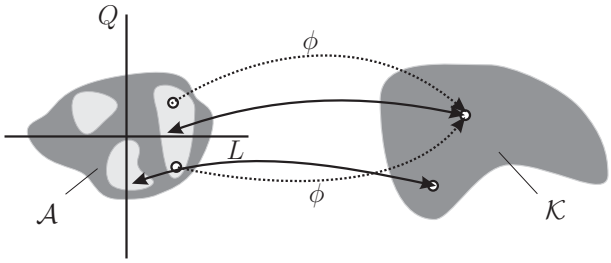


Figure 3.5.: Every equivalence class in $\mathcal{A}$ can be uniquely identified with a particular stabilizing controller.

This idea is illustrated in Figure 3.5.

We are associating *elements* in $\mathcal{K}$ with *sets* in $\mathcal{A}$. In order to create a map that is truly one-to-one, we restrict attention to class *representatives*. Two things are important to note here. First of all that by restricting attention to just one element (pair) in the class we do not loose generality. After all, all pairs within a class are equivalent, and by focusing on one element we effectively remove redundancy. Second that the choice of representatives is, in principle, arbitrary. We choose to select as a representative for a given class $[(Q, L)]$ the element $(Q_0, L_0) \in [(Q, L)]$ that satisfies the extra constraint $Q_0 - L_0 P = 0$. The following lemma says that every class has precisely one such member.

Lemma 3.4.9. *Let $(Q, L) \in \mathcal{A}$ be any admissible pair and let $[(Q, L)] \subset \mathcal{A}$ denote the corresponding equivalence class. There exists a unique admissible pair $(Q_0, L_0) \in [(Q, L)]$ such that $Q_0 - L_0 P = 0$.*

Proof. Immediate from the proof of Theorem 3.4.8 (take $F = 0$). ∎

In the Standard ILC problem, the quantity $Q - LP$ determines the *convergence speed* of the algorithm; the 'smaller' $Q - LP$ is, the faster convergence is achieved. Setting $Q_0 - L_0 P = 0$ implies that the recursion

defined in (2.79) converges to its fixed point after precisely one iteration. It is important to note that, although the location of the fixed point and the speed of convergence are both determined by the parameter pair (Q, L) they are nevertheless independent. That is to say, *we can reach any fixed point—within the reachable set induced by the admissible pairs—at any rate of convergence*. This is essentially what the proof of Theorem 3.4.8 tells us.

We define $\mathcal{A}_0$ to be the set of all class representatives or, equivalently, the set of all pairs $(Q, L) \in \mathcal{A}$ satisfying $Q - LP = 0$. Note that by definition every member of this set has the special form (LP, L). Note furthermore that $\mathcal{A}_0$ cannot contain any equivalent pairs since the representatives were taken from disjoint sets. As a consequence, the map $\phi_0 := \phi|_{\mathcal{A}_0}$, i.e. the restriction of ϕ to $\mathcal{A}_0$, is injective. Clearly ϕ_0 is also surjective and hence we conclude that $\phi_0 : \mathcal{A}_0 \mapsto \mathcal{K}$ is a *bijection*. In other words, the sets $\mathcal{A}_0$ and $\mathcal{K}$ are identical up to an isomorphism. This fact allows us to prove a well-known result.

Corollary 3.4.10 (Youla). *The set $\mathcal{K}$ of all (proper real-rational) controllers K stabilizing a (strictly proper) plant $P \in \mathcal{RH}_\infty$ can be parameterized as follows:*

$$\mathcal{K} \;=\; \left\{K : K = (I - LP)^{-1} L \;;\; L \in \mathcal{RH}_\infty\right\}$$

Proof. Note that the set of class representatives $\mathcal{A}_0$ has a trivial parameterization

$$\mathcal{A}_0 \;=\; \{(LP, L) : L \in \mathcal{RH}_\infty\}\,, \tag{3.39}$$

and recall that $\mathcal{A}_0$ and $\mathcal{K}$ are bijective under ϕ_0. Hence any given parameterization of $\mathcal{A}_0$ induces a parameterization of $\mathcal{K}$. In particular:

$$\mathcal{K} \;=\; \{\phi_0(LP, L) : L \in \mathcal{RH}_\infty\} \;=\; \left\{(I - LP)^{-1} L : L \in \mathcal{RH}_\infty\right\}$$

This completes the proof. ∎

Corollary 3.4.10 gives a parameterization of the set of all stabilizing controllers $\mathcal{K}$. It is easy to see that this parameterization is minimal. That is to say, there is no parameterization that uses fewer parameters. The corresponding block diagram is given in Figure 3.6.

The results in this section imply that the Standard ILC problem is equivalent to a compensator design problem for a stable plant. The compensator design problem has received a great deal of attention over the years and many of the results obtained (e.g. on robustness) can also be exploited

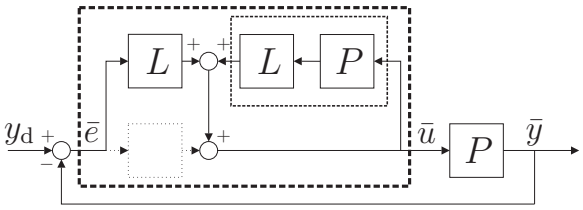


Figure 3.6.: Minimal parameterization of all stabilizing controllers. Shown is the Equivalent Controller (fat dashed box) under the constraint $Q = LP$ (thin dashed box). Compare with Figure 3.3.

in the context of ILC. We can simply apply any compensator design procedure and then calculate the corresponding admissible pair through the inverse map $\phi_0^{-1} : \mathcal{K} \mapsto \mathcal{A}_0$,

$$\begin{pmatrix} Q \\ L \end{pmatrix} = \begin{pmatrix} KP\,(I+KP)^{-1} \\ (I+KP)^{-1}\,K \end{pmatrix}. \tag{3.40}$$

For example, suppose we are to design an ILC scheme with small asymptotic error. In terms of compensator design this would translate into the problem of shaping the *output sensitivity function* $S = (I+PK)^{-1}$ such that $\bar{e} = Sy_d$ becomes small in an appropriate sense (e.g. within the bandwidth of interest). Using Theorem 3.4.10, it is easy to show that $\bar{e}$ can be expressed in terms of the free parameter $L \in \mathcal{RH}_\infty$ as follows

$$\bar{e} = (I - PL)\, y_d. \tag{3.41}$$

Designing for small $\bar{e}$ then translates into determining a good approximate inverse of P, see also Zames (1981) and Moore (1993). For the sake of the argument suppose P is a SISO minimum phase plant. We choose $L = ZP^{-1}$ (and $Q = Z$) where Z is typically designed to have lowpass characteristics in order to ensure properness of L without compromising the performance in the low frequency band. This coincides with the design procedures discussed in Moore (1993) and de Roover (1996). Having established this connection, one pressing question remains, that is: why bother to look at causal ILC at all? After all, it seems that there is no point in applying ILC if we can achieve the same performance more efficiently by implementing a feedback compensator. This point of view is particularly advocated by Goldsmith (2002). However, conclusive as the theory may be, without experimental practice to go on, definite conclusions cannot be drawn. If fact, when we set up an experiment, seeking to validation our analysis, the results we obtained

were not fully supportive. Though we found that the Equivalent Controller is indeed stabilizing (Theorem 3.4.7), we also found slight, but nevertheless significant, deviations in performance. More on these results in Section 3.6.

3.4.5. Causal ILC, Equivalent Feedback and Noise

As a footnote to the discussion on the implications of Equivalent Feedback, let us remark that our analyses did not account for the effect of possible (external) disturbances. This is important, for some of the conclusions obtained may not generalize to settings less ideal than ours. In particular, the very notion of Equivalent Feedback is based on the explicit assumption that all signals in the loop converge to a limit, which is not likely to happen in a setting with noise.

Our next discussion addresses the question as to what extent the Equivalent Controller is still 'equivalent' under various disturbances. We look at two cases: that of *input disturbances* (e.g. unmodelled dynamics and load disturbances), and that of *measurement noise* (e.g. quantization effects and sensor noise).

Adding measurement noise

Given some control input u_k, let $y_k = Pu_k$ denote the plant's *actual* output and let $\hat{y}_k$ denote the *measured* output. We assume that $\hat{y}_k = y_k + w_k$ where w_k represents the measurement noise. We consider the same update law as before, except that we now feed back the measured, instead of the actual output:

$$u_{k+1} \;:=\; Qu_k + L\,(y_\mathrm{d} - \hat{y}) \;=\; Qu_k + Le_k - Lw_k. \tag{3.42}$$

Here, e_k is defined in the usual way: $e_k := y_\mathrm{d} - y_k$. Assume that $\sup_i \|w_i\|$ is finite and let (Q, L) be admissible. Then $\{\|u_k\|\}$ is bounded. Note that the effect of the noise term w_k extends into the infinite future. That is to say, not only does it affect u_k, but also u_{k+1}, u_{k+2}, *et cetera*. As the update law is linear, we can easily isolate its (i.e. the noise term's) contribution. We set $y_\mathrm{d} = 0$ and solve (3.42) for u_k.

$$u_k \;=\; (Q - LP)^k\, u_0 + \underbrace{\sum_{i=0}^{k-1} (Q - LP)^i L w_i}_{u_k^w}. \tag{3.43}$$

The term on the right-hand side of (3.43), denoted by u_k^w, represents the total noise contribution to the system's input at trial k. Evidently, if w_i would be zero for all i, u_k^w would also be zero.

Now the question we ask is the following: Given a stabilizing controller K, how to choose Q and L so as to minimize the effect of noise *under the constraint that* $(I-Q)^{-1}L = K$? Based on Theorem 3.4.8 the same question may alternatively be put as: Given K, how to choose $F := Q - LP$ so as to make $\|u_k^w\|$ as small as possible?

It is obvious that for all constant (as in 'trial-independent') disturbances $w_i := w$, $i \geq 0$, the choice of F is immaterial. That is to say, as far as such disturbances are concerned, there is no advantage in applying Causal ILC instead of Equivalent Feedback Control. This is a direct consequence of the proof of Theorem 3.4.8 which says that for each $F \in \mathcal{RH}_\infty$ there is a unique pair $(Q_F, L_F) \in \mathcal{A}$ such that $Q_F - L_F P = F$ and $(I-Q_F)^{-1} L_F = K$, the operators Q_F and L_F being given as

$$Q_F = I - (I-F)(I+KP)^{-1}, \tag{3.44}$$

$$L_F = (I-F)K(I+PK)^{-1}. \tag{3.45}$$

Indeed, for any fixed disturbance w, the noise term u_k^w converges to a constant value $\bar{u}^w$, which is given as

$$\bar{u}^w = (I-Q+LP)^{-1}Lw = (I+KP)^{-1}Kw. \tag{3.46}$$

But what about non-constant, non-deterministic (stochastic) disturbances. Do the same conclusions apply? The answer is 'no'. Let us run a simulation on a standard second-order plant

$$P(s) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2}, \tag{3.47}$$

with parameter values $\zeta = 0.5$ and $\omega_0 = 10$. We select the (equivalent) controller K as

$$K(s) = \frac{4.00\text{e}2\ s^4 + 1.41\text{e}4\ s^3 + 2.50\text{e}5\ s^2 + 1.41\text{e}6\ s + 4.00\text{e}6}{2.50\text{e}1\ s^4 + 1.41\text{e}3\ s^3 + 3.00\text{e}4\ s^2 + 2.82\text{e}5\ s}. \tag{3.48}$$

For each k, let w_k be given as a zero-mean white noise signal with variance $\sigma_w^2 = 0.01$. We select $y_\mathrm{d} = 0, u_0 = 0$. For a number of values of F, we compute the variance of $y_N^w = Pu_N^w$, where N is a number depending on F. To be precise, we compute N as the smallest integer greater than $\log(F)/\log(\texttt{eps})$, where `eps` is the floating point relative accuracy (`eps` $\approx$ 1e-16). Striving for a fair comparison, this formula compensates for the difference in convergence speed. For $F = 0.25$ and $F = 0.50$, it respectively returns $N = 26$ and $N = 52$. Each experiment is repeated ten times, after

which we compute the average over all trials. Table 3.1 contains the results. For comparison, we computed the variance of the output in the equivalent feedback representation (also with $y_{\mathrm{d}} \equiv 0$) to $\mathtt{var}(y^w) = 6.9e-5$. The data in Table 3.1 suggests that as F tends to unity, $\mathtt{var}(y_N^w)$ tends to zero, while as F tends to zero, $\mathtt{var}(y_N^w)$ tends to a value which compares with that obtained with equivalent feedback. Our preliminary conclusion is that a slow rate of convergence is advantageous for suppression of measurement noise. It must be said however that we could also opt for fast convergence and conventional averaging over trials, in which case $\mathtt{var}(y_k^w)$ would decrease with $1/\sqrt{k}$.

F	0.25	0.50	0.63	0.71
N	26	52	78	104
$\mathtt{var}(y_N^w)$ (average)	4.3e-5	2.4e-5	2.0 e-5	1.3e-5

Table 3.1.: Simulation results for the case of measurement noise.

Input disturbance

Let u_k denote the *control* input and let d_k denote the input disturbance. We define the *actual* input u_k^d as the sum of u_k and d_k, i.e. $u_k^d := u_k + d_k$. We consider the same update law as before.

$$\begin{aligned} u_{k+1} &= Qu_k + L\,(y_{\mathrm{d}} - P\hat{u}_k) \\ &= Qu_k + Le_k - LPd_k. \end{aligned} \tag{3.49}$$

In first instance, let us assume that d_i is constant, i.e. $d_i = d$ for all i. Set $y_{\mathrm{d}} = 0$ and define $\bar{u} := \lim_{k\to\infty} u_k$, $\bar{u}^d := \lim_{k\to\infty} u_k^d$. It is easy to verify that $\bar{u}^d = \bar{u} + d$. From (3.49) we obtain the equality

$$\bar{u}^d = d - (I - Q + LP)^{-1}\,LPd = (I + KP)^{-1}\,d, \tag{3.50}$$

which again shows that the parameter F is immaterial when it comes to attenuating constant (trial-independent) disturbances.

To determine the effect of trial-dependent disturbances, we run the same simulation as before, but now with an input, instead of an output disturbance. We set $y_{\mathrm{d}} = 0, u_0 = 0$ and then we compute $y_N^d = Pu_N^d$ for different values of F, using the same formula. Table 3.2 shows the results. Let us remark that $\mathtt{var}(y_N^d)$ showed strong fluctuations. In other words, the averages listed give but a poor impression of the actual system behavior. But be it poor, it is an impression nevertheless. In the Equivalent Feedback

F	0.25	0.50	0.63	0.71
N	26	52	78	104
$\mathrm{var}(y_N^d)$ (average)	2.5e-5	1.9e-5	2.0e-5	1.9e-5

Table 3.2.: Simulation results for the case of input disturbance.

situation, the average variance of the output over ten trials was found to be 1.8e-6, which is significantly less than any of the values in the table.

The results furthermore suggest that, with respect to input disturbance, it does not matter all that much how we choose F. On the basis of these experiments we conclude that, as far as noise suppression is concerned, there is little reason to prefer causal ILC over conventional feedback control.

3.5. Extensions to CCF-ILC

This section deals with the CCF-ILC problem. Recall that CCF-ILC (Section 2.3.3) differs from Standard ILC (Sections 2.3.2 and 3.4) by the presence of the current cycle feedback term Ce_{k+1} in the iteration (2.74). Moreover, in contrast with the Standard ILC problem, the plant is not assumed to be stable and hence, for reasons of well-posedness, C is assumed to be stabilizing.

We would like to emphasize that the CCF-ILC problem is really a non-trivial generalization of the Standard ILC problem (rather: the latter is really a special case of the first) and thus deserves special treatment.

The following two problems are considered. Given a plant P along with a stabilizing controller C.

1. Let $(Q, L) \in \mathcal{A}$ be a given admissible pair for the family of iterations (3.9) and define the *equivalent controller* $K = (I - Q)^{-1}(L + C)$. Is K always stabilizing?

2. Given any stabilizing controller K. Does there always exist an admissible pair $(Q, L) \in \mathcal{A}$ for which K is an equivalent controller?

The respective answers to the above posed questions are 'Yes' and 'No'. We may note that in the context of the standard ILC problem both answers would have been affirmative (compare Theorems 3.4.7 and 3.4.8). As a matter of fact, the answer to the second question critically depends on C, and this dependence will be explored in depth. Both questions and their respective answers are of interest. The first because it tells us that

there always exists an equivalent stabilizing feedback controller. The second because it says that in case we do decide to apply ILC, the controller C should be considered a design parameter.

3.5.1. Equivalent feedback

The next lemma gives a necessary and sufficient condition for a pair (Q, L) to be admissible with respect to the class of iterations (3.9).

Lemma 3.5.1. *Consider the family of iterations (2.74). Let P be a real-rational transfer function and let C be a stabilizing controller for P. The pair (Q, L) is admissible if and only if*

$$\left\| (Q - LP)\,(I + CP)^{-1} \right\|_\infty \;<\; 1 \tag{3.51}$$

Proof. The proof runs along the same lines as that of Lemma 3.4.1. The idea is to consider the term $z_k := Qu_k^p + Le_k^p$, which can be viewed as 'the ILC contribution' to the feedback loop. If we can show boundedness of $\{z_k\}$ then, by internal stability, boundedness of $\{u_k\}$ and $\{y_k\}$ follows. Conversely, if we can construct $\{w_k\}$ and pick K, such that $\|z_K\|$ can be made arbitrarily large, it follows that either $\|u_K^p\|$ or $\|y_K^p\|$ can grow without bound, which would violate the condition for admissibility.

Define $F := (Q - LP)\,(I + CP)^{-1}$ and verify that z_k satisfies

$$z_{k+1} \;=\; Fz_k + Fw_k + (FC + L)\,y_{\mathrm{d}}. \tag{3.52}$$

Note that $(FC + L) \in \mathcal{RH}_\infty$. If we define $d := (FC + L)\,y_{\mathrm{d}}$ and $\tilde{w}_k := Fw_k$, Equation (3.52) simplifies to

$$z_{k+1} \;=\; Fz_k + \tilde{w}_k + d, \tag{3.53}$$

an expression resembling Eqn. (3.11). Let us observe that switching from w_k to $\tilde{w}_k$ does not affect the construction used in the proof for the Standard ILC case (Lemma 3.4.1). The rest of the proof would be a mere duplication and is thus omitted. ∎

The next theorem shows that the equivalent controller for CCF-ILC is internally stabilizing.

Theorem 3.5.2. *Given P strictly proper and let (Q, L) be an admissible pair. Then the equivalent controller $K = (I - Q)^{-1}\,(L + C)$ defines a proper real-rational matrix transfer function. Moreover, the corresponding feedback system (Figure 3.7) is well-posed and internally stable.*

Proof. First we prove properness of K. Admissability implies that a bounded $\bar{u}$ results for every $y_\mathrm{d} \in \mathcal{H}_2$. This is equivalent to saying that the input sensitivity matrix $U := (I - Q + (L + C)\,P)^{-1}\,(L + C)$ is stable, so that in particular, U is bounded at infinity. Strict properness of P implies that $U(\infty) = K(\infty)$ and hence K is also bounded at infinity (proper). To prove well-posedness, we need to show that $(I + K(\infty)P(\infty))$ is invertible, which is an immediate consequence of the above. To prove internal stability, consider the block diagram depicted in Figure 3.7. The dashed box represents the equivalent controller. The shaded box represents the ILC part of the overall system, which we denote by G_1. The remaining, non-ILC part of the system is denoted by G_2. The respective systems are given by

$$\begin{aligned} G_1 &= \begin{bmatrix} Q & L \end{bmatrix} \\ G_2 &= \begin{bmatrix} (I + CP)^{-1} \\ -P\,(I + CP)^{-1} \end{bmatrix} \end{aligned} \tag{3.54}$$

Note that G_1 and G_2 are both stable transfer matrices. The overall system can be represented as the feedback interconnection of the subsystems (see Figure 3.8). Under these conditions the overall system is internally stable if and only if (Zhou et al., 1996, Theorem 5.7)

$$(I - G_1 G_2)^{-1} \in \mathcal{RH}_\infty$$

where

$$(I - G_1 G_2)^{-1} = \left[I - (Q - LP)\,(I + CP)^{-1}\right]^{-1}$$

The above condition holds by assumption of admissability (is in fact a direct consequence of Lemma 3.5.1). This concludes the proof. ∎

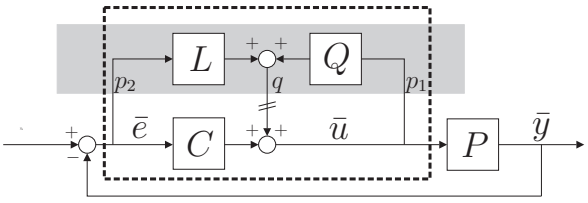


Figure 3.7.: Equivalent Feedback Controller (dashed) and the ILC-subsystem (shaded) for CCF-ILC.

As this settles the first question, we now turn to the second one: Is it true that, like in the case of Standard ILC, for every stabilizing controller $K \in \mathcal{K}$ we can find a corresponding admissible pair $(Q, L) \in \mathcal{A}$? The answer is no.

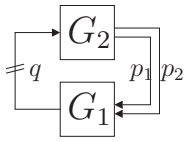


Figure 3.8.: The overall system as the interconnection of two stable subsystems G_1 and G_2.

Let $\mathcal{K}_{\mathcal{A}}$ denote the set of all equivalent controllers. We have the following theorem.

Theorem 3.5.3. *Let $P(s)$ be a proper real-rational plant, and let $C(s)$ be a proper real-rational controller such that the feedback interconnection of P and C is well-posed, and internally stable. Then the set $\mathcal{K}_{\mathcal{A}}$ of all equivalent controllers associated with the family of CCF-ILC iterations (2.74) satisfies the inclusion relation*

$$\mathcal{K}_{\mathcal{A}} \;\subseteq\; \mathcal{K}, \tag{3.55}$$

where equality (=) holds if and only if C is a strongly stabilizing (that is, C is both stable and stabilizing).

Proof. (if) The fact that $\mathcal{K}_{\mathcal{A}} \subseteq \mathcal{K}$ was established in Theorem 3.5.2. To prove: $\mathcal{K}_{\mathcal{A}} = \mathcal{K}$ iff $C \in \mathcal{RH}_\infty$. Suppose $C \in \mathcal{RH}_\infty$ and let K be any stabilizing controller. Define

$$\begin{aligned} Q &= (K-C)\,(I+PK)^{-1}\,P, \\ L &= (K-C)\,(I+PK)^{-1}\,. \end{aligned} \tag{3.56}$$

Clearly $Q, L \in \mathcal{RH}_\infty$. Moreover $(Q-LP)\,(I+CP)^{-1} = 0$, which is sufficient for admissability (see Lemma 3.5.1). This proves sufficiency. ∎

Proving necessity (only if) turns out to be harder. For that we need a few intermediate results, such as the following lemma.

Lemma 3.5.4. *Let $(Q, L) \in \mathcal{A}$ be admissible, and let $K := (I-Q)^{-1}\,L$ denote the associated Equivalent Controller. Then there exists $(Q_0, L_0) \in \mathcal{A}$ such that $(I-Q_0)^{-1}\,(L_0+C) = K$ and $(Q_0 - L_0P)\,(I+CP)^{-1} = 0$.*

Proof. It is clear that there exist $(Q_0, L_0) \in \mathcal{A}$ satisfying the given conditions if and only if the following set of equations has a solution:

$$\begin{aligned} (L+C) &= (I-Q)\,K, \\ (Q-LP)\,(I+CP)^{-1} &= 0. \end{aligned} \tag{3.57}$$

The unique solution to the above set of equations is given by

$$\begin{aligned} Q_0 &= (K-C)\,P\,(I+KP)^{-1}, \\ L_0 &= (K-C)\,(I+PK)^{-1}. \end{aligned} \tag{3.58}$$

From (3.57) we obtain

$$K - C = L + QK. \tag{3.59}$$

We substitute (3.59) into (3.58) and upon we inspection we conclude that $Q_0, L_0 \in \mathcal{RH}_\infty$. By construction, (Q_0, L_0) is admissible. This concludes the proof. ∎

What Lemma 3.5.4 says is that again, like in the case of Standard ILC, we can restrict attention to a smaller set of representative pairs without loosing generality. We exploit this fact to characterize the set $\mathcal{K}_\mathcal{A}$. Note that the set $\mathcal{A}_0$ of all $(Q, L) \in \mathcal{A}$ such that $(Q - LP)\,(I + CP)^{-1} = 0$ can be parameterized as follows

$$\mathcal{A}_0 = \{(Q, L) = (ZN, ZM); Z \in \mathcal{RH}_\infty\} \tag{3.60}$$

where $P = M^{-1}N$ is a left-coprime factorization over $\mathcal{RH}_\infty$. For stable P we can take $M = I$ and $N = P$. The corresponding parameterization coincides with the one we found in Section 3.4 for the case of Standard ILC (Equation 3.39). Through (3.60) we arrive at the following efficient parameterization of the set $\mathcal{K}_\mathcal{A}$.

Lemma 3.5.5. *Let $C = V^{-1}U$ and $P = M^{-1}N$ be any left-coprime factorization of the plant and the controller respectively. Then the set of all equivalent controllers $\mathcal{K}_\mathcal{A}$ is parameterized as*

$$\mathcal{K}_\mathcal{A} = \left\{K = (V - VZM)^{-1}\,(U + VZN)\,; Z \in \mathcal{RH}_\infty\right\} \tag{3.61}$$

Proof. The equivalent controller is defined as

$$K = (I - Q)^{-1}\,(L + C)$$

With $(Q, L) \in \mathcal{A}_0$, this expression evaluates to

$$\begin{aligned} K &= (I - ZM)^{-1}\left(V^{-1}U + ZN\right) \\ &= (V - VZM)^{-1}\,(U + VZN) \end{aligned} \tag{3.62}$$

This concludes the proof. ∎

Although the above parameterization seems to depend on specific factorizations, in actual fact the choice of coprime factors is immaterial. This is immediate from the fact that left-coprime factors are unique up to a left multiplication with a bistable transfer function (unit of $\mathcal{RH}_\infty$).

The following lemma restates an important result in control theory, namely the parameterization of all stabilizing controllers for a given plant P. This result is due (independently) to Youla and Kučera.

Lemma 3.5.6 (Youla-Kučera). *Given $C = V^{-1}U$ and $P = M^{-1}N$ with (U, V) and (M, N) left-coprime (over $\mathcal{RH}_\infty$. Assume C is stabilizing. Then the set $\mathcal{K}$ of* all *stabilizing controllers is given by (Zhou et al., 1996, Theorem 12.7)*

$$\mathcal{K} \;=\; \left\{ K = \left(V - \tilde{Z}M\right)^{-1} \left(U + \tilde{Z}N\right); \tilde{Z} \in \mathcal{RH}_\infty \right\} \tag{3.63}$$

If we compare the respective parameterizations of $\mathcal{K}_\mathcal{A}$ (Lemma 3.5.5) and $\mathcal{K}$ (Lemma 3.5.6) we see that they are equivalent if and only if for every $\tilde{Z} \in \mathcal{RH}_\infty$ there exists $Z \in \mathcal{RH}_\infty$ such that $\tilde{Z} = VZ$. The condition for equality is clearly satisfied in case C is strongly stabilizing ($C \in \mathcal{RH}_\infty$) since then V is bistable and Z can be taken to be $Z = V^{-1}\tilde{Z}$. This agrees with Theorem 3.5.3. We conclude this section with the remaining part of the proof of Theorem 3.5.3.

Proof. (of Theorem 3.5.3, only if) We need to show that $\mathcal{K}_\mathcal{A} = \mathcal{K}$ only if $C \in \mathcal{RH}_\infty$. Suppose $C \notin \mathcal{RH}_\infty$. Take $\tilde{Z} = I$ and let K be the corresponding stabilizing controller (Lemma 3.5.6). By uniqueness of the Youla parameter it is clear that the corresponding controller K belongs to $\mathcal{K}_\mathcal{A}$ if and only if there exist $Z \in \mathcal{RH}_\infty$ such that $VZ = I$. This however implies that V is bistable and $C = V^{-1}U$ stable, which contradicts our starting assumption. This concludes the proof. ∎

3.6. Experimental verification

3.6.1. The Experimental setup: the Linear Motor Motion System

We conducted some experiments on a linear motor system, the LiMMS (LInear Motor Motion System). A picture of the LiMMS is shown in Figure 3.9. The motor configuration consists of a base plate, covered with permanent magnets, and a sliding part (the *translator*), which holds the electric coils

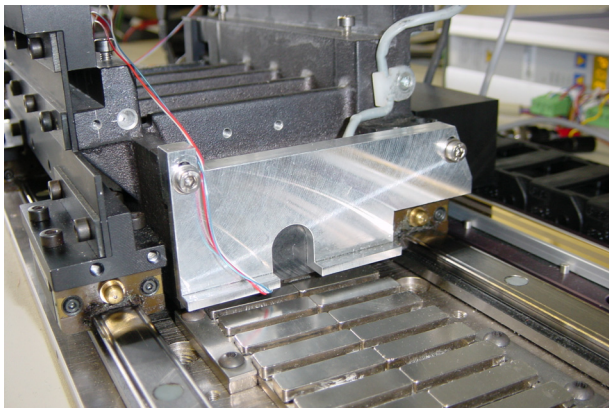


Figure 3.9.: The experimental setup: a magnetically actuated linear motor

and their iron cores. The principle of operation is a follows—see Velthuis (2000). A three-phase current generates a sequence of attracting and repelling forces between the magnetic poles of the coils in the translator and that of the permanent magnets in the base plate. The configuration is such that for all i, the phase of the current applied to coil i differs by $2\pi/3$ (modulo 2π) from that applied to coil $i+1$; this effects a net motion.

During operation, the LiMMS suffers from stiction, sliding friction, and *cogging*. Cogging is an effect that is common to the class of systems the LiMMS represents. It results from a strong magnetic interaction between the permanent magnets in the base plate and the iron cores of the coils of the translator. The cogging force effectively tries to align the magnets and the iron cores to stable positions of the translator.

The LiMMS has an effective range of about 0.6m. An optical (incremental) sensor with a resolution of about 1 μm measures the translator's relative displacement. 'Absolute' displacement is defined with respect to a fixed home position. As the LIMMS becomes operational, it goes through an initialization phase, in which the translator is first aligned with the magnets in the base plate, and then steered to its home position, where the encoder is reset.

In each experiment, the LiMMS was to track a second-order polynomial reference signal, as depicted in Figure 3.10. In standard operation mode, the system was controlled by a PD-type feedback controller,

$$C(s) \;=\; 177700\left(\frac{0.0084s+1}{0.00084s+1}\right). \tag{3.64}$$

For the design of the ILC scheme, we used the following, rudimentary, model

of the plant dynamics:

$$P(s) \;=\; \frac{1}{ms^2}. \tag{3.65}$$

We estimated the mass parameter to $m = 5$kg. Note that this model captures just the basic dynamic behavior (that of a moving mass).

3.6.2. Experiment Design

Let us detail the experiment objective. We consider a family of CCF-ILC iterations,

$$u_{k+1} \;=\; Qu_k + Le_k + Ce_{k+1}. \tag{3.66}$$

Our *hypothesis* is that to every causal iteration of the form (3.66), there corresponds an equivalent feedback controller

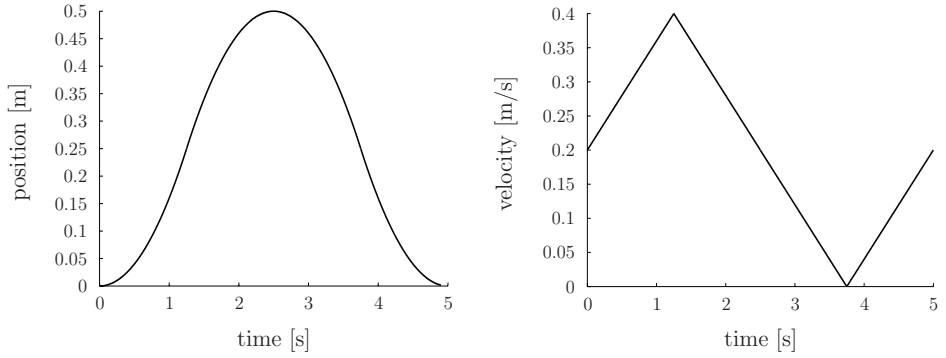
$$K \;:=\; (I - Q)^{-1}(L + C), \tag{3.67}$$

which gives the same performance (tracking error), without a single iteration. To test this hypothesis, we conduct a series of experiments. Each experiment consists of a sequence of steps, detailed below.

1. Pick some *admissible* pair (Q, L).
2. Run the iteration (3.66) until it has converged.
3. Record and store relevant data over a period of ten consecutive trials.
4. Compute the average over ten trials.
5. Switch to equivalent feedback (3.67).
6. Wait for transients to die out.
7. Record and store relevant data over a period of ten consecutive trials.
8. Compute the averages.
9. Compare the results of steps 4 and 8.

A few remarks are in order: (a) We select Q and L so that $L := QP^{-1}$, and

$$Q(s) := \frac{1}{(s\tau + 1)^2}. \tag{3.68}$$



(a) Position profile (b) Velocity profile

Figure 3.10.: The reference trajectory (left) that the system has to track. It has a period of 5s and a amplitude of 0.5m. The figure on the left shows position; The one on the right, velocity.

By construction $Q - LP = 0$, so that by Lemma 3.5.1, the pair (Q, L) is admissible. This guarantees that the design is at least *nominally* convergent. We are aware of other synthesis procedures wherein L is matched to the inverse of the closed-loop map $(I + PC)^{-1} P$, instead of to the inverse of P. These procedures would probably yield better performance. But given our objective, which is not to select the optimal parameters, but rather to validate our hypothesis, we want to keep things as simple as possible; (b) with respect to step 2, we remark that—by construction, see above—convergence is practically achieved within a few (even one or two) trials. Convergence is established by means of visual inspection; (c) the system is not actually reset after each trial. Instead, it is periodically excited by the same reference input. In the literature, this is known as *repetitive control*; (d) The reason for averaging over ten trials is to eliminate, or at least reduce the effect of random disturbances.

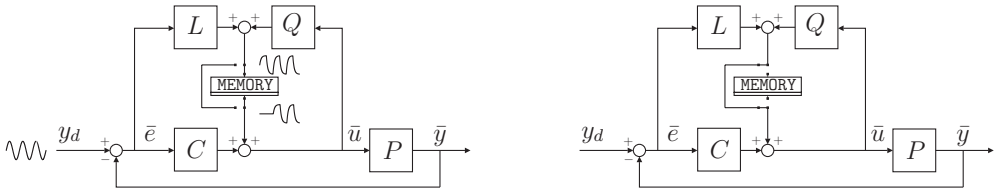
The results of the experiments are presented in Section 3.6.4. Coming up next is an overview of implementation details.

3.6.3. Implementation details

The control system was designed and implemented in Matlab Simulink. Using the real-time workshop we created targeted C-code which we downloaded on a DSP board inside a PC. The DSpace software that we used for interfacing, allowed us to change parameters and monitor the system's performance, both in real time.

Dynamic filters were implemented using continuous-time 'blocks' . This was done to match the 'analog' output of the DA convertors. The sample frequency was set to 1kHz. With a trial length of 5s, this amounts to 5000 samples per signal per trial. The buffer length of the memory block was chosen accordingly.

In order to allow for a fair comparison, and to enhance the transparancy of the design, we chose to match the structure of the Equivalent Controller to that of the ILC scheme. A transition from ILC to Equivalent Feedback was effected by shortcutting the memory block. The basic schematic is shown in Figure 3.11.



(a) ILC (b) Equivalent Feedback

Figure 3.11.: A schematic, showing the respective implementations of CCF-ILC (left) and the corresponding equivalent feedback controller (right).

In Figure 3.12 we show how switching from *nominal* feedback to ILC affects the system's performance. We observe that the tracking error is significantly reduced, from about 120μm (peak value) to about 40μm. Most apparent is the successful suppression of the inertial effect in the acceleration and deceleration phase. We also observe that the iteration practically converges within one trial, as expected.

3.6.4. Results

We now present the results of our experiments. We compared the *average* tracking error in ILC with the average tracking error obtained using Equivalent Feedback. The same experiment was repeated for several different values of τ (see 3.68). Figures 3.13 and 3.14 show the results for $\tau = 1/150$, and $\tau = 1/325$. Let us first establish the fact that, relative to the nominal feedback situation (without ILC or equivalent feedback) both methods greatly improve upon the tracking accuracy. This is important, since it is conceivable that for certain parameter combinations the

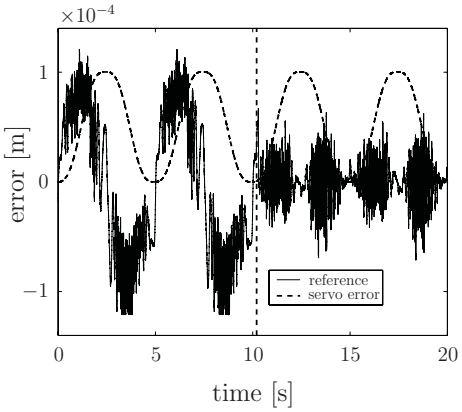
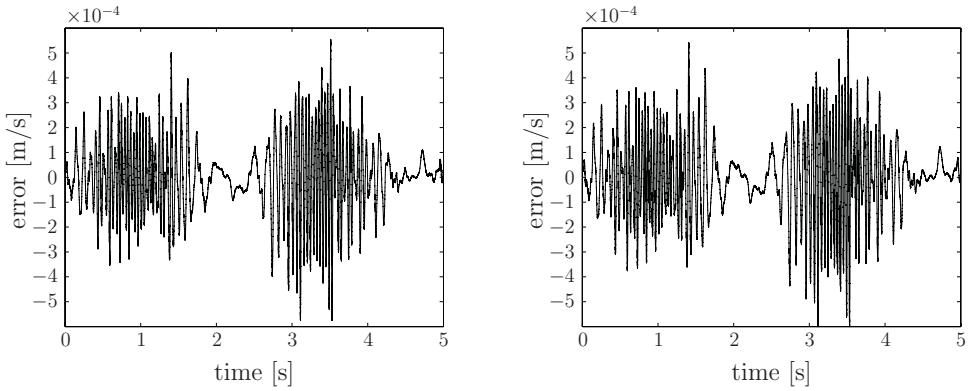
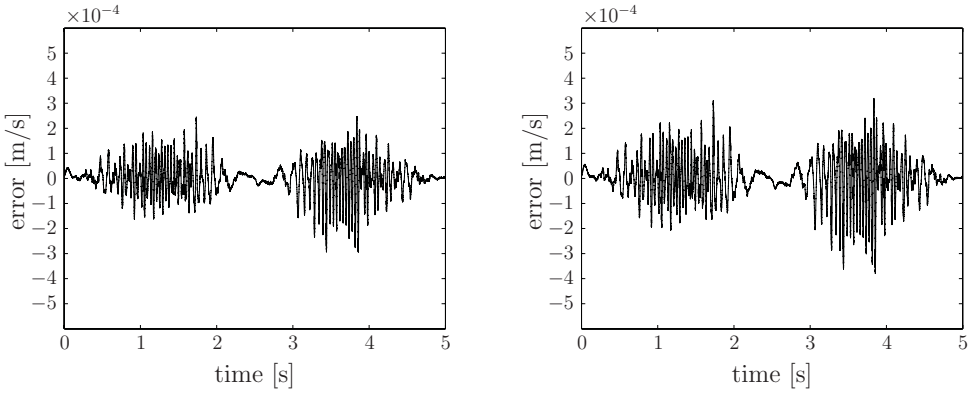


Figure 3.12.: The system under standard feedback without ILC (left of the vertical dashed line), and with ILC (right of the vertical dashed line). Shown also (not in scale) is the cycloidal reference input. Note that the system practically converges after one trial.



(a) ILC (b) Equivalent Feedback

Figure 3.13.: The average tracking error in ILC compared with that of the Equivalent Feedback configuration. The average was computed over ten consecutive trials (cycles). These results were obtained for $\tau = 1/150$.

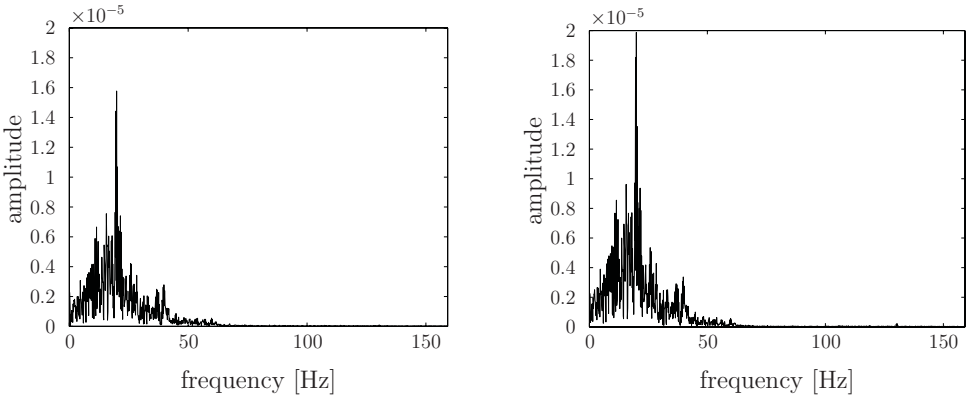


(a) ILC (b) Equivalent Feedback

Figure 3.14.: The average tracking error in ILC compared with that of the equivalent feedback configuration. The average was computed over ten consecutive trials (cycles). The scale is the same as in Figure 3.13. These results were obtained for $\tau = 1/325$.

impact of learning is negligable, especially so when the nominal PD controller is very well tuned. In that case there would be nothing to compare. Secondly, we observe that in both cases the Equivalent Controller is stabilizing—as expected, since the corresponding ILC scheme is convergent (see Theorem 3.5.2). At the same time, it is clear (especially from Figure 3.14) that the respective performances are not *identical*. The difference is small, but not negligible. Besides, it is persistent: we observed the same situation in all experiments. The general trend is that the difference increases with $1/\tau$. Closer inspection of the data shows that the cogging effect is reponsible for the larger part of the error. FFT analysis reveals peak activity at 24Hz (see Figure 3.15). When we compute the first derivative of the reference position with respect to time (see Figure 3.10b), we find that the maximum (reference) velocity is 0.40m/s. At the maximum the second derivative is obviously zero. During this time, the LiMMS operates at relatively constant speed. Given that the magnets in the base plate are situated $1.6 - 1.7$cm apart, we may indeed expect significant activity at and around $0.4/0.0165 \approx 24$Hz.

These results pose some serious questions. Do these findings invalidate our conclusions? That is hard to say. On the basis of Figure 3.11a, we would expect that *if* the iteration converges, then by definition, the signal going into the memory block would be identical to that going out. This suggests



(a) ILC (b) Equivalent Feedback

Figure 3.15.: FFT analysis on the tracking error in ILC (left) compared with that of the equivalent feedback configuration (right). The respective plots have the same scale. Note the peak at 24Hz. These results were obtained for $\tau = 1/325$.

that we can bypass all the iterations, by simply shortcutting the memory block (Figure 3.11b). But maybe this is not necessarily true. The reasoning is correct under the assumption that the iteration has a *unique* fixed point. But this we did not validate. Neither did we explicitly address the issue since in the context of linear systems, global convergence implies uniqueness. It is conceivable that the nonlinearities in our system allow for multiple solutions of the fixed point equation, in which case the feedback system depicted in (Figure 3.11) is not well-posed. However, in simulation studies (based on a more elaborate model of the plant) we could not confirm this conjecture. In all cases the outcome was the same: the respective performances were identical.

Another factor that could potentially explain the difference is noise (see Section 3.4.5). As indicated before, when noise enters the loop, signals no longer converge. The iteration still 'converges' to a certain neighborhood of the fixed point, but within this neighborhood it does not actually tend to a fixed point. Strictly speaking, the notion of equivalent feedback does not apply to this case. Still we would expect that *on average* (recall that we do average over ten trials), the solution of the iteration would tend to what we would obtain with Equivalent Feedback. But again, this may not be so. For it is not so clear how the noise propagates through the iterations. The results from Section 3.4.5 suggest that ILC may have some (positive) effect

on noise surpression, but the question is whether this effect can account for the difference observed. We ran some simulations to find out. We did detect differences in the respective performances. But unlike in the real setup, these differences were not significant.

A third factor, which is more of a guess again, is the software. We used a high-level language to specify our control scheme. The resulting model had to be compiled so as to create portable C-code that could be targeted to our DSP board. We do not know what code is actually exported to, and executed by the DSP board. Even if we assume that the software does what it is supposed to do, we still do not know if our code is what we want it to be. To put it differently: we can only hope that the high-level structural equivalence is reflected at the code level.

3.7. Discussion

We studied the synthesis problem associated with a class of causal iterations, and 'solved' it, to the extent that we showed that the problem can be recast as a compensator design problem. More accurately: we found that both methods are essentially equivalent.

Of all questions that remain, the most pressing, no doubt, is the following: why would would we design and implement an iterative scheme, knowing that, at no extra cost, the same performance can be obtained using a simple feedback controller? And in view of this, what good is our contribution?

As for the first question, we concur that there appears to be little reason to continue research on the type of causal iterations we considered, and indeed all the more reason to consider alternatives, which is what we will do in the next chapter. Yet, the final word has not been spoken. Our experimental results were not fully supportive and more work in this direction is to be conducted.

As for the second question, we believe our contribution to be valuable, if only because of the following:

- Our theory on admissible pairs and equivalent feedback casts new light on the compensator design problem, providing a new interpretation of classical results on the parameterization of stabilizing controllers, as contained in e.g. Zames (1981) and Youla et al. (1976)).

- By showing that the location of the fixed point and the speed of convergence to the fixed point are independent, we justified an assumption implicit in many existing synthesis procedures.

- Our approach using equivalent admissible pairs generalizes to families of iterations involving noncausal operators, in which case an interpretation in terms of Equivalent Feedback cannot be given.

3.7.1. Causal ILC and equivalent feedback

It is fair to say that the result on Equivalent Feedback, first appearing in Goldsmith (2001), took the community by surprise. On the one hand, this is not much of a surprise itself, as ILC was never meant to have a stong bearing on mainstream control methods. On the other hand, certain papers had been hinting at the same result, see e.g. Moore et al. (1992) and de Roover (1996).

In this respect, ILC research took an important turn when it adopted $\mathcal{H}_\infty$-techniques. For only then it became apparent what (causal) ILC was capable of, and, more importantly, what it was not. The explicit idea of Equivalent Feedback made it absolutely clear that the problem of (causal) ILC and that of conventional feedback are very much akin and confirmed what was already believed for long, namely that both methods subject to the same performance limitations.

3.7.2. Noncausal ILC

In a conventional feedback situation, the system's output is directly fed back into its input. By construction, this kind of 'closed-loop feedback' imposes a causality constraint on the feedback map. For if the current value of the input were to depend on future values of the output, we would have to assume the latter to be known ahead of time. Which, for a typical physical system, is not a realistic assumption. In ILC, the process of information feedback takes place offline, and may involve noncausal operations. In the next chapter we discuss how we can deploy such operations to overcome certain performance limitations.

4

Noncausal ILC

Overview – *We extend the two-parameter synthesis problem by allowing both operators to be noncausal. We discuss how the additional freedom may be deployed with particular emphasis on settings involving non-minimum phase behavior. We investigate the possibilities of Equivalent Feedback and show that the candidate equivalent controller is generally destabilizing.*

4.1. Introduction

Conventional feedback control studies feedback systems comprising a *plant* $P : U \mapsto Y$, $y(t) := (Pu)(t)$, and a *compensator* $C : Y \mapsto U$,

$$u(t) \;:=\; (Ce)(t), \tag{4.1}$$

where $e(t) := y_{\mathrm{d}}(t) - y(t)$. Typically, the plant is assumed causal, which is to say that for all t_0, the value of the output y at time $t = t_0$ is assumed not to depend on values of the input u at times $t > t_0$. In closed-loop feedback, the compensator is always causal, even in the unlikely case the plant is not. In ILC, this is not necessarily the case.

For we may recall that ILC is an *offline* method; an offline method that is perhaps best described as *open-loop feedback*. After all, notwithstanding its *open-loop* nature, ILC *is* a feedback method; It feeds back information in much the same way as conventional methods do (see Chapter 3 for details).

In any case, being an offline method, ILC allows for 'noncausal' operations, by means of which the 'future' of past trials can take part in the construction of the current control action.

In the present chapter we explain why this could be useful. We also return to our synthesis problem, this time without imposing any causality

constraints. The chapter's first objective is to justify the additional design freedom. As we look into plants with non-minimum phase behavior we will see that these systems alone provide ample motivation. More generally, we show that noncausal iterations can overcome various closed-loop related performance constraints, such as Bode's Sensitivity Integral, Poisson's Integral Formula, etc.

The chapter's second objective is to establish a result on Equivalent Feedback. We show that like in the case of causal ILC, an 'equivalent' (virtual) closed-loop feedback stucture may be defined. But it turns out that the candidate Equivalent Controller is generally destabilizing (when interpreted and implemented as a causal map). More precisely, we find that the Equivalent Controller is *non*causal which means that it cannot be implemented in closed loop.

The outline of the chapter is as follows. In Section 4.A we review some preliminaries on noncausal operators. This involves in particular a discussion on the effects of truncation, and some implementation issues. In Section 4.2 we show how noncausal operators may be employed in the context of the ILC synthesis problem. Section 4.3 covers the topic of Equivalent Feedback, and Section 4.4 concludes with a discussion of the main results.

4.2. Noncausal ILC

Let us turn to noncausal ILC. After introducing the basic concepts, we show why, in various situations, it may be advantageous to consider noncausal operators. For details concerning notation and terminology, we refer to Appendix 4.A.

4.2.1. Literature on Noncausal ILC

The idea of noncausal processing has been widely discussed in the ILC literature, under various different headings. It is there in Arimoto's original algorithm—Arimoto et al. (1984), see Section 2.2.1, whose derivative action assumes some preview (be it infinitesimally small) of the future. It is there in its many discrete-time extensions, where its derivative action is replaced by forward shifts. And it is there is many other papers, which we cannot even begin to list here.

In this thesis, we adopt a continuous-time perspective. Ignoring discretization effects, our framework provides us with the conceptual advantage of being able to distinguish between non*causal* and non*proper* operators. Arimoto's algorithm and most of its offspring is based on the use

of nonproper, rather than noncausal operators. Indeed, the main idea in these approaches is to introduce sufficient derivative action (enough forward shifts) to compensate for the system's relative degree, reducing the overall phase shift, thus making the system amenable to a form of high-gain feedback (infinite gain in fact, spread out over multiple trials) which forces the error down to machine precision.

Though powerful, the approach has a number of shortcomings, some of which cannot be resolved, some of which can be. As indicated in Section 2.2.2, for this approach to be succesful, we need to know the system's relative degree, and the sign of the first nonzero Markov parameter. Restrictive as this condition may be, in this context, there is little we can do about it. But there is another problem that we *can* do something about. And that has to do with the derivative action. It is well known that computing derivatives is a hazardous undertaking, especially when the source signal is contaminated with noise. This was also recognized some twenty years ago, shortly after the original algorithm was introduced. At the time, this led to an increase of interest in P-type learning rules and the like. Recall that D-type rules were known to converge under relatively mild conditions. As these P-type learning rule would not converge under equally mild conditions, many of them were equiped with a forgetting factor. This forgetting factor generally improved convergence, but at the same time compromised performance. In this chapter we will show that by using noncausal operators, even *bounded* noncausal operators, the aggressive D-action, characteristic of most Arimoto-type update laws, may be smoothed out over multiple trials, leading to a well-behaved algorithm, at no extra cost.

In our approach we focus on bounded noncausal, rather than (unbounded) nonproper operators (which we do not consider to be noncausal, or a limit case at best). We address several performance limitations inherent to causal update laws and show that most of these do not apply to noncausal update laws. Additionally, we show how to use noncausal operators in connection with non-minimum phase plants. In this respect, our approach bears on the work of Tomizuka (1987), whose Zero Phase Error Tracking Algorithm (ZPETK) is well-known in the general tracking literature.

4.2.2. Introduction: a motivation

To motivate the idea of using noncausal operators, let us go back to Section 2.2.2. In that section we discussed several perspectives on the problem of ILC. As we may recall, one of these perspectives was connected with the

problem of finding the minimizing argument of the functional

$$J(\Delta u;\gamma) \;=\; \left\| \begin{matrix} y_d - P\,(u+\Delta u) \\ \gamma \Delta u, \end{matrix} \right\|^2, \tag{4.2}$$

the solution of which led us to the update law (compare with (2.53))

$$u_{k+1} \;=\; u_k + \gamma^{-1}\left(\gamma^{-1}P + \gamma\,(P^*)^{-1}\right)^{-1} e_k. \tag{4.3}$$

To arrive at this expression, we had to assume that P is invertible (but not necessarily boundedly invertible, e.g. $P^{-1}(s)$ must exist, but need not define a bouded operator on $\mathcal{H}_2$). Update law (4.3) is of the form

$$u_{k+1} \;=\; u_k + Le_k, \tag{4.4}$$

with L defined as

$$L \;:=\; \gamma^{-1}\left(\gamma^{-1}P + \gamma\,(P^*)^{-1}\right)^{-1}. \tag{4.5}$$

If P is stable—which indeed we assume—then the adjoint operator $P^*(s) := P^T(-s)$ is antistable (i.e. has all its poles in the open RHP). The poles of L are the zeros of $(\gamma P(s)) + (\gamma P(-s))^{-1}$. It is immediate that if s_0 is a pole of L then so is $-s_0$. Hence, in general, L will have poles in both the left, and the right-half plane. This also shows in the next example.

Example 4.2.1. *Let $P(s)$ be given as*

$$P(s) \;=\; \frac{1}{s+1}. \tag{4.6}$$

We select $\gamma = 1$ and apply the update law (4.3) :

$$\begin{aligned} u_{k+1} &= u_k + \left(P(s) + (P(-s))^{-1}\right)^{-1} e_k \\ &= u_k + \left(\frac{1}{s+1} - s + 1\right)^{-1} e_k \\ &= u_k + \frac{-(s+1)}{(s+\sqrt{2})(s-\sqrt{2})} e_k \\ &= u_k + \frac{-\frac{1}{2}+\frac{1}{4}\sqrt{2}}{s+\sqrt{2}} e_k + \frac{-\frac{1}{2}-\frac{1}{4}\sqrt{2}}{s-\sqrt{2}} e_k \end{aligned} \tag{4.7}$$

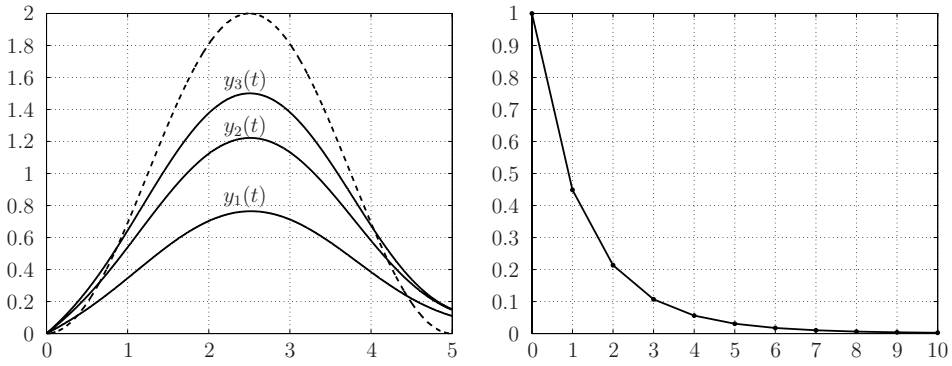
Here, we decomposed the noncausal operator

$$L(s) := \frac{-(s+1)}{(s+\sqrt{2})(s-\sqrt{2})} \tag{4.8}$$

into a causal part $L_c(s)$, and an anticausal part $L_{ac}(s)$ (both stable):

$$L_c(s) := \frac{-\frac{1}{2} + \frac{1}{4}\sqrt{2}}{s + \sqrt{2}}, \quad L_{ac}(s) := \frac{-\frac{1}{2} - \frac{1}{4}\sqrt{2}}{s - \sqrt{2}} \tag{4.9}$$

We select the desired output as $y_d(t) := 1 - \cos(2\pi t/T)$ with $T = 10$. For details concerning implementation we refer to Appendix 4.A. The initial input is set to zero, i.e. $u_0 \equiv 0$. Figure 4.1a shows the output $y_k(t)$ after the first three trials ($k = 1, 2, 3$). Figure 4.1b depicts the normalized mean squared tracking error during the first ten trials ($k = 1, \ldots, 10$). The error converges nicely, as does the input (not shown).

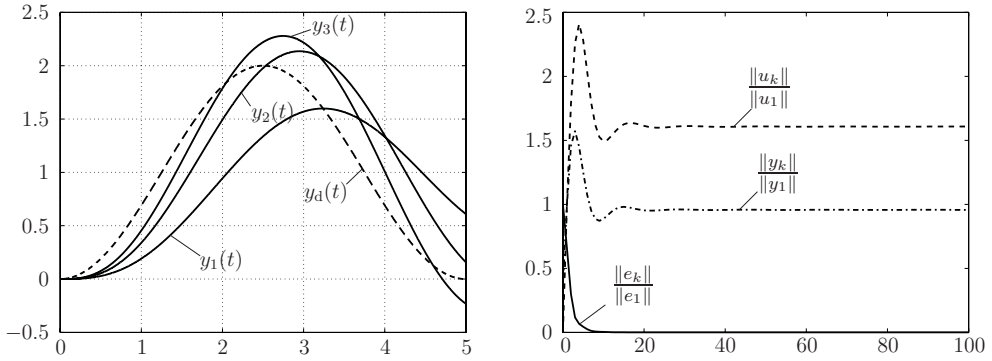


(a) $\{y_k(t) \ : \ k = 1, 2, 3\}$ (b) $\{\frac{\|e_k\|}{\|e_1\|} \ : \ k = 1, \ldots, 10\}$

Figure 4.1.: The evolution of the iterative system $u_{k+1} = u_k + Le_k$ with L as in (4.8) and P as in (4.6). The plot on the left shows the output y_k after the first three trials, along with the desired output y_d (dashed line). The plot on the right shows the normalized mean squared tracking error during the first ten trials.

Example 4.2.1 shows impressive performance. Yet, the same performance might have been obtained using just causal operators. One may recall Example 2.1.3. In that example, we asked the same system to track a discontinuous output so as to show that convergence of the output sequence generally does not imply convergence of the input sequence. But had we asked the system to track a different y_d (viz. the one in Example 4.2.1), both the output and the input sequence would have converged (see Figure 4.2). Conversely, had we replaced y_d in Example 4.2.1 by that of Example 2.1.3, the output (and thus the error) would still converge, but the input would

not. In other words, the results from this example are not conclusive, as the advantage of using noncausal iterations does not show. This advantage becomes more apparent when, in the next section, we consider plants with higher relative degree, or plants with non-minimum phase behavior.



(a) $\{y_k(t) : k = 1, 2, 3\}$ (b) $\{\frac{\|y_k\|}{\|y_1\|}, \frac{\|u_k\|}{\|u_1\|}, \frac{\|e_k\|}{\|e_1\|} : k = 0, \ldots, 100\}$

Figure 4.2.: The evolution of the iterative system $u_{k+1} = Qu_k + Le_k$ for $Q = L = 1$ and P as in (4.6). The plot on the left shows the output $y_k := Pu_k$ after the first three trials, along with the desired output y_d (dashed). The plot on the right shows the normalized mean squared input (dashed), output (dash-dot), and error (solid) for the first hundred trials.

4.2.3. Perfect Tracking and the Sensitivity Integral

What is the impact of noncausal operators on the system's performance? And what is the role of P in that? These are the questions we will be dealing with next.

Perfect tracking

Let us once again consider the family of one-parameter iterations,

$$u_{k+1} \;=\; u_k + Le_k, \tag{4.10}$$

with $L \in \mathcal{RL}_\infty$. We will henceforth assume that P is a SISO plant. The update law above induces a sequence of inputs $\{u_k\}$ and a sequence of

outputs $\{Pu_k\}$. We may verify that $y_k \to y_\mathrm{d}$ for all $y_\mathrm{d} \in \mathcal{L}_2(j\mathbb{R})$ as $k \to \infty$ if and only if

$$|1 - L(j\omega)P(j\omega)| \;<\; 1 \tag{4.11}$$

for *almost all* $\omega \in \mathbb{R}$ (that is: for all $\omega \in \mathbb{R}$ except on a set of measure zero). In the situation that y_k actually converges to y_d , we speak of *perfect tracking*. Whether in that case the input sequence also converges depends on y_d, as well as on the plant characteristics. More precisely, it depends on how well the desired output *matches* the plant characteristics. If we assume that the desired output is *feasible*, that is, if we assume that there exists $u_\mathrm{d} \in \mathcal{L}_2(j\mathbb{R})$ such that $y_\mathrm{d} = Pu_\mathrm{d}$, then the sequence of inputs necessarily converges to u_d. This follows from the fact that under the said condition the iteration

$$(u_\mathrm{d} - u_{k+1}) \;=\; (1 - LP)\,(u_\mathrm{d} - u_k) \tag{4.12}$$

converges to zero.

Causality constraints and pole-zero excess

In the light of our current discussion on the use of noncausal operators, an interesting question to ask is: does there always exist $L \in \mathcal{RH}_\infty$ (the space of *causal* operators) such that the $|1 - L(j\omega)P(j\omega)|$ is less than unity for almost all ω? And if not, how about if we extend the range of L to $\mathcal{RL}_\infty$? The following lemma provides a partial answer.

Lemma 4.2.2. *Let $P \in \mathcal{RH}_\infty$ be a plant having a pole-zero excess of at least two. Then for all $L \in \mathcal{RH}_\infty$ we have that $|1 - L(j\omega)P(j\omega)| \geq 1$ on a set with nonzero measure.*

Proof. Let us note that if $|1 - L(j\omega)P(j\omega)| < 1$ for almost all ω, then

$$\int_{-\infty}^{\infty} \log|1 - L(j\omega)P(j\omega)|\mathrm{d}\omega \;<\; 0. \tag{4.13}$$

To prove the statement, we show that under the conditions stated, (4.13) does not hold. The proof hinges upon a result by Poisson, which may be stated as follows. Let $F : \mathbb{C} \mapsto \mathbb{C}$ be a function, analytic in the closed right-half plane (RHP), such that

$$\lim_{R\to\infty} \frac{|F(Re^{j\theta})|}{R} \;=\; 0, \tag{4.14}$$

for all $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. Then the value of $F(s)$ at any point $s = x + jy$ in the open RHP may be recovered from $F(j\omega)$ by the integral relation

$$F(s) \;=\; \frac{1}{\pi}\int_{-\infty}^{\infty} F(j\omega)\frac{x}{x^2+(y-\omega)^2}\mathrm{d}\omega. \tag{4.15}$$

We apply this result to evaluate $\log|1-LP|$. Let us adopt the shorthand notation $V := 1 - LP$. We cancel the RHP zeros of V by introducing

$$\tilde{V} \;:=\; B_{\text{zeros}}^{-1}V, \tag{4.16}$$

where $B_{\text{zeros}}(s)$ is the *Blaschke product*

$$B_{\text{zeros}}(s) \;:=\; \prod_i \frac{z_i - s}{\bar{z}_i + s}, \tag{4.17}$$

formed from the open RHP zeros of V. Note that $\tilde{V}$ is analytic in the open RHP. For any open RHP point $s = x + jy$ we have

$$\log\tilde{V}(s) \;=\; \frac{1}{\pi}\int_{-\infty}^{\infty} \log\tilde{V}(j\omega)\frac{x}{x^2+(y-\omega)^2}\mathrm{d}\omega. \tag{4.18}$$

Taking real parts on both sides (recall that $\mathrm{Re}(\log(z)) = \log(|z|)$) gives

$$\log|\tilde{V}(s)| \;=\; \frac{1}{\pi}\int_{-\infty}^{\infty} \log|\tilde{V}(j\omega)|\frac{x}{x^2+(y-\omega)^2}\mathrm{d}\omega. \tag{4.19}$$

Note that in the integrand on the right-hand side, we may replace $|\tilde{V}(j\omega)|$ by $|V(j\omega)|$. We set $y = 0$ and multiply both sides by πx to get

$$\pi x\log|V(x)| + \pi x\log|B_{\text{zeros}}^{-1}(x)| \;=\; \int_{-\infty}^{\infty}\log|V(j\omega)|\frac{x^2}{x^2+\omega^2}\mathrm{d}\omega \tag{4.20}$$

Letting $x \to \infty$, the right-hand side approaches $\int_{-\infty}^{\infty}|V(j\omega)|\mathrm{d}\omega$, and under the assumption that P has pole-zero excess greater than one, the first term on the left, $x\log|V(x)|$, approaches 0. For the remaining term on the left-hand side we have:

$$\begin{aligned}
\lim_{x\to\infty}\log|B_{\text{zeros}}^{-1}(x)| \;&=\; \lim_{x\to\infty}\log\prod_i\left|\tfrac{\bar{z}_i+x}{z_i-x}\right| \\
&=\; \lim_{x\to\infty}\mathrm{Re}\sum_i x\log\frac{1+\frac{\bar{z}_i}{x}}{1-\frac{z_i}{x}} \\
&=\; \sum_i \mathrm{Re}\;z_i \\
&\geq\; 0.
\end{aligned} \tag{4.21}$$

This concludes the proof. ∎

The result of Lemma 4.2.2 states that if we restrict attention to causal operators, the condition for perfect tracking cannot be satisfied for plants with relative degree greater than one. For plants having relative degree zero or one, we may distinguish between those that are minimum phase and those that are not. A *minimum-phase system* is a system having all its zeroes in the in the open left-half plane. For such systems, we have the following lemma.

Lemma 4.2.3. *Let $P \in \mathcal{RH}_\infty$ be a stable minimum-phase plant whose relative degree does not exceed one. Then we can always find $L \in \mathcal{RH}_\infty$ such that $|1 - L(j\omega)P(j\omega)| < 1$ for almost all $\omega \in \mathbb{R}$.*

Proof. Case (a): relative degree zero. Being minimum phase, it follow that P must be bistable. We select $L := P^{-1}$ so that $1 - LP = 0$. Case (b): relative degree one. In this case we select $L := ZP^{-1}$ with $Z := 1/(s+1)$. It follows that

$$|1 - L(j\omega)P(j\omega)|^2 \;=\; \frac{\omega^2}{\omega^2 + 1} \tag{4.22}$$

is less than unity for almost all $\omega \in \mathbb{R}$ (and hence so is $|1 - L(j\omega)P(j\omega)|$). This concludes the proof. ∎

For non-minimum phase systems of arbitrary relative degree, it is easy to verify that the condition for perfect tracking ($|1 - L(j\omega)P(j\omega)| < 1$ for almost all ω) is never satisfied. For suppose s_0 is an RHP zero of P. Then $1 - L(s_0)P(s_0) = 1$ and by the Maximum Modulus Principle we have that $\sup_\omega |1 - L(j\omega)P(j\omega)| \geq 1$ with equality ($=$) if and only if $1 - L(s)P(s)$ is a constant, that is, if and only $L = 0$. In any case $1 - L(j\omega)P(j\omega)$ equals or exceeds unity on a set of nonzero measure. In the next paragraph we show that regardless whether P is minimum phase and has relative degree less than two or not, we can always find $L \in \mathcal{RL}_\infty$ (the space of *non*causal operators) such that $|1 - L(j\omega)P(j\omega)| < 1$ for almost all $\omega \in \mathbb{R}$.

Implications for the two-parameter problem

To conclude this subsection, let us point out that our discussion is not limited to situations in which we seek to achieve perfect tracking. Indeed, let us consider a family of two-parameter iterations,

$$u_{k+1} \;=\; Qu_k + Le_k. \tag{4.23}$$

Let $Q, L \in \mathcal{RH}_\infty$ be an admissible pair ($\|Q - LP\|_\infty < 1$). Without loss of generality (see Section 3.4) we may assume that $Q - LP = 0$, in which case

we have

$$\begin{aligned} \bar{e} &= y_{\mathrm{d}} - P\bar{u} \\ &= y_{\mathrm{d}} - P\left(I - Q + LP\right)^{-1} L y_{\mathrm{d}} \\ &= \left(I - PL\right) y_{\mathrm{d}}. \end{aligned} \tag{4.24}$$

For $\bar{e}$ to be small, we require $|I - PL|$ to be small (in an appropriate frequency range). In an (equivalent) feedback setting, $I - PL$ would be the (output) sensitivity function $S = (I + PK)^{-1}$ associated with the controller $K = (I - LP)^{-1} L$. Bode's Sensitivity Integral states that, provided P has relative degree two or more, the sensitivity function can only be made small (smaller than unity) over a certain frequency range and necessarily peaks (exceeds unity) in another. More precisely,

$$\int_0^\infty \log|S(j\omega)|\mathrm{d}\omega \;\geq\; 0, \tag{4.25}$$

which is exactly the result of Lemma 4.2.2, be it in a different disguise. In other words, not only does Lemma 4.2.2 inform us about the (im)possibility of perfect tracking, it also shows that the performance achievable with Causal ILC is constrained in the indicated sense. As we will prove shortly, these design limitations are due to bandwidth constraints which may be overcome by resorting to noncausal operators.

4.2.4. Non-minimum phase plants and Poisson's Integral

A (SISO) real rational plant $P(s)$ is said to be non-minimum phase if it has one or more unstable zeros, i.e. zeros in the open RHP. For instance, the plant

$$P(s) \;:=\; \frac{s-1}{s+1} \tag{4.26}$$

has one unstable zero at $s = 1$. It is well-known that unstable zeros impose constraints on the achievable bandwidth in feedback systems—see Freudenberg and Looze (1985); Engell (1988). And we may expect the same to be true for causal ILC. Indeed, let $z_0 = x_0 + jy_0$ with $x_0 > 0$ be an open-RHP zero of P. Then for all $L \in \mathcal{RH}_\infty$ we have that

$$\int_{-\infty}^{\infty} \log(|1 - P(j\omega)L(j\omega)|)\frac{x_0}{x_0^2 + (y_0 - \omega)^2}\mathrm{d}\omega \;=\; \pi \log|B_{\mathrm{zero}}^{-1}(z_0)|, \tag{4.27}$$

where $B_{\text{zero}}^{-1}(s)$ is the Blaschke product

$$B_{\text{zeros}}^{-1}(s) \;:=\; \prod_i \frac{p_i - s}{\bar{p}_i + s}, \tag{4.28}$$

formed from open RHP *zeros* of $I - PL$ (we denote them as p_i to distinguish them from the open RHP zeros of P. The p_i's may alternatively be interpreted as the RHP *poles* of the (equivalent) controller $K = (I - LP)^{-1} L$). We define

$$w_{z_0}(\omega) := \frac{1}{\pi}\left(\frac{x_0}{x_0^2 + (\omega - y_0)^2} + \frac{x_0}{x_0^2 + (\omega + y_0)^2}\right), \tag{4.29}$$

and rewrite (4.27) as

$$\int_0^\infty \log(|I - P(j\omega)L(j\omega)|) w_{z_0}(\omega) \mathrm{d}\omega \;=\; \log|B_{\text{zeros}}^{-1}(z_0)|. \tag{4.30}$$

Inspection of equality (4.30) shows that its right-hand side is nonnegative. Since $w_{z_0}(\omega)$ is positive it follows that if $\log|I - PL|$ is negative over some frequency range, it must be positive over another. This we already know from the Bode's sensitivity relation, be it that (4.30) also holds for plants whose pole-zero excess is less than two. But because of the weighting function w_{z_0}, the result expressed by Equation (4.30) is stronger. We may rewrite w_{z_0} to obtain

$$w_{z_0}(\omega) \;=\; 2\mathrm{Re}\left(\frac{z_0}{\omega^2 + z_0^2}\right). \tag{4.31}$$

In case of our example (4.26) we have $z_0 = 1$. It follows that

$$w_1(\omega) \;=\; 2\left(\frac{1}{\omega^2 + 1}\right). \tag{4.32}$$

The function $w_1(\omega)$ is plotted in Figure 4.3. We may observe that the function is approximately constant for $\omega \ll 1$, whereas the function tends to ω^{-2} for $\omega \gg 1$. From (4.30) it follows that

$$\int_0^\infty \log(|I - P(j\omega)L(j\omega)|) \underbrace{\left(\frac{2}{\omega^2 + 1}\right)}_{w_1(\omega)} \mathrm{d}\omega \;\geq\; 0 \tag{4.33}$$

As $w_1(\omega) \leq 1$ for all $\omega \geq 1$, it follows that if the gain of (the sensitivity function) $1 - PL$ is smaller than unity for all $\omega \leq 1$, it is likely to peak at some frequency $\omega > 1$. Thus, if excessive peaks are to be avoided, the bandwidth $1 - PL$ should drop off (well) before $\omega = 1$. This observation generalizes as follows (see Engell (1988); Freudenberg and Looze (1985); Bosgra and Kwakernaak (2001))

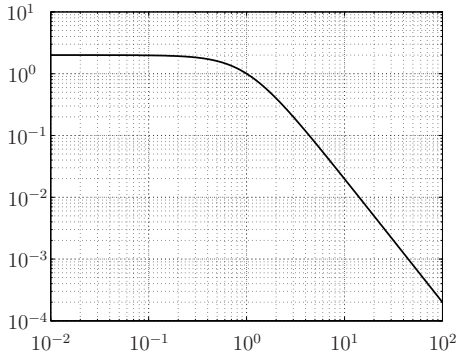


Figure 4.3.: The function $w_{z_0}(\omega)$ (4.31) for $z_0 = 1$.

1. The upper limit of the band over which effective disturbance attentuation is possible (i.e. in which $|1 - PL|$ is small) is constrained from above by the magnitude of the smallest RHP zero.

2. If the plant has unstable poles, the achievable performance is further impaired. This effect is especially pronounced when one or several RHP pole-zero pairs are close.

3. If the plant has no RHP zeros, the maximally achievable bandwidth is solely constrained by the plant capacity.

Observation (2) is listed only for sake of completeness; it does not concern us, since we have assumed that the plant is stable (as we are working in open loop). Observation (1) certainly applies, as we demonstrated above for the simple plant (4.26) having just one RHP zero. Observation (3) we already discussed in the previous paragraph with the constraints given by Lemma 4.2.2.

Let us now move on to show that with noncausal operators none of the aforementioned constraints applies. We have the following lemma.

Lemma 4.2.4. *Let $P \in \mathcal{RH}_\infty$ be a stable transfer matrix, having full normalrank. Then we can always find $L \in \mathcal{RL}_\infty$ such that*

$$\bar{\sigma}(I - P(j\omega)L(j\omega)) \quad \leq \quad 1, \tag{4.34}$$

for almost all $\omega \in \mathbb{R}$.

Proof. Define $L = (I + P^*P)^{-1} P^* = P^* (I + PP^*)^{-1}$. Here $P^* := P^T(-s)$ denotes the adjoint of P. Note that $I + P(j\omega)P^T(-j\omega)$ is positive definite for all ω, which implies that L does not have any poles on the imaginary

axis (and this includes the point at infinity). It follows that $L \in \mathcal{RL}_\infty$. We substitute L into $I - PL$ to get

$$\begin{aligned} I - PL &= I - PP^* \left(I + PP^*\right)^{-1} \\ &= \left(I + PP^*\right)^{-1}. \end{aligned} \tag{4.35}$$

Let us recall that $\bar{\sigma}\left(\left(I + PP^*\right)^{-1}\right) = 1/\underline{\sigma}\left(I + PP^*\right)$. Now we have

$$\underline{\sigma}\left(\left(I + P(j\omega)P^T(-j\omega)\right)\right) \geq 1, \tag{4.36}$$

where equality (=) holds if and only if $P(j\omega)$ looses rank (which may happen only a finite number of times as we have assumed that P has full normalrank). This concludes the proof. ∎

In conclusion, there is good reason to consider the use of noncausal operators in the context of Iterative Learning Control. In general, the performance we obtain by means of noncausal operators is at least as good and often better than what we can obtain with causal operators. In any case, it is evident that by insisting on causality, we simply deny the possibilities of open-loop feedback.

4.3. Equivalent Feedback

4.3.1. Admissible pairs

In Section 3.3, Definition 3.3.1, we formulated two conditions for a given pair of *causal* operators to be admissible with respect to a class of CCF-ILC iterations. As we are now extending our scope to the general class of *noncausal* operators, we are in want of a new definition. We could adapt Definition 3.3.1, replacing each instance of $\mathcal{RH}_\infty$ by $\mathcal{RL}_\infty$ and each instance of $\mathcal{H}_2$ by $\mathcal{L}_2(j\mathbb{R})$. But instead we propose the following, pragmatic definition (restricting attention a class of Standard ILC iterations).

Definition 4.3.1 (Noncausal admissible pairs). *Given u_0,$y_d \in \mathcal{L}_2(j\mathbb{R})$. Let $\{w_k\}_{k\geq 0}$ be a sequence such that $w_k \in \mathcal{L}_2(j\mathbb{R})$ for all k and $\sup_k \|w_k\| < \infty$. Consider the next class of iterations,*

$$u^w_{k+1} = Qu^w_k + Le^w_k + w_k, \quad k = 0, 1, \ldots \tag{4.37}$$

Assume $(Q, L) \in \mathcal{RL}_\infty$. We say that the pair (Q, L) is admissible (with respect to the said class of iterations) if

$$\sup_\omega \bar{\sigma}\left(Q(j\omega) - L(j\omega)P(j\omega)\right) < 1. \tag{4.38}$$

The above definition is motivated by the result of Lemma 3.4.1. We may note that if (Q, L) is admissible, then we have that the unperturbed iteration (Eq. (4.37) with $w_k \equiv 0$) converges to a unique fixed point $\bar{u} \in \mathcal{L}_2$. In Section 4.A.2 we prove that if $F := Q - LP$ is a contraction then so is the finite-time operator $F_{[0,T]}$. This implies that the corresponding iteration over finite time (for notation, see Appendix 4.A)

$$u'_{k+1} \;=\; Q_{[0,T]}u'_k + L_{[0,T]}e'_k \tag{4.39}$$

converges to a unique fixed point in $\mathcal{L}_2[0, T]$.

In line with the notation introduced in the previous chapter, we denote the set of all admissible pairs (in the sense of Definition 4.3.1) by $\mathcal{A}$. Additionally, we introduce the subset of all *causal* admissible pairs $\mathcal{A}_\mathrm{c}$

$$\mathcal{A}_c \;=\; \{(Q, L) \in \mathcal{A} : Q, L \in \mathcal{RH}_\infty\}\,. \tag{4.40}$$

We may now define an equivalence relation on $\mathcal{A}$ by saying that two pairs (Q_1, L_1), and (Q_2, L_2) (both in $\mathcal{A}$) are equivalent if

$$(I - Q_1)^{-1} L_1 \;=\; (I - Q_2)^{-1} L_2. \tag{4.41}$$

Again we can show that to each $(Q, L) \in \mathcal{A}$ there corresponds $(Q_0, L_0) \in \mathcal{A}$ such that $(Q_0, L_0) \simeq (Q, L)$ and $Q_0 - L_0P = 0$. Thus we may restrict attention to a set of respresentatives $\mathcal{A}_0 := \{(Q, L) \in \mathcal{A} : Q - LP = 0\}$. In Section 3.4 we found that $\mathcal{A}_0 \cap \mathcal{A}_\mathrm{c}$ and the set of all stabilizing controllers $\mathcal{K}$ are bijective. Let us see how this result generalizes.

4.3.2. Equivalent Feedback

For each $(Q, L) \in \mathcal{A}$ let us define the candidate equivalent controller

$$K \;:=\; (I - Q)^{-1} L \tag{4.42}$$

We ask: is K always stabilizing? In view of our conclusions on the use of noncausal operators, the answer should be 'no'. For suppose K would be stabilizing for each $(Q_\mathrm{nc}, L_\mathrm{nc}) \in \mathcal{A}$, then by Theorem 3.4.8 we know that there must exist $(Q_\mathrm{c}, L_\mathrm{c}) \in \mathcal{A}_c$ such that $(Q_\mathrm{c}, L_\mathrm{c}) \simeq (Q_\mathrm{nc}, L_\mathrm{nc})$. But this would imply that there is no point in considering noncausal operators, which is in conflict with our earlier findings. Thus we conclude that K is generally destabilizing.

To make this argument precise, let us restrict attention to the smaller set of representatives $\mathcal{A}_0 := \{(LP, L) : L \in \mathcal{RL}_\infty\}$. For each $(LP, L) \in \mathcal{A}_0$ we have (compare (4.42))

$$K_0 \;=\; (I - LP)^{-1} L. \tag{4.43}$$

(Zhou et al., 1996, Corollary 5.5) states that K_0 is stabilizing if and only if

$$K_0 \left(I + PK_0\right)^{-1} \quad \in \quad \mathcal{RH}_\infty. \tag{4.44}$$

By substitution of (4.43) we find that this condition is equivalent with

$$L \quad \in \quad \mathcal{RH}_\infty. \tag{4.45}$$

Since we have assumed that $L \in \mathcal{RL}_\infty$, we conclude that the candidate equivalent controller (4.43) is generally destabilizing. In fact, Condition (4.45) suggests that the set of representatives $\mathcal{A}_0$ may be divided into a set $\mathcal{A}_0 \cap \mathcal{A}_c$, isomorphic to the set of stabilizing controllers, and a set $\mathcal{A}_0 - (\mathcal{A}_0 \cap \mathcal{A}_c)$, isomorphic to the set of destabilizing controllers.

4.3.3. A 2D 'Equivalent Control' Configuration

Let $L \in \mathcal{RL}_\infty - \mathcal{RH}_\infty$ and $P \in \mathcal{RH}_\infty$. From the discussion in Section 4.3.2 it is clear that no causal LTI map $C : \mathcal{L}_2 \mapsto \mathcal{L}_2$ exist for which the feedback system

$$u \quad = \quad C(y_\mathrm{d} - y) \tag{4.46}$$
$$y \quad = \quad Pu \tag{4.47}$$

has a unique bounded solution $(u, y) \in \mathcal{L}_2^2$ such that in addition $y = PLy_d$.

The fact that no such map exists does not imply that no other 'equivalent control' configurations exist. Indeed, suppose there exist $L_\mathrm{c}(s)$, $L_\mathrm{ac}(s)$, such that

1. $L(s) = L_\mathrm{c}(s)L_\mathrm{ac}(s)$,
2. $L_\mathrm{c}(s), L_\mathrm{ac}(-s) \in \mathcal{RH}_\infty$,

and consider the 2D control configuration depicted in Figure 4.4. It is easy to verify that this system implements the map $y = PLy_d$ by means of an anticausal prefilter $L_\mathrm{as}(s)$ and a stabilizing compensator

$$K' := \left(I - L_\mathrm{c}P\right)^{-1} L_\mathrm{c}. \tag{4.48}$$

From a synthesis point of view, this control configuration is equivalent with the class of noncausal iterations. This is evident from the fact that with $(Q, L) \in \mathcal{A}_0$ (which we may assume without loss of generality), the sequence of outputs $\{Pu_k\}$, induced by the iteration

$$u_{k+1} \quad = \quad Qu_k + Le_k \tag{4.49}$$

converges to some $\bar{y} \in \mathcal{L}_2(\mathbb{R})$,

$$\bar{y} = PLy_d. \tag{4.50}$$

From an implementation point of view however, the two are clearly not equivalent. For as soon as the model P disagrees with the true plant, the 2D control configuration no longer matches the virtual feedback solution induced by the family of noncausal iterations. This is an important distinction with equivalent feedback for causal ILC.

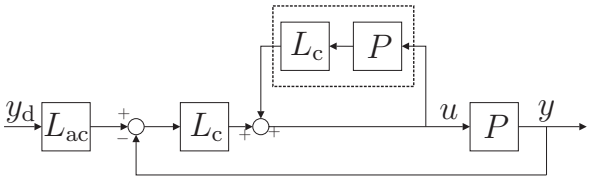


Figure 4.4.: A 2D 'Equivalent Control' configuration. With $L_c(s), L_{ac}(-s) \in \mathcal{RH}_\infty$ and $L := L_c L_{ac}$, this system implements the map $y = PLy_d$.

4.4. Discussion

We considered the problem of noncausal ILC and argued that there is good reason to consider the use of noncausal operators. This is most evident in non-mimimum phase systems, for in such systems, the design limitations imposed by the causility constraints are particularly restrictive. But also in minimum phase systems with high relative degree, noncausal operators may significantly improve the achievable performance.

Our discussion focused on fundamental issues. Much less attention was given to the actual synthesis problem. The one 'procedure' that we did consider (see Section 4.2, particularly Eq. (4.3)) aims at perfect tracking and assumes full plant knowledge. It may not be robust against plant uncertainty. Possibly, we could robustify this algorithm, by re-introducing the Q-parameter (which was now set to I). In its generality however, the synthesis problem is still open.

4.A. Preliminaries: Noncausal operators

4.A.1. Notation and Terminology

With slight abuse of terminology we say that a noncausal operator is an operator that is not *necessarily* causal, in the sense that the family of noncausal operators includes, but does not coincide with, the family of causal operators. Indeed, also *anti*-causal operators (to be defined shortly) belong to the set of noncausal operators, as do (linear) combinations of causal and anti-causal operators. The Venn diagram in Figure 4.5 illustrates the relations among the different classes.

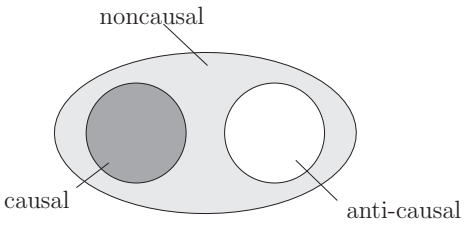


Figure 4.5.: Venn diagram representing the class of noncausal operators, which includes that of causal, and anti-causal operators.

In this chapter we consider noncausal, finite-dimensional LTI operators of the form

$$\begin{aligned}(Gu)(t) &:= Du(t) + (g * u)(t) \\ &= Du(t) + \int_{-\infty}^{\infty} g(t-\tau)u(\tau)\mathrm{d}\tau. \end{aligned} \tag{4.51}$$

We assume that $\lim_{t\downarrow 0} g(t)$ and $\lim_{t\uparrow 0} g(t)$ exist, but they need not be equal. Recall that G is said to be *causal* if $g(t)$ is zero for negative time. The rationale behind this terminology is evident from (4.51); if the said condition is satisfied, the value of Gu, evaluated at time t, solely depends on the *past* of u, which, for any given t_o may be defined as the set $\{u(t) : t \leq t_0\}$. In case g is zero for positive time, the value of Gu, evaluated at a certain time t will also depend on the *future* of u ($\{u(t) : t \geq t_0\}$). By the same token, G is *anti-causal* if $g(t)$ is zero for positive time, in which case its output is solely determined by future values of its input.

Each G of the form (4.51) can be written as the sum of a causal and an anti-causal operator, as follows. Select $D_{\mathrm{c}}, D_{\mathrm{ac}}$, such that

$$D = D_{\mathrm{c}} + D_{\mathrm{ac}}, \tag{4.52}$$

and define:

$$g_{\rm c}(t) \quad = \quad \begin{cases} g(t) & \text{if } t \geq 0 \\ 0 & \text{elsewhere} \end{cases} \quad ; \quad g_{\rm ac}(t) \quad = \quad \begin{cases} g(t) & \text{if } t < 0 \\ 0 & \text{elsewhere} \end{cases} \tag{4.53}$$

We may note that $g(t) \equiv g_{\rm c} + g_{\rm ac}$. Now we form the pair of operators

$$\begin{aligned} (G_{\rm c}u)(t) &:= D_{\rm c}u(t) + \int_{-\infty}^{\infty} g_{\rm c}(t-\tau)u(\tau){\rm d}\tau; \\ (G_{\rm ac}u)(t) &:= D_{\rm ac}u(t) + \int_{-\infty}^{\infty} g_{\rm ac}(t-\tau)u(\tau){\rm d}\tau. \end{aligned} \tag{4.54}$$

Observe that $G_{\rm c}$ is causal, and that $G_{\rm ac}$ is anti-causal. In addition, $G_{\rm c} + G_{\rm ac} = G$ (which is to say that $G_{\rm c}u + G_{\rm ac}u = Gu$ for all u in the domain of G).

For every $g \in \mathcal{L}_1(-\infty, \infty)$ we define the *two-sided* Laplace Transform

$$G(s) \quad := \quad \int_{-\infty}^{\infty} g(t)e^{-st}{\rm d}t, \tag{4.55}$$

whose domain of definition extends over all $s \in \mathbb{C}$ for which the above integral converges. The two-sided Laplace transform and its inverse define an (isometric) isomorphism between the time-domain space $\mathcal{L}_2(-\infty, \infty)$ and the frequency-domain space $\mathcal{L}_2(j\mathbb{R})$ and their respective subspaces $\mathcal{L}_2[0, \infty)$ and $\mathcal{H}_2$. Applied to the impulse response of a stable finite-dimensional LTI operator (which, in terms of (4.51), is not g, but rather $D\delta(t) + g(t)$, where $\delta(t)$ is the Dirac Delta function), it induces two spaces of interest. The first, corresponding to the class of *causal* operators, is the space $\mathcal{RH}_\infty$. It consist of all proper and real rational transfer matrices with no poles in the closed right-half plane. The second, corresponding to the class of all stable finite-dimensional LTI operators, possibly noncausal, is $\mathcal{RL}_\infty$. It consists of all proper and real rational transfer matrices with no poles on the imaginary axis. A transfer matrix with poles in the open right-half plane may define a stable noncausal system. For example $G(s) = 1/(1-s)$ is the *two*-sided Laplace transform of

$$g(t) = \begin{cases} e^t & \text{if } t < 0 \\ 0 & \text{elsewhere} \end{cases}$$

(defined for all $s \in \mathbb{C}$ with $\mathrm{Re}(s) < 1$). It defines a stable system $y = g*u$. In general, an element $G(s) \in \mathcal{RL}_\infty$ may have poles both in the open left-half, and open right-half plane. The poles in the open right-half plane correspond

to the anti-causal part of the impulse response (typically consisting of a sum of exponentials that decrease with negative time). Every element $G(s)$ in $\mathcal{RL}_\infty$ may be decomposed into a causal part $G_{\rm c}(s)$ and an anticausal part $G_{\rm ac}(s)$ such that $G(s) = G_{\rm c}(s) + G_{\rm ac}(s)$ with $G_{\rm c}(s), G_{\rm ac}(-s) \in \mathcal{RH}_\infty$.

In Section 4.A.2 we show that the action of the antistable operator $G_{\rm ac}(-s)$ may be implemented as a causal operation on a time-reversed signal. Hence, the location of the poles (relative to the imaginary axis) merely determines the causality of the operation.

For every $G(s) \in \mathcal{RL}_\infty$, the $\mathcal{L}_2(-\infty,\infty)$-induced operator norm is defined as

$$\|G\|_\infty \;=\; \sup_{\omega\in\mathbb{R}} \bar\sigma\left(G(j\omega)\right), \tag{4.56}$$

where $\bar\sigma$ denotes the largest singular value.

Example 4.A.1. *We consider the noncausal operator $G : \mathcal{L}_2(-\infty,\infty) \mapsto \mathcal{L}_2(-\infty,\infty)$, $Gu(t) := (g * u)(t)$, where g is given as*

$$g(t) = \begin{cases} e^t & \text{if } t < 0 \\ e^{-t} & \text{elsewhere} \end{cases}.$$

Note that G is of the form (4.51) (the feedthrough term, D, is absent). Let us decompose G into a causal part $(G_c u)(t) := \int_{-\infty}^{\infty} g_c(t-\tau)u(\tau)d\tau$,

$$g_c(t) \;:=\; \begin{cases} e^{-t} & \text{if } t \geq 0 \\ 0 & \text{elsewhere} \end{cases} \tag{4.57}$$

and a noncausal part $(G_{ac}u)(t) := \int_{-\infty}^{\infty} g_{ac}(t-\tau)u(\tau)d\tau$

$$g_{ac}(t) \;:=\; \begin{cases} e^{t} & \text{if } t < 0 \\ 0 & \text{elsewhere} \end{cases} \tag{4.58}$$

We compute the two-sided Laplace transform $G(s)$

$$\begin{aligned} G(s) \;&=\; \int_{-\infty}^{0} e^t e^{-st}dt + \int_{-\infty}^{0} e^{-t}e^{-st}dt \\ &=\; \frac{1}{1-s} + \frac{1}{1+s} \end{aligned} \tag{4.59}$$

Equation (4.59) suggests that we decompose $G(s)$ into a part

$$G_c(s) \;:=\; \frac{1}{(1+s)} \tag{4.60}$$

and a part

$$G_{ac}(s) \;:=\; \frac{1}{(1-s)} \tag{4.61}$$

such that $G(s) = G_c(s) + G_{ac}(s)$ *and* $G_c(s), G_{ac}(-s) \in \mathcal{RH}_\infty$.

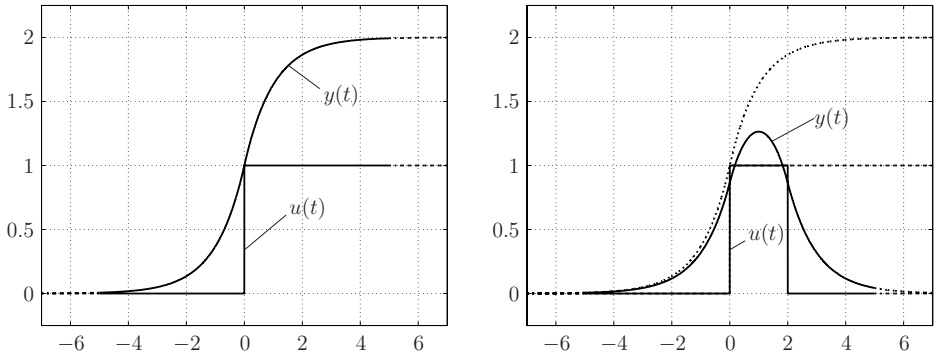
We apply the input

$$u(t) \;=\; \begin{cases} 1 & \text{if } 0 \le t \le T \\ 0 & \text{elsewhere} \end{cases} \tag{4.62}$$

and compute the output $y(t)$ *as*

$$y(t) \;=\; \begin{cases} 2 - e^{-t} - e^{(t-T)} & \text{if } 0 \le t \le T \\ e^{t} - e^{(t-T)} & \text{if } t < 0 \\ e^{-(t-T)} - e^{-t} & \text{elsewhere} \end{cases} \tag{4.63}$$

The response of the system is shown in Figure 4.6 for two different values of T. *Observe that although the respective inputs are equal for all* $t \le 2$, *the outputs are not. This is a general feature of noncausal systems, and something to keep in mind: we cannot just ignore the future as we sometimes do with causal systems.*



(a) $T = \infty$ (b) $T = 2$

Figure 4.6.: The response of the system $y(t) := (g * u)(t)$ to the input (4.62) for different values of T.

4.A.2. Noncausal operators on $[0, T]$

In the framework introduced, we assumed that all signals are defined over infinite time. For causal systems we could easily justify this assumption by pointing to the following facts:

1. If $u(t) = 0$ for all $t < t_0$, then for any causal system G, we have that $(Gu)(t) = 0$ for all $t < t_0$. Without loss of generality t_0 can be set to zero. In effect, we can ignore negative time.

2. If $u_1(t) = u_2(t)$ for all $t < t_0$ then $(Gu_1)(t) = (Gu_2)(t)$ for all $t < t_0$. Thus, the truncated response of the original system (defined over infinite time) coincides with the response of the truncated system (defined over finite time).

When we give up causality, we allow a system's response to be partly determined by future values of its input. As we are generally dealing with systems having an infinite impulse response (IIR), this means that in order to compute the output at any given time, we need to know the input's complete future. Needless to say, we generally lack such complete information. In ILC, signals are defined on $[0, T]$. That means that, as t approaches T, the window on the future gets smaller and smaller. To account for this fact, we do the following. With each G defined over infinite time, we associate an operator $G_{[0,T]}$, defined over finite time. Let G be a stable (in the sense of $\mathcal{L}_2(\mathbb{R}) \mapsto \mathcal{L}_2(\mathbb{R})$) convolution operator of the form

$$(Gu)(t) \;=\; Du(t) + \int_{-\infty}^{\infty} g(t-\tau)u(\tau)\mathrm{d}\tau \quad (-\infty < t < \infty). \tag{4.64}$$

We define the *trunctated operator* $G_{[0,T]} : \mathcal{L}_2[0, T] \mapsto \mathcal{L}_2[0, T]$ as

$$(G_{[0,T]}u)(t) \;:=\; Du(t) + \int_{0}^{T} g(t-\tau)u(\tau)\mathrm{d}\tau \quad (0 \le t \le T). \tag{4.65}$$

The former is used for analysis and synthesis; the latter for actual implementation. Naturally, the analysis would not make sense, if we would not first verify that certain statements about G translate in a useful way to statements about $G_{[0,T]}$. In the next subsection we prove a particularly useful proposition, which relates contractivity of $G_{[0,T]}$ to that of G.

4.A.3. A contraction on $\mathcal{L}_2[0, T]$

In Example 4.A.1 we saw that $(Gu)(t)$ and $(G_{[0,T]}u)(t)$ do not (necessarily) coincide on $[0, T]$. To see how G and $G_{[0,T]}$ are related, let us define the projection operator $\Lambda_{[0,T]} : \mathcal{L}_2(-\infty, \infty) \mapsto \mathcal{L}_2(-\infty, \infty)$,

$$(\Lambda_{[0,T]}u)(t) \;:=\; \begin{cases} u(t) & \text{if } 0 \le t \le T \\ 0 & \text{elsewhere} \end{cases} \tag{4.66}$$

The linear operator Λ—we drop the index—is idempotent, i.e. $\Lambda(\Lambda u) = u$ for all u. Moreover, Λ is nonexpansive, i.e. $\|\Lambda\| = 1$. We have the following lemma.

Lemma 4.A.2. *Let G be a stable LTI operator of the form (4.64) and let $G_{[0,T]}$ denote the corresponding truncated operator (4.65). For all $u \in \mathcal{L}_2(-\infty,\infty)$, define $u' \in \mathcal{L}_2[0,T]$ as $u'(t) := u(t)$, $0 \le t \le T$. We have that*

1. *$(G_{[0,T]}u')(t) = (G\Lambda u)(t)$ for all $t \in [0,T]$.*
2. *$\left\|G_{[0,T]}u'\right\|_{\mathcal{L}_2[0,T]} = \left\|\Lambda G\Lambda u\right\|_{\mathcal{L}_2(-\infty,\infty)}$.*
3. *If G is a contraction on $\mathcal{L}_2(-\infty,\infty)$, then $G_{[0,T]}$ is a contraction on $\mathcal{L}_2[0,T]$.*

Proof. Part (1): For all $t \in \mathbb{R}$, the response $(G\Lambda u)(t)$ is given as

$$\begin{aligned}(G\Lambda u)(t) &= Du(t) + \int_{-\infty}^{\infty} g(t-\tau)\Lambda u(\tau)\mathrm{d}\tau \\ &= Du(t) + \int_{0}^{T} g(t-\tau)u(\tau)\mathrm{d}\tau\end{aligned}$$

On $[0,T]$ we have that $u(t) = u'(t)$. Thus for all $t \in [0,T]$

$$(G\Lambda u)(t) = (G_{[0,T]}u')(t) \tag{4.67}$$

Note that $(Gu)(t)$ is also defined for $t < 0$. Part (2): The statement follows from

$$\begin{aligned}\left\|\Lambda G\Lambda u\right\|^2_{\mathcal{L}_2(-\infty,\infty)} &:= \int_{-\infty}^{\infty} \left[(\Lambda G\Lambda u)(t)\right]^T \left[(\Lambda G\Lambda u)(t)\right] \mathrm{d}t \\ &= \int_{0}^{T} \left[(Gu)(t)\right]^T \left[(Gu)(t)\right] \mathrm{d}t \\ &= \int_{0}^{T} \left[(Gu')(t)\right]^T \left[(Gu')(t)\right] \mathrm{d}t \\ &= \left\|G_{[0,T]}u'\right\|^2_{\mathcal{L}_2[0,T]}\end{aligned} \tag{4.68}$$

Part (3): For $G_{[0,T]}$ to be a contraction, there must exist $\lambda < 1$ such that $\|G_{[0,T]}\tilde{u}\| \le \lambda\|\tilde{u}\|$ for all $\tilde{u} \in \mathcal{L}_2[0,T]$. From part (2) we deduce that $\|G_{[0,T]}\tilde{u}\| \le \lambda\|\tilde{u}\|$ for all $\tilde{u} \in \mathcal{L}_2[0,T]$ if and only if $\|\Lambda G\Lambda u\| \le \lambda\|\Lambda u\|$ for all $u \in \mathcal{L}_2(-\infty,\infty)$. Suppose G is a contraction, i.e. $\|G\| < 1$, then we have

$$\|\Lambda G\Lambda u\| \le \|G\Lambda u\| \le \|G\|\|\Lambda u\| \tag{4.69}$$

Here we used the fact that Λ is nonexpansive. Equation (4.69) shows that there exists λ ($\lambda = \|G\|$) such that $\|\Lambda G \Lambda u\| \leq \lambda \|\Lambda u\|$ for all u. This in turn implies that $G_{[0,T]}$ is a contraction. This concludes the proof. ∎

4.A.4. Implementation issues

Next we show that the action of G can be evaluated in terms of causal operations on u and Πu. Let us define the permutation operator $\Pi_T : \mathcal{L}_2(-\infty, \infty) \mapsto \mathcal{L}_2(-\infty, \infty)$,

$$(\Pi_T u)(t) \;:=\; u(T-t) \tag{4.70}$$

It is easy to verify that Π_T is linear, and norm-preserving ($\|\Pi_T u\| = \|u\|$ for all u). Moreover, $\Pi_T(\Pi_T u) = u$. Figure 4.7 shows the effect of Π_T, acting on a particular u. In the sequel we drop the index T and simply denote Π_T as Π. We select $g_{\rm c}, g_{\rm ac}, D_{\rm c}, D_{\rm ac}$ in accordance with (4.52)-(4.53), and define

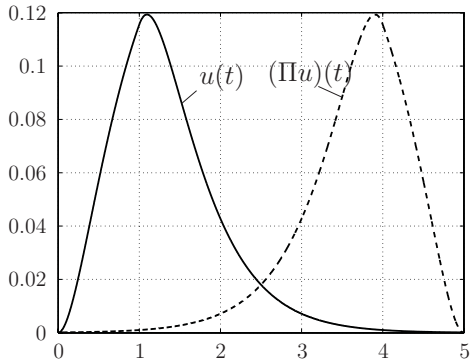


Figure 4.7.: The action of the permutation operator Π_T. The solid curve is $u(t)$; the dashed $(\Pi_T u)(t)$. The plot corresponds to the parameter value $T = 5$.

$$\begin{aligned} y_{\rm c}(t) \;&:=\; D_{\rm c} u(t) + \int_{-\infty}^{\infty} g_{\rm c}(t-\tau) u(\tau) {\rm d}\tau; \\ y_{\rm ac}(t) \;&:=\; D_{\rm ac} u(t) + \int_{-\infty}^{\infty} g_{\rm ac}(t-\tau) u(\tau) {\rm d}\tau. \end{aligned} \tag{4.71}$$

Verify that $y(t) = y_{\rm c}(t)+y_{\rm ac}(t)$ for all $t \in [0,T]$. The noncausal contribution $y_{\rm ac}(t)$ is computed as follows

$$\begin{aligned}
(\Pi y_{\rm ac})(\eta) &= y_{\rm ac}(T-\eta) \\
&= D_{\rm ac}(\Pi u)(\eta) + \int_{T-\eta}^{\infty} g_{\rm ac}(T-\tau-\eta)u(\tau)\mathrm{d}\tau \\
&= D_{\rm ac}(\Pi u)(\eta) + \int_{T-\eta}^{\infty} g_{\rm ac}(T-\tau-\eta)(\Pi u)(T-\tau)\mathrm{d}\tau \\
&= D_{\rm ac}(\Pi u)(\eta) + \int_{-\infty}^{\eta} g_{\rm ac}(-(\eta-\xi))(\Pi u)(\xi)\mathrm{d}\xi \\
&:= G^*_{\rm ac}(\Pi u) \qquad (4.72)
\end{aligned}$$

Note that since $g_{\rm ac}(-t) = 0$ for all $t < 0$, the above defines a causal operation on Πu, which we denote as $G^*_{\rm ac}(\Pi u)$. When we solve for $y_{\rm ac}(\eta)$, we get

$$\begin{aligned}
y_{\rm ac} &= \Pi(\Pi y_{\rm ac}) \\
&= \Pi G^*_{\rm ac}\Pi u \qquad (4.73)
\end{aligned}$$

The causal part evaluates to $y_{\rm c} = G_{\rm c}u$, so that

$$y = \left(G_{\rm c} + \Pi G^*_{\rm ac}\Pi\right) u \qquad (4.74)$$

Let us once again emphasize that $y(t)$ is the response of the system G to the input u which, in general, does not coincide with the response of the system $G_{[0,T]}$ to the truncated input u'. The latter response may be computed from (see Lemma 4.A.2)

$$(G_{[0,T]}u')(t) = \left(G_{\rm c} + \Pi G^*_{\rm ac}\Pi\right)\Lambda u(t) \qquad (4.75)$$

Example 4.A.3. *To illustrate the ideas presented in this section, let us once again consider the operator G previously defined in Example 4.A.1. For this system we have*

$$\begin{aligned}
(G_c v)(t) &:= D_c v(t) + \int_{-\infty}^{\infty} g_c(t-\tau)v(\tau)\,d\tau \\
&= \int_{-\infty}^{t} e^{-(t-\tau)}v(\tau)\,d\tau \qquad (4.76) \\
(G^*_{ac}w)(t) &:= D_{ac}w(t) + \int_{-\infty}^{\infty} g_{ac}(-(t-\tau))w(\tau)\,d\tau \\
&= \int_{-\infty}^{t} e^{-(t-\tau)}w(\tau)\,d\tau \qquad (4.77)
\end{aligned}$$

The output of the truncated system may be evaluated as

$$\begin{aligned}(G_{[0,T]}u')(t) &= (G_c\Lambda u)(t) + (\Pi G_{ac}^*\Pi\Lambda u)(t) \\ &= \int_0^t e^{-(t-\tau)}u'(\tau)d\tau + \Pi\int_0^t e^{-(t-\tau)}(\Pi u')(\tau)d\tau \qquad (4.78)\end{aligned}$$

This corresponds to the schematic shown in Figure 4.8. The general schematic (for arbitrary G) may be obtained by substuting $G_c(s)$ for $\frac{1}{s+1}$ in the box labeled G_c, and $G_{ac}(-s)$ for $\frac{1}{s+1}$ in the box labeled G_{ac}. On the interval $[0, 10]$, we apply the input $u(t)$,

$$u(t) = \begin{cases} 1 & \text{if } 0 \le t \le 3 \\ 0 & \text{elsewhere} \end{cases} \qquad (4.79)$$

The system's reponse $(G_{[0,10]}u)(t)$ is given in Figure 4.9.

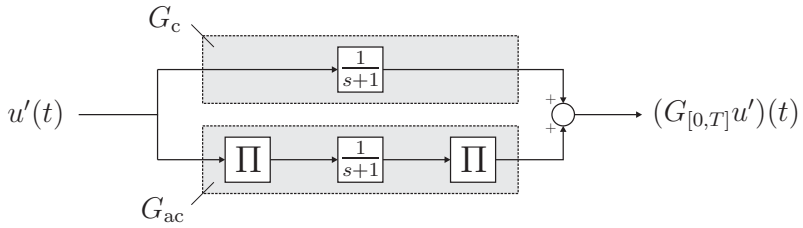
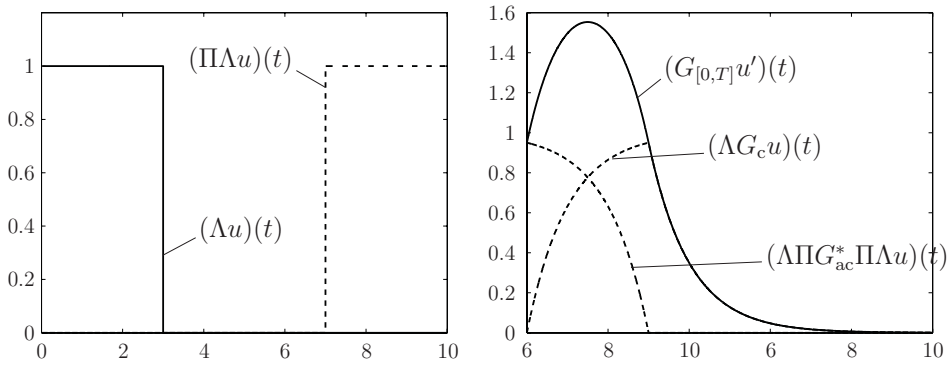


Figure 4.8.: Causal implementation of the noncausal operator $G_{[0,T]}$.



(a) $u'(t)$ (b) $(G_{[0,T]}u')(t)$

Figure 4.9.: The system's response $(G_{[0,T]}u')(t)$ (right) to the input (4.79) (left). In the left plot, the solid and the dashed plot respectively depict $(\Lambda u)(t)$ and $(\Pi\Lambda u)(t)$. In the right plot, the solid curve is the actual response, and the dashed curves correspond to the causal part $(\Lambda G_{\mathrm{c}}u)(t)$ and the noncausal part $(\Lambda\Pi G^*_{\mathrm{ac}}\Pi\Lambda u)(t)$ respectively.

5

Learning and adaptation: trial-dependent update rules

Overview – *In this final chapter we return to the general problem of Learning Control. Considering the algorithms discussed in previous chapters, we ask: "In what sense, and to what extent, do these algorithms implement a 'learning' behavior?". Key issue is their alleged ability to adapt; focal point, the use of iteration in relation to plant uncertainty.*

We argue that even as these algorithms succesfully enhance performance, they do not necessarily 'learn', in the sense that this 'succes' is typically due to, and thus dependent on, detailed plant knowledge. We identify this to be a typical feature of 'trial-independent' update laws (constant-coefficient recurrences), and by that we are led to consider 'higher-level' update laws, that is: update laws that update update laws, etc. We provide two elementary examples of 'level-one' update laws, and show that both have certain adaptive abilities (in a sense to be made precise).

As we turn to discuss the use of iteration, we argue that this use includes, but is not limited to, performance refinement. We propose an alternative, not necessarily conflicting view, in which the use of iteration is explained in terms of a need for information—a need arising from a lack of a priori *knowledge. We argue that rather than having all algorithms conform to a particular structure, we ought to have them conform to the data.*

5.1. Introduction

In the preceding chapters we concerned ourselves with questions of analysis and synthesis, relating to a family of first-order linear recurrences, parameterized by a pair of bounded linear operators. As we initially assumed both operators to be causal (Chapter 3), we soon dropped this constraint (Chapter 4) so as to maximally exploit ILC's open-loop nature. The main findings pertaining to the distinction between causal and noncausal update laws are summarized below.

- Under the assumption of *causality* the said family of recurrences allows for an Equivalent Feedback implementation. The associated (causal) feedback map is a function of the synthesis parameters only and is always stabilizing (provided the underlying update law is convergent). Ergo, the feedback system can be implemented without problems. In case of Standard ILC (open loop, stable plant) the achievable performance (bandwidth) is constrained by the plant capacity (the pole-zero excess) and the presence of right-half plane zeroes. In case of CCF-ILC (current-cycle feedback, possibly unstable plant) the achievable performance is further impaired by the RHP poles of the open-loop gain, and by the fact that the set of Equivalent Controllers is generally just a subset of *all* stabilizing controllers (unless the current-cycle operator is stable).

- In a *noncausal* setting, performance is constrained neither by the presence of RHP zeroes, nor by the plant capacity. A candidate Equivalent Controller can be defined, but its 'application' is limited to *virtual* feedback systems since the controller is noncausal and the actual (closed-loop) feedback system ill-posed.

Now we ask ourselves the following question: In view of the basic ideas that make up Learning Control, what can we say about this problem we solved? And more in particular, what is it these algorithms can do and what is it they cannot do?

To start with, let us outline the problem we solved; A somewhat simplified version reads as follows: Given a (SISO) plant $P : U \mapsto Y$, along with some desired output $y_{\mathrm{d}} \in Y$, find a map $F : U \times Y \mapsto U$,

$$u_{k+1} \;:=\; F(u_k, e_k), \tag{5.1}$$

such that $\lim_{k\to\infty} Pu_k = y_{\mathrm{d}}$. We obtained the following solution:

$$F(u_k, e_k) \;=\; u_k + \left(\gamma^2 I + P^* P\right)^{-1} P^* e_k. \tag{5.2}$$

Here, $P^* := P^T(-s)$ denotes the adjoint of the plant and $\gamma > 0$ is a parameter that controls the speed of convergence (recall that γ penalizes the size of the increment $\|\Delta u_k\|$ relative to that of the error $\|e_{k+1}\|$—for details refer to Section 2.2.2).

Implementation of the update law (5.1) with F as in (5.2) involves *non-causal* operations. That, however, is not a problem, as the update is computed offline, *after* completion of the current cycle.

By construction, the *output* sequence $\{Pu_0, Pu_1, \ldots\}$ converges for all $u_0 \in U$ and all $y_\mathrm{d} \in Y$. The input sequence $\{u_0, u_1, \ldots\}$ converges if and only if y_d is a *feasible* output.

This, in a nutshell, is the problem we solved. Now what about the problem we started out with? What about "achieving a desired result when only partial knowledge of the plant is available" (Section 1.3.2)? Indeed, what about 'learning'?

5.2. The problem of Learning Control

5.2.1. Performance, performance robustness and learning

As we take up the question of *learning*, we need to consider at least two aspects. First we must look at *performance* proper; Is it true that, relative to conventional means, these algorithms enhance performance? The answer is, unmistakably, 'yes'. Then there is the issue of uncertainty; Is it true that, again relative to conventional means, these algorithms enhance performance *robustness* (against uncertainty in the plant model)? In this case, the answer, as we will expound on presently, is 'no'.

For one may note that, in the problem statement above, P was assumed to be *given*. This is evidently a strong, not to say restrictive, assumption, and it remains to be seen whether this assumption can be relaxed without compromising performance. And so we wonder:

- If we are to assume full plant knowledge, then why engage in a process of physical iteration? Why not compute the optimal input *offline*, through *numerical iteration*?
- If we are to relax the assumption of full plant knowledge, *how much do we need to assume* to make this algorithm converge? Can we make it more robust without compromising performance?
- Lastly, if the latter question is to be answered in the negative, then what makes this an algorithm for Learning Control, if not its ability to deal with uncertainty?

Why iterate in case of full plant knowledge?

Let us explore these issues in some more detail. First the issue of full plant knowledge and the possibility of doing numerical, rather than physical iteration. Our perspective is that physical *interaction* with the plant is one of the defining features of Learning Control; The data itself is to steer the plant towards a desired behavior. In case of full plant knowledge, there is nothing we can learn that we do not already know. In other words, there would be simply no need for Learning Control.

On performance robustness

What if we relax this assumption of full plant knowledge? Indeed, what if there is a mismatch between the actual plant and the plant model, will the algorithm still converge? It is hard to say anything in general, but the following example may give us some insight.

Suppose the actual plant P is given as $P := P_0\,(1+\delta_M)$ where P_0 represents the nominal plant, and δ_M is a bounded multiplicative pertubation. For the given update law to converge the next condition must be satisfied:

$$|1 - \frac{(P_0^* P_0)}{(\gamma^2 + P_0^* P_0)}\,(1+\delta_M)\,| \;<\; 1 \quad \text{for almost all } \omega. \tag{5.3}$$

This implies that $1+\delta_M$ must be positive real. We may observe that cannot be met in case P has two or more parasatic poles. Whether this makes the algorithm sufficiently robust or not will depend on the specific application. The important thing for us to note, is that for those plants that do satisfy Condition (5.3) the error converges to zero, whereas for those that do not, the error diverges to infinity. By construction, stability (convergence) implies performance. At first sight, this appears to be at variance with the situation in feedback control, where performance and stability robustness are typically *traded off*. However, the discrepancy dissapears when we consider generalized update laws such as (5.4), this being a 'robustified' version of update law (5.2):

$$u_{k+1} \;=\; Q\left(u_k + \left(\gamma^2 I + P^* P\right)^{-1} P^* e_k\right). \tag{5.4}$$

We may assume that Q is *nonexpansive* so that indeed the condition for convergence is relaxed. If the nominal design is competitive, we get good performance whenever the actual plant is close to the nominal plant, reasonable performance when it is not so close and no performance whatsoever when the actual plant deviates from the nominal plant to the extent that the algorithm no longer converges. Stability robustness may be increased

by lowering the gain of Q, and, in a typical situation, by reducing its bandwidth (generically, we cannot assume Q to be of a particular form, but if we confine ourselves to e.g. servomechanisms, it makes sense to assume that Q has lowpass characteristics). In view of the original algorithm, we argue that, for good performance, Q should be close to unity ($Q = 1$). In other words, we cannot get good performance *and* stability robustness at the same time. So, in general, the synthesis problem in ILC *does* exhibit a tradeoff between performance and stability robustness.

Learning Control?

We cannot but conclude that, like in conventional feedback control, here too uncertainty poses a threat, as good performance cannot be obtained without compromising the stability robustness. Back in Chapter 1, when we motivated the concept of Learning Control, we pointed to a potential for dealing with uncertain systems. We trusted that our algorithm had inherited some of this potential.

But the truth of the matter is that there is no such potential. Our algorithm turns out to be just as sensitive to a lack of *a priori* knowledge as conventional control is. This in stark constrast with the fact that, in addition to prior knowledge (on which the feedback design is based), our algorithm commands a wealth of data adding on to whatever information available at initialization. So indeed, if not its ability to deal with uncertainty then what is it that makes this an algorithm for Learning Control?

5.2.2. Adaptation, Iteration, and Learning

Next we argue that the problem of Learning Control, the way we see it, is really about uncertainty, and not so much about performance.

Uncertainty: the problem of Learning Control

In Learning Control we seek to enhance a system's performance through a series of (automated) trials. This seems to work out fine as long as we assume to know the system we are working with. But what if this assumption fails: What if we do not know the system, or perhaps only approximately?

At first, we would think that, in that case, iteration would make up for the lack of *a priori* knowledge. But upon reflection, this thought, which struck us as self-evident, proves to make no sense. For how are we to update the current input if we cannot (or can only approximately) predict the effect a change in input will have on the system's output?

Assuming (extreme) uncertainty, the following scenario may help illustrate the issue at hand. We are given control over a certain plant P that we know little or nothing about. As we desire to get to know the system, we decide to set up a number of experiments (trials). In each experiment we apply a certain input u_k and the system returns the corresponding output $y_k = Pu_k$. The data is stored so that, if we so desire, we can use it to steer the next experiment. During the course of these experiments we gather a certain, possibly large, amount of data $\mathcal{D} := \{(u_i, y_i), i = 1, 2, \ldots\}$. But what, exactly, does this data tell us about our system?

That depends on the system itself. In an LTI setting, for instance, it takes just one obversation to characterize the system's complete IO behavior. In this case, the problem of Learning Control degenerates as there is no need for repeated iteration. In general, the data forms a mere subset of the system's (input-output) behavior, the subset being not necessarily respresentative of the total behavior. So when it comes to predicting the system's response to an input that is not in the data set, or, to constructing the input corresponding to a given output, the data might not be of much use. What is lacking is a proper embedding: a *model*. A model may be *fitted* to the data, but that will not resolve the problem. For however well such a model may capture the *observed* behavior, there are is absolute guarantee that it will do equally well on the *total* behavior.

In contrast with e.g. the literature on System Identication and Statistical Learning Theory, the ILC literature contains but little reference to this problem. Its focus seems to lie elsewhere. In fact, the way it is generally conceived, Learning Control is not at all about "turning experience into knowledge" (data into performance). And if not many papers would suggest otherwise, we would not be discussing the issue here. In any case, based on our earlier discussion, we cannot but conclude that in the sense alluded to above, ILC does not 'learn'.

Iteration and Learning

To make this claim more substantial, let us have another look at the update law we have been considering throughout this thesis:

$$u_{k+1} \;=\; Qu_k + Le_k. \tag{5.5}$$

What does this equation represent? Basically, it respresents the law that governs the evolution of the iterative system. It is a rule that tells how to automatically generate a new input from the data collected during past trials. Now suppose this algorithm exhibits some intelligent behavior, then where does it reside? Is it in the (structure of the) rule itself? Or perhaps

in the choice of its parameters, Q and L? A simple analysis reveals that it must be in the latter. For suppose this learning process (the algorithm) converges. Then whatever it is that was learned must show up in the 'fixed point' equation

$$(I - Q + LP)\,\bar{u} \;=\; Ly_{\mathrm{d}}. \tag{5.6}$$

The quantity $\bar{u}$ is something we do not know and generally cannot solve for (as we cannot assume to know P) so if anything was learned then it must be $\bar{u}$. But note that the value of $\bar{u}$ is solely determined by Q, L and P, and not by the evolution of the input $\{u_k\}$. In that sense, it is not surprising that in a causal context we can retrieve the same quantity without a single iteration (without learning?). And to some extent the same conclusions hold in a noncausal context. Learning and iteration are not the same thing, neither does iteration *imply* learning. For if the family of causal iterations is said to implement a learning behavior, then why not conventional feedback control?

Adaptation

The distinction between iteration and learning bears on the question *how* the data is used. Algorithms such as the one above are evidently *data-driven*; the data is used, but not necessarily in an instrumental way. Indeed what our update law lacks is the ability to *interpret* the data. Even as the data invalidates the model on which the design is based, the update law just keeps on updating the input, 'ignorant' of what is actually going on. In that sense, our algorithm does not *adapt*.

Now suppose we would allow the parameters Q and L to depend on the data. Would we then consider our algorithm to be adaptive? That depends. We obviously need a rule to make the dependence explicit. And once we define a rule, we are back at the same problem but now at the level of the update rule that updates the update rule. The recursiveness is evident and in that sense the existence of a truly adaptive, ('self-adaptive') algorithm is doubtful. But if we confine ourselves to the level at which the control input is updated, then yes, we could consider such algorithms to be adaptive. An example will follow.

5.3. On trial-dependent update laws

5.3.1. Motivation

For no particular reason, we have thus far confined our studies to so called *trial-independent* update laws (formally: constant-coefficient recurrence relations). We have seen that such update laws, though data-driven, generally lack the adaptive ability (flexibility) we would expect them to have, given that they are to implement a learning behavior.

As we studied this particular class of algorithms, we learned that performance would always depend on a foursome factors. In no particular order: the plant itself (not the plant *model*), the desired output, and two design parameters. With the first two fixed, and possibly unknown (the plant), this leaves us only two factors, over which we can exercise some control. Yet, with these two we could, in principle, tune our performance variable to whatever we want it to be (no constraints as such apply). However, proper tuning would require detailed knowledge of the plant. Which means that, at the end of the day, we face an ordinary design problem not unlike the sort we find in conventional feedback control. In fact, not unlike that at all (see Chapter 3).

In view of the fact that these parameters are the main determinants for performance it is somewhat odd that both are fixed at initialization and cannot be modified during execution. As a first step towards enhanced performance robustness we could allow these parameters to depend on the actual trial data. Which is precisely what we will do as we next consider a class of trial-dependent update laws.

Though it is unlikely that even such algorithms are truly adaptive in the sense explained above, they form an interesting object of study nonetheless. Besides, they are instrumental in determining what 'learning' is about.

5.3.2. A class of trial-dependent update laws

Part of the results in this section appeared inVerwoerd et al. (2004b). Consider the next class of update laws:

$$u_{k+1} \;=\; Q_{k+1}u_k + L_{k+1}e_k, \qquad k = 0, 1, \ldots \tag{5.7}$$

For all k, we assume both $Q_k : U \mapsto U$ and $L_k : Y \mapsto U$ to be bounded LTI operators, possibly noncausal. We additionally assume the plant $P : U \mapsto Y$ to be stable. So as to familiarize ourselves with this new class of update laws, we first investigate and characterize the (asymptotic) behavior of (5.7) for

any *given* sequence of operator pairs $\{(Q_k, L_k)\}_{k=1}^{\infty}$. We assume that $\{Q_k\}$ and $\{L_k\}$ are *strongly converging*.

Definition 5.3.1 (Strong convergence). *Adapted from Curtain and Zwart (1995). Let $\{F_k, k \geq 1\}$ be a sequence of bounded affine operators (on X). If*

$$\|F_k x - Fx\| \;\; \rightarrow \;\; 0, \tag{5.8}$$

as $k \rightarrow \infty$ for all $x \in X$ then we say that F_k converges strongly *to F.*

Define $F_k(u) := (Q_k - L_k P)\, u + L_k y_{\mathrm{d}}$ and rewrite (5.7) as

$$u_{k+1} \;\; = \;\; F_{k+1}(u_k). \tag{5.9}$$

Our next objective is to find conditions on $\{F_k\}$ under which the sequence $\{u_k\}$, induced by (5.9), converges to a unique 'fixed point' $\bar{u}$, independent of u_0. It is evident that classical fixed point theory will not help us here. Fortunately, the classical theory is readily extended to accommodate statements about strongly converging sequences of operators. For notational convenience we adopt the shorthand notation

$$\left(\prod_{i=1}^{k} F_i\right)(x) \;\; := \;\; F_k\Big(F_{k-1}\big(\cdots F_1(x)\cdots\big)\Big).$$

We need the following definitions:

Definition 5.3.2 (Fixed point). *Let $\{F_k, k \geq 1\}$ be a sequence of bounded affine operators that strongly converges to a limit $F := \lim_{k\rightarrow\infty} F_k$ (see Definition 5.3.1). We say that a point $x \in X$ is a fixed point of the sequence of operators $\{F_k\}$ if it is a fixed point of F, i.e. if $\lim_{k\rightarrow\infty} \|F_k(x) - x\| = 0$.*

Definition 5.3.3 (Contraction). *Let $\{F_k, k \geq 1\}$ be a strongly converging sequence of bounded affine operators, and let λ_k denote the Lipschitz constant corresponding to F_k. We say that $\{F_k\}$ is* contractive *if $\limsup_k \lambda_k < 1$.*

Note that in case F_k is constant (as a function of the trial number k), Definition 5.3.3 coincides with that given in Section 2.1.4. We have the following result.

Theorem 5.3.4. *Let X be a Banach space and let $\{F_k, k \geq 1\}$ be a strongly converging sequence of bounded affine operators on X. Let $F := \lim_{k\rightarrow\infty} F_k$.*

Suppose $\{F_k\}$ is contractive. Then there exists a unique fixed point $\bar{x} \in X$ such that $F(\bar{x}) = \bar{x}$. Furthermore, for all $x \in X$, we have that

$$\bar{x} \;=\; \lim_{k\to\infty} \left(\prod_{i=1}^{k} F_i \right)(x). \tag{5.10}$$

A proof of Theorem 5.3.4 can be found in Section 5.A. Later we show how to apply this theorem to prove convergence for a class of adaptive update laws in which Q_k and L_k are not *given*, but depend on the data instead. For now let us just assume that $\{Q_k\}$ and $\{L_k\}$ are strongly converging sequences of operators. Define $Q := \lim_{k\to\infty} Q_k$ and $L := \lim_{k\to\infty} L_k$ and assume that the conditions of Theorem 5.3.4 are satisfied. Then the iteration $u_{k+1} := F_{k+1}(u_k)$, with $F_k : U \mapsto U$ defined as

$$F_k(u) \;:=\; (Q_k - L_k P)\, u + L_k y_{\mathrm{d}}, \tag{5.11}$$

converges to a unique fixed point $\bar{u} \in U$ which, independent of u_0 is given as

$$\begin{aligned} \bar{u} \;&=\; \lim_{k\to\infty} \left(\prod_{i=1}^{k} F_i \right)(u_0) \\ &=\; \lim_{k\to\infty} \left((Q_k - L_k P) \left(\prod_{i=1}^{k-1} F_i \right)(u_0) + L_k y_{\mathrm{d}} \right) \\ &=\; (Q - LP)\, \bar{u} + L y_{\mathrm{d}}. \end{aligned} \tag{5.12}$$

It is worth noting that, had we started from the update rule

$$u_{k+1} \;=\; Q u_k + L e_k, \tag{5.13}$$

we would obtain the exact same fixed point equation. This proves that, in an asymptotic sense, update laws (5.7) and (5.13) are equivalent. The importance of this observation is tied up with the fact that $\{Q_k\}$ and $\{L_k\}$, and in particular their respective limits Q and L, are *known*, or at least computable on the basis of *a priori* knowledge. Would we consider update laws in which, for instance, the operator L_{k+1} depends on the current error e_k, an equivalent (feedback) scheme could still exist, but in that case, we would not be able to determine the value of $L := \lim_{k\to\infty} L_k$ (and thus that of the equivalent controller), without actually running the scheme.

5.3.3. Non-contractive, trial-dependent update laws

In the next section we will consider two examples in which Q_k and L_k both depend on trial data. Right now, there is a specific subset of (5.7) that calls for our attention:

$$u_{k+1} \;=\; u_k + L_{k+1}e_k. \tag{5.14}$$

We assume that each of the operators $L_k : Y \to U$, $k = 1, 2, \ldots$ is bounded and that the sequence $\{L_k\}$ strongly converges to zero as k tends to infinity.

Motivation

The reason why consider this specific set is this: of all iterations in the larger set defined by (5.7), this subset contains precisely those update laws that are not trivially equivalent with an update law of the form (5.13). We choose to treat this set seperately, for would we embed it in a general discussion, its properties would not come out well.

Besides, we do not wish to be constrained by the conditions of Theorem 5.3.4. These conditions are general, but restrictive; they are not likely to be met in case $Q_k \equiv I$, as in (5.14), or even when $\lim_{k\to\infty} Q_k = I$. This is easy to see; Rewrite (5.14) to obtain

$$\begin{aligned} u_{k+1} &= F_{k+1}(u_k) \\ &= (I - L_{k+1}P)\, u_k + L_{k+1}y_{\mathrm{d}}, \end{aligned} \tag{5.15}$$

and note that $\{F_k, k \geq 1\}$ is contractive if and only if the condition

$$\|I - L_{k+1}P\| \;<\; 1 \tag{5.16}$$

is satisfied for (almost) all k. That is to say, only if LP is invertible over U, where L is defined as the limit $L := \lim_{k\to\infty} L_k$. There is no reason to assume that this condition holds for general L nor that there even exists L for which it could hold.

By restricting attention to the class of update laws (5.14), specialized techniques can be deployed that do not rely on the contractivity of the transition map. But this comes at a price. In case of (5.14) with the conditions as stated, one readily observes that as L_k tends to zero, F_k will tend to identity. Since every point in the domain of an identity map is a fixed point, it is clear that uniqueness of the fixed point is generally lost.

Convergence analysis

Next a theorem is presented, which states conditions under which the class of update rules (5.14) converges to a bounded solution. It requires the notion of summable sequences.

Definition 5.3.5 (Summable sequence). *Let $\{a_1, a_2, \ldots\}$ be a sequence of positive real numbers. The sequence is said to be* summable *if*

$$\lim_{N\to\infty} \sum_{k=1}^{N} a_k \quad < \quad \infty \tag{5.17}$$

We have the following theorem.

Theorem 5.3.6 (Convergence on U). *Let $\{L_k\}$ be a sequence of bounded linear operators from Y to U. Suppose $\{\|L_k\|\}$ is summable. Then the sequence of control inputs $\{u_k\}$ converges to some bounded input $\bar{u} \in U$.*

A proof of Theorem 5.3.6 can be found in Section 4.A. Note that the assumption that L_k strongly converges to zero is implicit in the condition that $\{\|L_k\|\}$ is summable.

Application

The next example shows how the result of Theorem 5.3.6 may be applied.

Example 5.3.7. *Let $U = Y = \mathcal{H}_2$ and suppose $P \in \mathcal{RH}_\infty$. For $k \geq 1$ we define $L_k := L/k^2$. Assume $L \in \mathcal{RH}_\infty$. It is easy to see that $\{\|L_k\|\}$ is summable:*

$$\sum_{k=1}^{\infty} \|L_k\| \quad = \quad \sum_{k=1}^{\infty} \|L\|/k^2 \quad = \quad (\pi^2/6)\|L\| \quad < \quad \infty. \tag{5.18}$$

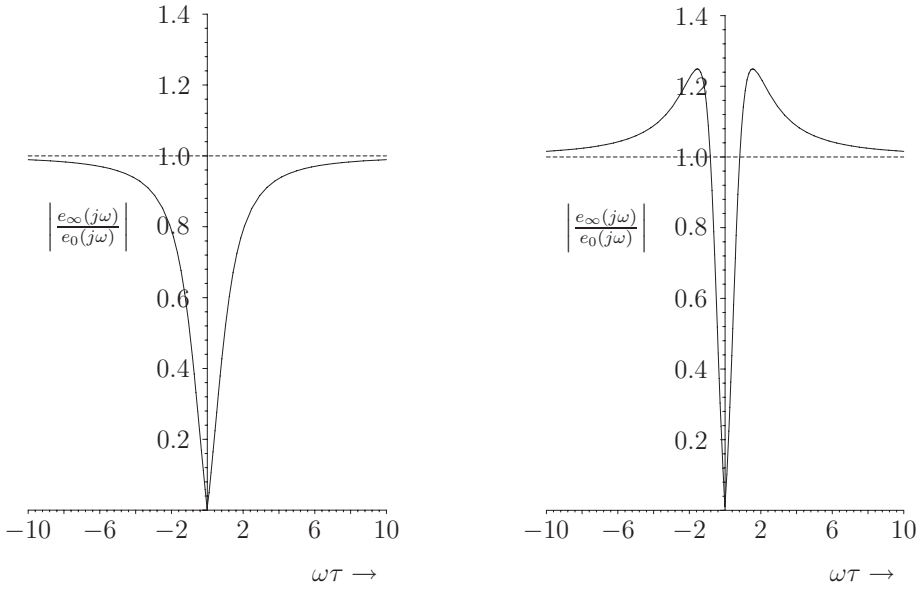
Theorem 5.3.6 says that, provided L and P are both bounded, the sequence of inputs $\{u_k\}$ converges to a bounded solution in $\mathcal{H}_2$. This very fact implies that convergence does not depend on detailed knowledge of the plant. The error e_k satisfies $e_k = Z_k e_0$, where $Z_k : Y \mapsto Y$ is defined as $Z_k := \prod_{i=0}^{k-1} (I - L_{i+1}P)$. We define the map $Z : Y \mapsto Y$,

$$Z \quad := \quad \lim_{k\to\infty} Z_k. \tag{5.19}$$

This map relates the final error, e_∞, to the initial error, e_0. In some cases, a closed-form expression for Z can be found. As it turns out, in our example

$$Z(s) \quad = \quad \frac{\sin\left(\pi\sqrt{L(s)P(s)}\right)}{\pi\sqrt{L(s)P(s)}}. \tag{5.20}$$

A plot of $Z(j\omega)$ for $P(s) = 1/(s\tau + 1)$ and $L = 1$ is given in Figure 5.1a.



(a) $P = 1/(s+1)$ (b) $P = 1/(s+1)^2$

Figure 5.1.: The function $|Z(j\omega)|$ (5.20) vs. $\omega\tau$. The dashed line indicates the 0 dB level. These plots correspond to the parameter value $L = 1$.

In our example, the initial error is significantly attenuated in the low-frequency range (up till $\omega\tau \approx 1$) and never amplified ($|Z(j\omega)| < 1$ for all ω). One may wonder whether this property is intrinsic to the class of update rules. It is not, as can be seen from the case $P := 1/(s\tau+1)^2$, depicted in Figure 5.1b. More generally, the following proposition shows that $|Z(j\omega)|$ is constrained by some integral relation. That is, *if* we insist on causality.

Theorem 5.3.8 (Sensitivity integral). *Assume that $P \in \mathcal{RH}_\infty$ has a pole-zero excess greater than one and let $\{L_k\}$ be a summable sequence in $\mathcal{RH}_\infty$. Then $Z(s)$ as defined in (5.19) satisfies the integral relation*

$$\int_{-\infty}^{\infty} \log|Z(j\omega)|d\omega \;\geq\; 0. \tag{5.21}$$

Proof. Theorem 5.3.8 is a direct consequence of Bode's integral relation. From (5.19) we have

$$\int_{-\infty}^{\infty} \log|Z(j\omega)|\mathrm{d}\omega \;=\; \sum_k \int_{-\infty}^{\infty} \log|1 - L_k(j\omega)P(j\omega)|\mathrm{d}\omega. \tag{5.22}$$

The result now follows from Lemma 4.2.2 which says that under the conditions stated (and for all k),

$$\int_{-\infty}^{\infty} \log|1 - L_k(j\omega)P(j\omega)|\mathrm{d}\omega \;\geq\; 0. \tag{5.23}$$

This concludes the proof. ∎

The result expressed by Theorem 5.3.8 is sometimes referred to as the *waterbed effect*. What it says is that if the (initial) error is attenuated in one frequency region, it is amplified in another. It particular, the error can never be zero over some interval with nonzero measure.

Again, these limitations may be overcome by dropping the causality constraint. To see this, let us consider the update law

$$u_{k+1} \;=\; u_k + \gamma_k P^* e_k. \tag{5.24}$$

We assume that $0 < \gamma_k < 2/\|P\|^2$ for all k and $\lim_{k\to\infty} \gamma_k = 0$. If $\sum_k \gamma_k$ is finite then it follows from Theorem 5.3.6 that the sequence $\{u_k\}$ converges to a bounded solution $\bar{u}$. In this case, the map $Z(j\omega)$ satisfies

$$Z(j\omega) \;=\; \frac{\sin(\pi\left|P(j\omega)\right|)}{\pi\left|P(j\omega)\right|}. \tag{5.25}$$

Note that $\sin(|x|)/|x|$ is less than unity for all $x \neq 0$; Hence $|Z(j\omega)| < 1$ for (almost) all ω. And this is true for all $P \in \mathcal{RH}_\infty$ (except $P = 0$, in which case $Z = 1$).

5.3.4. Adaptive update schemes

In closing we present two algorithms that bring about a certain adaptive behavior. The first is based on the class of causal iterations we studied in Chapter 3. The second relates to a class of noncausal iterations.

Gain adaptation in causal learning rules

Consider the next class of update laws:

$$u_{k+1} \;=\; \gamma\left(Qu_k + Le_k\right). \tag{5.26}$$

We assume that the parameter γ is nominally set to 1 and that $Q, L \in \mathcal{RH}_\infty$ are given. Now as we run this algorithm it may turn out that it does not converge. If we fix γ, there is nothing we can do about this. But if we allow γ to 'adapt' we may enforce convergence in a fairly simple way. The

idea is as follows. First we note that for γ 'small enough' the algorithm will surely converge. That is to say, there exists $\bar{\gamma} > 0$ such that for all $\gamma < \bar{\gamma}$ the sequence $\{u_k\}$ is convergent. After each cycle, we test for convergence, i.e. we check whether $\|u_{k+1} - u_k\| < \rho\|u_k - (\gamma_{k-1}/\gamma_k)u_{k-1}\|$ for some fixed $\rho < 1$—more on this shortly—and decrease γ if necessary. Now consider the following algorithm:

Algorithm 5.3.9 (Causal update law, gain adaptation).

1. *Set* $\gamma_0 := 1$.
2. *Select an initial input* u_0.
3. *Set* $k := 0$.
4. *Apply input* u_k, *record the output* y_k.
5. *Compute a new input.*

$$u_{k+1} := \gamma_k (Qu_k + Le_k) \tag{5.27}$$

6. *Update* γ *according to the following rule* ($k \geq 1$).

$$\gamma_{k+1} := \begin{cases} \gamma_k & \text{if } \|u_{k+1} - u_k\| < \rho\|u_k - \frac{\gamma_{k-1}}{\gamma_k}u_{k-1}\|, \\ \gamma_k \left(\frac{\|u_k - \frac{\gamma_{k-1}}{\gamma_k}u_{k-1}\|}{\|u_{k+1} - u_k\|} \right) \frac{\rho}{1+\delta} & \text{otherwise.} \end{cases}$$

(For $k = 0$ *take* $\gamma_{k+1} := \gamma_k$, *i.e.* $\gamma_1 := \gamma_0$.*)*

7. *Set* $k := k + 1$. *Repeat from 4.*

The idea behind the Step 6. is as follows. We fix ρ to a value less than unity and we say that the map $\gamma_k (Q - LP)$ is *sufficiently contractive* if its Lipschitz constant does not exceed ρ. Suppose this is true for a certain k. Then we have the following inequality.

$$\begin{aligned} \|u_{k+1} - u_k\| &= \|\gamma_k (Q - LP) \left(u_k - \frac{\gamma_{k-1}}{\gamma_k} u_{k-1} \right) \| \\ &\leq \rho \| \left(u_k - \frac{\gamma_{k-1}}{\gamma_k} u_{k-1} \right) \|. \end{aligned} \tag{5.28}$$

If (5.28) is *not* satisfied, it follows that $\gamma_k(Q - LP)$ is not sufficiently contractive. In that case we decrease γ_k by a certain factor, defined as the ratio

between the actual 'gain' and the maximum admissible gain, ρ, times some additional term $(1+\delta)$ (for any $\delta > 0$, this construction ensures that the sequence $\{\gamma_k\}$ can reach the set $[0, \bar{\gamma}]$ within a finite number of steps). In case inequality (5.28) *is* satisfied, γ_k may or may not be small enough and we leave it unchanged.

For all $Q, L \in \mathcal{RH}_\infty$ and all γ_0, Algorithm 5.3.9 converges to a bounded solution $\bar{u}$. To see this, note that Condition (5.28) can only be violated a finite number of times, since for all $\delta > 0$ and all $\bar{\gamma} > 0$ there exists $N < \infty$ such that $\gamma_0 \left(1/(1+\delta)\right)^N < \bar{\gamma}$. The result thus basically follows from Theorem 5.3.4 which says that if we have a strongly converging sequence of operators almost all of which are contractive then the iteration $u_{k+1} = F_{k+1}(u_k)$ converges to a unique fixed point. It is true that for Condition (5.28) to hold, the map $F_k : U \mapsto U;\ F_k(u) = \gamma_{k-1}\left(Qu + L(y_\mathrm{d} - Pu)\right)$ need not be contractive, so strictly speaking Theorem 5.3.4 does not apply. But for our argument this does not matter. Indeed, all we need is that the next inequality is satisfied for *almost* all k:

$$\|u_{k+1} - u_k\| \ \leq \ \rho \|u_k - u_{k-1}\|, \tag{5.29}$$

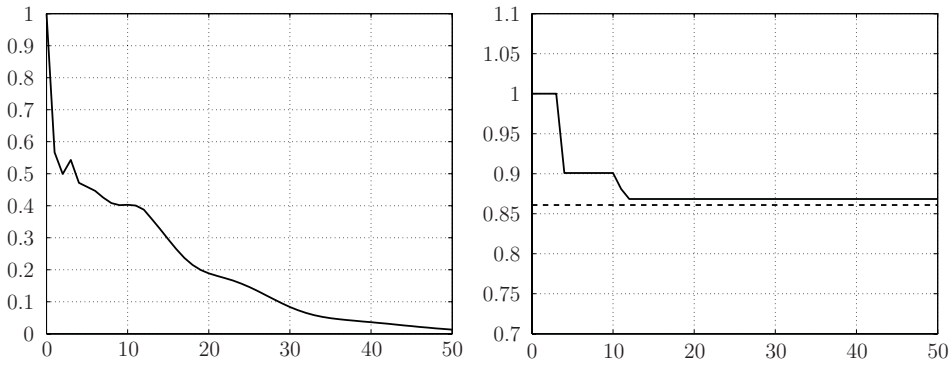
and this is true by construction.

Example 5.3.10. *Let P be given as*

$$P(s) \ := \ \frac{1}{(s+1)^2} \tag{5.30}$$

We select $Q = 1$, and $L = (s+1)^2/(0.1s+1)^2$. For the nominal update law to be convergent, the map $1 - LP$ must be contractive. This is not the case; In fact, $\|1 - LP\|_\infty \approx 1.15$. We select $\gamma_0 := 1, \rho := 0.99, \delta := 0.01$ and apply Algorithm 5.3.9. Figure 5.2a shows the evolution of the norm of the increment $\Delta u_k := u_{k+1} - u_k$, normalized with respect to $\|\Delta u_0\|$. Initially the increment falls off rapidly, but then in the third trial, we observe an increase. As a result, γ is decreased—See Figure 5.2b. In the tenth trial we observe a similar phenomenon; the increment's increase is counteracted by a decrease in γ. From that point on $\|\Delta u_k\|$ decreases monotonically and γ is kept constant as expected. We remark that in this example γ never actually reaches the "safe" value $\rho/\|1 - LP\|_\infty$ (the dashed line in Figure 5.2)—at least not within the first 50 trials.

The strength of Algorithm 5.3.9 lies in the fact that it eliminates the risk of compromising the stability robustness, thus allowing for a less cautious design. At the same time it does not guarantee good performance. As a matter of fact, it does not resolve the performance issue at all, since it



(a) $\|\Delta u\|/\|\Delta u_0\|$ (b) γ_k

Figure 5.2.: Simulation results—see Example 5.3.10 for details. The figures on the left and the right respectively show (a) the evolution of the increment $\|\Delta u_k\|$ and (b) the gain γ_k, both as a function of the trial number k.

assumes Q and L—the main determinants— to be given. In addition, the present algorithm is not equipped to cope with noise.

Surely we could improve on its design; based on Algorithm 5.3.9 we could come up with a dozen other update schemes, each exploiting the particular features we would assume the plant to have. But we believe that strategy would not take us very far. What we would find is that the complexity of the resulting algorithm is inversely proportional to what we assume to know about the plant. But the question is where this complexity originates: in the problem, or in the solution? If it is, like we think, in the solution, then it may pay off to look for other ways of exploiting the same information.

Learning Control based on single parameter adaptation—a noncausal example

Let us consider the next algorithm. Starting point is the update law given in Equation (5.31).

$$u_{k+1} \;=\; u_k + \lambda P^* e_k. \tag{5.31}$$

It seems evident that, in order to implement this scheme, we need to know P. But that is not entirely true. For what we need to know is P^*e_k, not P. The difference is important, as we may evaluate P^*e_k even if we do not know P. This is by virtue of a result from Chapter 4 which says that

$$P^*e_k \;=\; \Pi P \Pi e_k. \tag{5.32}$$

Recall that the operator Π is defined as $(\Pi e)(t) := e(T-t)$. The conclusion is that even if we do not know P, we can compute P^*e_k from the system's response to the auxiliary input Πe_k. The drawback of this approach is that we need to double the number of iteration steps. From (5.31) it follows that the error e_k satisfies the update equation

$$e_{k+1} \;=\; (1-\lambda P^*P)\, e_k \tag{5.33}$$

For convergence we require

$$\left|1-\gamma P(j\omega)^*P(j\omega)\right| \;<\; 1 \quad \text{for almost all } \omega \tag{5.34}$$

We may observe that for λ small enough this condition is always satisfied. In particular, λ should be smaller than $2/\left\|P\right\|_\infty^2$. Hence, if we have a lower bound on $\|P\|_\infty$ we can easily pick λ accordingly. But even if we do not, we can make the algorithm converge by setting up an adaptive scheme as follows.

Algorithm 5.3.11 (Noncausal update law, single parameter adaptation).

1. *Pick some* $\gamma_0 > 0$.
2. *Choose an initial input* u_0.
3. *Set* $k := 0$.
4. *Apply the input* u_k *to the system, record the output* $y_k := Pu_k$.
5. *Compute the error* e_k *as* $e_k := y_d - y_k$.
6. *Compute* Πe_k *as* $(\Pi e_k)(t) := e_k(T-t)$.
7. *Apply the auxilary input* Πe_k *to the system, record the output* $P\Pi e_k$.
8. *Compute* $P^*e_k := \Pi P \Pi e_k$.
9. *Compute a new input* $u_{k+1} = u_k + \lambda_k P^* e_k$.
10. *Update* λ_k *according to the following rule.*

$$\lambda_{k+1} \;:=\; \begin{cases} \lambda_k & \text{if } \|e_k\| < \|e_{k-1}\|, \\ \lambda_k \left(\frac{\|e_{k-1}\|}{\|e_{k-1}\|+\|e_k\|}\right)^{\frac{1}{1+\delta}} & \text{otherwise.} \end{cases} \tag{5.35}$$

11. Set $k := k + 1$. Repeat from 4.

The idea behind this update scheme is as follows. If the norm of the error in the current trial is greater than or equal to that in the previous trial, then apparently λ is not small enough; In that case we decrease λ as indicated. If not, we leave λ unchanged. This way we ensure that λ_k is a nonincreasing function of k. By construction the error grows a finite number of times at most. Consequently, the sequence $\{\|e_k\|\}$ is bounded and converges to a limit. However, convergence may be arbirarily slow.

Example 5.3.12. *We apply the algorithm in closed loop. That is, we consider the feedback interconnection of a plant P_{open} with a compensator C and define $P : U \mapsto Y,\ P(u) := (1 + CP_{open})^{-1} CP_{open}u$. Let P_{open} and C be given as*

$$P_{open}(s) \quad := \quad \frac{1}{37s^2 + 10} \quad ; \quad C(s) \quad := \quad 275280\frac{0.02s + 1}{0.002s + 1}. \tag{5.36}$$

We select the reference trajectory y_d to equal the unit step response of the system

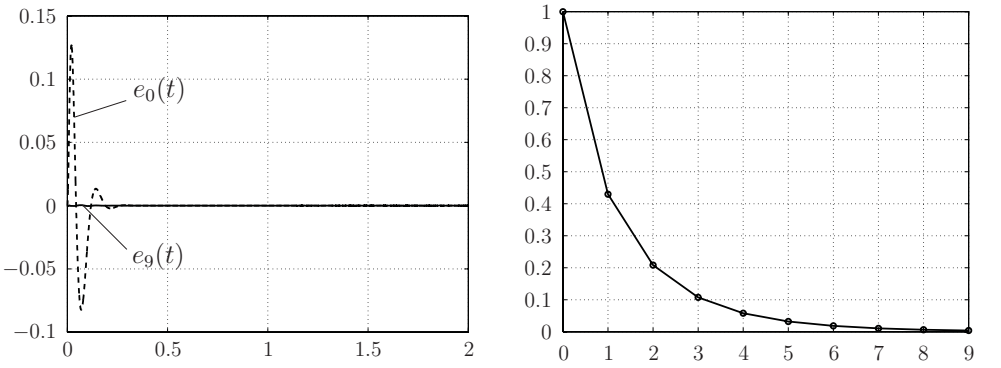
$$P_{ref}(s) \quad := \quad \frac{1}{(0.02s + 1)^2} \tag{5.37}$$

We choose $\lambda_0 := 1, \delta := 0.01$ and $u_0 := y_d$. Figure 5.3a shows the error in trials 0 (just feedback) and 9. Figure 5.3b shows the normalized mean squared error $\|e_k\|/\|e_0\|$ as a function of the trial number k. We observe that the error decreases exponentially as a function of the trial number. The monotonicity is due to the fact that we chose λ_0 small enough (smaller than $2/\|P\|_\infty^2 \approx 1.22$). Figure 5.4a shows what happens if we start from $\lambda_0 = 2$. We observe that during the first few trials, the norm of the error grows. Meanwhile—see Figure 5.4b—λ is decreased and at $k = 3$, it reaches a value of 0.7, which is well below the critical value 1.22. From that moment on the error decreases monotonically.

Algorithm 5.3.11 provides us with a means to get good tracking performance in the face of extreme (parametric) uncertainty. Its major weakness, as said, is that no upper bound on the speed of convergence can be given. The problem is caused by the zeroes of P. Assuming we have some nominal model P_0, we could replace λ by a frequency weighted gain $\lambda\left(\gamma^2 + P_0^* P_0\right)^{-1}$—see Equation (5.38).

$$u_{k+1} \quad = \quad u_k + \lambda_k \left(\gamma^2 + P_0^* P_0\right)^{-1} P^* e_k \tag{5.38}$$

For γ small enough this could speed up convergence, provided P is close enough to P_0.



(a) $\{e_k(t) : k = 0, 9\}$ (b) The error e_k (normalized RMS)

Figure 5.3.: Simulation results corresponding to Algorithm 5.3.11 (see Example 5.3.12 for details). The plot on the left shows the error during trials 0 (dashed) and 9 (solid); The plot on the right depicts the normalized mean-squared error for trials 0 through 9. The parameter λ_0 was set to 1.

5.4. Discussion

5.4.1. Brief review

In this chapter, we looked at a class of trial-dependent update rules of the form

$$u_{k+1} \quad := \quad Q_{k+1}u_k + L_{k+1}e_k. \tag{5.39}$$

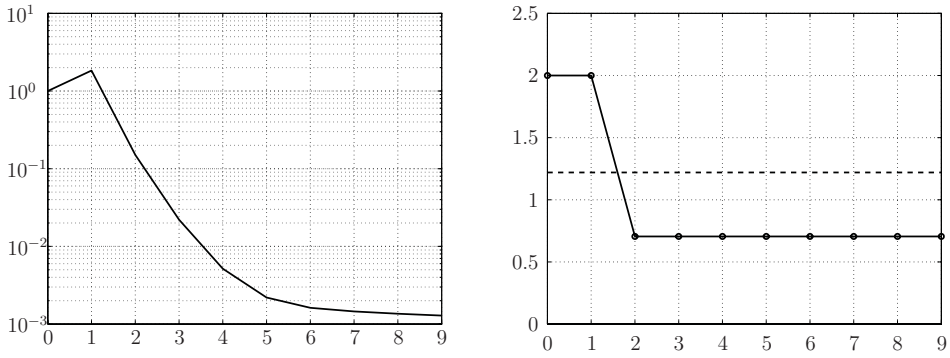
Initially, we assumed $\{(Q_k, L_k)\}$ to be *given* and strongly converging. When we additionally assumed the sequence of operators to be *contractive*, we could prove this class to be equivalent with a class of trial-independent update laws,

$$u_{k+1} \quad := \quad Qu_k + Le_k. \tag{5.40}$$

We also looked at a particular subset of *non-contractive* update rules,

$$u_{k+1} \quad := \quad u_k + L_{k+1}e_k \tag{5.41}$$

where we assumed $\{L_k\}$ to strongly converge to zero. We found this class to converge under the relatively mild condition that $\{\|L_k\|\}$ be summable. We studied the achievable performance and found it to be constrained, much like it is in the general class. Besides, we found that performance would



(a) The error e_k (normalized RMS) (b) λ_k for $k = 0, 1, \ldots, 9$.

Figure 5.4.: This figure relates to Algorithm 5.3.11 and Figure 5.3. Shown is the effect of choosing λ_0 too large ($\lambda_0 = 2$). During the first two trials ($k = 0, 1$) the error grows as may be obverved from the plot on the left. As a result, λ is decreased until below the critical value (the dashed line in the figure on the right). From that moment on, the error decreases monotonically.

depend on the initial condition, which made it less generic, and hence less attractive.

Finally, we considered two examples of trial-dependent update laws in which we did allow $\{(Q_k, L_k)\}$ to depend on trial data.

5.4.2. Conclusion

What can we conclude from our analysis? First of all, it reveals that, performancewise, there is no reason to prefer trial-dependent laws over trial-independent ones. Both of them (and that includes the aforementioned non-contractive update rules) subject to the same constraints.

But then, it was not performance that led us to consider trial-dependent laws. For that we could easily attain with (see Lemma 4.2.4) trial-independent update rules. What made us decide to look at such update laws was the hope they could help us with the uncertainty issue. Earlier we had concluded that trial-independent update laws did not and could not warrant a robust performance. And that mainly because of their static structure.

We figured that if we had this structure removed and replaced it with a dynamic equivalent, we would be able to accommodate some kind of

adaptation. Example 5.3.10 lacked many desirable properties, and probably would not work very well in a noisy situation; it sufficed to show that by adapting a single gain parameter, it is possible to have a competitive design without compromising stability robustness.

Perhaps the second example was more interesting, particularly from a performance point of view. For whereas the first example did not show particularly good behavior, this one did. To say, the algorithm we developed would converge to zero error on the basis of but a single assumption: that the plant be LTI.

Of course, we used the insight we had obtained in studying trial-independent update laws; the basic structure was given. The trick was to make use what of what did know (the fact that the system is LTI) and to iterate on what we did *not* know (the system gain). This combination turned out to be very effective.

Somehow, this seems exemplary to what Learning Control is about, namely: to combine what is known (prior knowledge) with what is learned during iteration (trial data). The algorithms we discussed are not likely to work well in practice for reasons we have not considered. Even so the underlying philosophy may have some potential.

5.A. Proofs

Proof of Theorem 5.3.4

Proof. First we prove uniqueness. Let $\{F_i\}$ be contractive and suppose F has two fixed points $x, x' \in X$ with $x \neq x'$. Let λ_i denote the Lipschitz constant of the linear operator $(F_i)_\emptyset$ and define $\lambda := \limsup_i \lambda_i$. For all i we have

$$\|F_i(x) - F_i(x')\| \;\leq\; \lambda_i \|x - x'\|. \tag{5.42}$$

In particular,

$$\lim_{i\to\infty} \|F_i(x) - F_i(x')\| \quad\leq\quad \lambda\, \|x - x'\| \quad<\quad \|x - x'\|. \tag{5.43}$$

Here we used the fact $\lambda < 1$ (which follows from the assumption of contractivity). On the other hand we have

$$\|F_i(x) - F_i(x') - (x - x')\| \;\leq\; \|F_i(x) - x\| + \|F_i(x') - x'\|, \tag{5.44}$$

which implies that

$$\lim_{i\to\infty} \|F_i(x) - F_i(x')\| \;=\; \|x - x'\|. \tag{5.45}$$

Equations (5.43) and (5.45) cannot both hold true, unless $x = x'$. But this contradicts our starting assumption. Hence we conclude that F has only one fixed point.

Next we prove that if $\bar{x} := \lim_{n\to\infty} \left(\prod_{i=1}^{n} F_i\right)$ is well defined (in the sense that the limit exists), $\bar{x}$ must be a fixed point of F. Indeed, suppose $\lim_{n\to\infty} \left(\prod_{i=1}^{n} F_i\right)$ exists. Then, by definition

$$\lim_{n\to\infty} \left(\prod_{i=1}^{n} F_i\right) \;=\; \lim_{n\to\infty} F_n \left(\prod_{i=1}^{n-1} F_i\right). \tag{5.46}$$

By assumption, the left hand side of (5.46) converges to $\bar{x}$. The claim is that the right hand side converges to $F(\bar{x})$. This can be shown as follows. Take any $x_0 \in X$, define $x_n := \left(\prod_{i=1}^{n} F_i\right)(x_0)$, and observe that

$$\begin{aligned} \|F_n(x_{n-1}) - F(\bar{x})\| &= \|F_n(x_{n-1}) - F_n(\bar{x}) + F_n(\bar{x}) - F(\bar{x})\| \\ &\leq \|F_n(x_{n-1}) - F_n(\bar{x})\| \;+\; \|F_n(\bar{x}) - F(\bar{x})\| \\ &= \|F_n(x_{n-1} - \bar{x})\| \;+\; \|F_n(\bar{x}) - F(\bar{x})\| \\ &\leq \lambda_n \|x_{n-1} - \bar{x}\| \;+\; \|F_n(\bar{x}) - F(\bar{x})\|. \end{aligned} \tag{5.47}$$

Here we used the fact that for each n, F_n is a bounded affine operator (with Lipschitz constant λ_n). As n tends to infinity both terms on the right hand side of (5.47) vanish. The first because, by assumption, x_{n-1} converges to $\bar{x}$ and $\{\lambda_n\}$ is uniformly bounded; The second because $\{F_n\}$ is strongly converging. This shows that $F_n(x_{n-1})$ tends to $F(\bar{x})$. We conclude that $\bar{x}$ is a fixed point of F.

Next we show that the sequence $\{\left(\prod_{i=1}^{n} F_i\right)(x)\}_{n=1}^{\infty}$ does indeed converge for all $x \in X$. To that end, take any $x_0 \in X$ and define x_n as before. Let N denote the smallest integer for which $\sup_{i>N} \lambda_i < 1$. For all $n > N$ we have

$$\left(\prod_{i=1}^{n} F_i\right)(x_0) \;=\; \left(\prod_{i=N+1}^{n} F_i\right)(x_N) \tag{5.48}$$

For $i \geq 1$ we define $\tilde{F}_i := F_{i+N}$. To prove the statement, we show that for all $x \in X$, the term $\tilde{x}_n := \left(\prod_{i=1}^{n} \tilde{F}_i\right)(x)$ tends to a limit.

Decompose the operator $\tilde{F}_n$ into an affine part $(\tilde{F}_n)_a$, and a linear part $(\tilde{F}_n)_\emptyset$:

$$\tilde{F}_n(x) := (\tilde{F}_n)_\emptyset(x) + (\tilde{F}_n)_a(x) \tag{5.49}$$

For all $n \geq 1$ we have

$$\tilde{x}_n \;=\; \left(\prod_{i=1}^{n} (\tilde{F}_i)_\emptyset\right)(x) + \sum_{j=1}^{n-1} \left(\prod_{i=j}^{n-1} (\tilde{F}_{i+1})_\emptyset\right)(\tilde{F}_j)_a(x) + (\tilde{F}_n)_a(x)$$

Define $A := \sup_n \|(\tilde{F}_n)_a\|$ and recall that $\|(\tilde{F}_n)_\emptyset\| = \lambda_{n+N}$ is uniformly bounded by some constant $\lambda < 1$. Thus

$$\begin{aligned} \|\tilde{x}_n\| &\leq \lambda^n \|x_0\| + \left(\sum_{j=0}^{n-1} \lambda^j\right) A \\ &= \lambda^n \|x_0\| + \left(\frac{1-\lambda^n}{1-\lambda}\right) A \end{aligned} \tag{5.50}$$

Clearly the right hand side of (5.50) is bounded and hence $\{\tilde{x}_n\}$ is bounded. What remains to be shown is that $\|\tilde{x}_n - \tilde{x}_{n-1}\|$ tends to zero as $n \to \infty$. We have that

$$\begin{aligned} \|\tilde{x}_n - \tilde{x}_{n-1}\| &= \|\tilde{F}_n(\tilde{x}_{n-1}) - \tilde{F}_n(\tilde{x}_{n-2}) + \tilde{F}_n(\tilde{x}_{n-2}) - \tilde{F}_{n-1}(\tilde{x}_{n-2})\| \\ &\leq \|\tilde{F}_n(\tilde{x}_{n-1}) - \tilde{F}_n(\tilde{x}_{n-2})\| + \|\tilde{F}_n(\tilde{x}_{n-2}) - \tilde{F}_{n-1}(\tilde{x}_{n-2})\| \\ &\leq \lambda \|\tilde{x}_{n-1} - \tilde{x}_{n-2}\| + \|\tilde{F}_n(\tilde{x}_{n-2}) - \tilde{F}_{n-1}(\tilde{x}_{n-2})\| \end{aligned}$$

The second term on the right hand side vanishes as n tends to infinity. This is because $\{\tilde{F}_n(\tilde{x}), n \geq 1\}$ is Cauchy for all $\tilde{x} \in X$. Consequently, since $\lambda < 1$, $\|\tilde{x}_n - \tilde{x}_{n-1}\|$ converges to zero. This concludes the proof. ∎

Proof of Theorem 5.3.6

We need the following lemma.

Lemma 5.A.1 (Convergence of an infinite product). *If $a_i \geq 0$ for all values of i, the product $\prod_{i=1}^k (1 + a_i)$ and the series $\sum_{i=1}^k a_i$ converge or diverge together.*

Proof. See Titchmarsh (1939) ∎

We prove the main theorem.

Proof. (of Theorem 5.3.6) Using induction it can be shown that for $k \geq 0$

$$u_k = \prod_{i=0}^{k-1} (I - L_{i+1}P)\, u_0 + \sum_{j=0}^{k-1} \prod_{i=j+1}^{k-1} (I - L_{i+1}P)\, L_{j+1} y_{\mathrm{d}}.$$

Boundedness is proved term by term, starting from the first:

$$\begin{aligned} \|\prod_{i=0}^{k} (I - L_{i+1}P)\, u_0\| &\leq \prod_{i=0}^{k} \|I - L_{i+1}P\|\ \|u_0\| \\ &\leq \prod_{i=0}^{k} (1 + \|L_{i+1}\|\|P\|)\ \|u_0\|. \end{aligned}$$

Boundedness follows from the summability assumption and Lemma 5.A.1. The same argument applies to the second term,

$$\begin{aligned}
&\sum_{j=0}^{k} \prod_{i=j+1}^{k} (I - L_{i+1}P)\ L_{j+1}\, y_{\mathrm{d}} \\
&\quad \leq \sum_{j=0}^{k} \prod_{i=j+1}^{k} (1 + \|L_{i+1}\|\,\|P\|)\ \|L_{j+1}\|\,\|y_{\mathrm{d}}\| \\
&\quad \leq \prod_{i=0}^{\infty} (1 + \|L_{i+1}\|\,\|P\|) \sum_{j=0}^{k} \|L_{j+1}\|\,\|y_{\mathrm{d}}\|, \qquad (5.51)
\end{aligned}$$

where both the sum and the product converge by assumption of summability (and Lemma 5.A.1). This concludes the proof. ∎

6

Discussion and Conclusion

Overview – *After some general remarks, we briefly review and discuss the main results. Then we draw some conclusions.*

6.1. Some general remarks

6.1.1. A critique?

> "Don't mind criticism. If it is untrue, disregard it; if unfair, keep from irritation; if it is ignorant, smile; if it is justified it is not criticism, learn from it. "
>
> —Author unknown

This thesis is neither complete nor wholly impartial. From the very start, it was our intention to communicate a particular view. A view that would find itself at odds, not only with popular accounts, but with the larger part of the scientific literature as well. To maximize the effect this view would have, we thought it good to present the discrepancies as sharp as we possibly could, in the hope our contribution would spur more discussion.

No doubt, some will consider our contribution a critique. And, in a way, it is. But even so, it is no mere criticism. As we mentioned in the opening chapter, our intention was to lay bare ILC's distinctiveness. We think ILC has considerable potential, be it that it does not always show. If we wish to further this potential, me must know how to exploit it. Herein our contribution lies.

6.1.2. First-order recurrences and the problem of Learning Control

We set out to compare two, apparently distinct, methods of control. We chose to focus on a well-established class of first-order linear recurrences. This class originates one of the strongest currents in ILC research, and is representative of a large majority of algorithms. Indeed, to date most algorithms still conform to the same basic structure as was at the basis of Arimoto's 1984 D-type learning rule—a structure which looks like this:

$$u_{k+1} \;=\; F(u_k, e_k). \tag{6.1}$$

Here, u_{k+1}, u_k and e_k respectively denote the future input, the current input, and the current error. The operator $F : U \times Y \mapsto U$ is typically *linear* (this is true even if the plant is not) in which case it may be decomposed into an operator $Q : U \mapsto Y$, acting on the first argument (the control input), and an operator L, acting on the second (the tracking error):

$$F(u, e) \;=\; Qu + Le. \tag{6.2}$$

In this thesis we assumed Q and L to be bounded (on $[0, \infty)$). This assumption may seem restrictive, excluding many algorithms of interest, but it really is not. For we do not need unbounded operators, at least not in the context of linear, time-invariant systems. After all, any finite-dimensional LTI operator $G(s) \in \mathcal{RL}_\infty$ can be decomposed into a causal and an anti-causal part, both bounded. As for derivative-type (nonproper) operators: these can be looked upon as improper limits of particular sequences of (bounded) noncausal operators. By using (bounded) noncausal, instead of nonproper operators, the derivative action may be spread out (as it were) over multiple trials. The additional advantages: a well-behaved algorithm and smooth (monotone) convergence.

6.2. Conclusion

6.2.1. Main Results

Let us now review the main results of the thesis.

A two-parameter synthesis problem

We chose to define the problem of Iterative Learning Control as an optimization problem over the space of bounded linear operator pairs (Section 2.3),

and distinguished two cases: That of (a) Standard ILC (Section 2.3.2), and (b) Current-Cycle Feedback ILC (CCF-ILC), also known as Current Iteration Tracking Error ILC, or CITE-ILC (Section 2.3.3). A special case of CCF-ILC, Standard ILC assumes a stable plant and does not allow any information to be fed back *within* the same trial (that is to say, no *closed-loop* feedback). CCF-ILC does not impose stability conditions on the plant, but instead assumes the overall system to be stabilized by some feedback compensator. This compensator obviously *does* feed back information within the same trial. Hence the term 'current-cycle feedback'.

Standard ILC, and CCF-ILC both relate to the same class of first-order (linear) recurrences:

$$u_{k+1} \;=\; Qu_k + Le_k + Ce_{k+1}, \tag{6.3}$$

a class that is characterized by a threesome parameters: (a) an operator acting on the previous input, Q; (b) an operator acting on the previous error, L, and (c) an operator acting on the current error, C. The first two are considered design parameters; The third is not. In both Standard ILC and CCF-ILC, the objective is to find the parameter combination (Q, L) that maximizes *asymptotic* performance. As said, in Standard ILC, we assume $C = 0$.

The notion of admissible pairs

In Chapter 3 we took our first shot at the problem, starting with Standard ILC. For historical reasons, we decided to focus on a class of *causal* algorithms, contraining both design parameters to be causal.

We started by introducing a notion of *admissibility*, see Definition 3.3.1. One would like to think of this notion as the ILC analog of internal stability. Roughly speaking, a pair of operators (Q, L) is said to be admissible if: (a) both the input, and output sequence *converge* to some bounded fixed point, independent of the initial condition; (b) the convergence refered to in (a) is *robust* against all bounded additive disturbances. That is to say, the solution to the iterative system thus perturbed should remain close to that of the unperturbed system, and the both should converge in case the disturbances should subside.

Regarding admissibility, we found a single condition to be both necessary and sufficient (Section 3.4.1, Lemma 3.4.1). This condition was already known to be sufficient. We proved it to be necessary. We discussed the fact that there are different ways to define a notion like admissibility. Typical of our definition is: (a) that we require both operators to be bounded on the

entire right semi-infinite time interval and (b) that we want our solutions to be robust in the sense explained above.

Redundancy and equivalent pairs

With this condition of admissibility at our disposal, we were able to tell what parameter combinations would warrant an acceptable solution and what combinations would not. Next we had to select, from among the 'good' combinations, the ones that would give best performance.

It turned out that this very set of 'good combinations' (the set of admissible pairs) was redundant. That is to say, we found that two pairs of operators could induce different sequences that would eventually converge to the same fixed point (see in particular Equation (3.24) and Example 3.4.5). In view of the focus of ILC research, which is on *asymptotic* performance, it appeared sensible to equate these pairs, and so we introduced an equivalence relation (Section 3.4.2, Definition 3.4.3). For two pairs of operators to be considered equivalent, they would have to meet a certain equivalence condition and for this condition to make sense, we had to assume the plant to be strictly proper.

Using this relation we were able to identify equivalent pairs. Now we could remove redundancy (see Section 3.4.4). What we did was to assign, within each equivalence class, one particular member to represent the class. But then a problem presented itself. For what member would we choose? The choice, of course, was immaterial, as all iterations would converge to the same fixed point anyway. Yet we had to choose one.

We discovered that, to distinguish equivalent pairs we had to involve another variable: convergence speed. Indeed, we found that 'equivalent' iterations would differ, not only in terms of *how* they would reach the fixed point, but also *how fast* they would reach it. This one parameter, convergence speed, thus represented all possible ways in which we could select a set of representatives; All we had to do was fix this parameter and we would get our representatives for free. We chose to set it to zero, which meant that from that point on, we would restrict attention to those iterations that converge within a single trial.

Equivalent Feedback

This notion of equivalence turned out to have an unexpected interpretation in terms of Equivalent Feedback. Concisely put, the principle of Equivalent Feedback states that to every iteration of the form (6.3), with (Q, L) admissible and causal, there corresponds a particular (causal) feedback map,

the Equivalent Controller, which, in closed loop, would reproduce the exact same control effort, without iterations. We found that admissible pairs, equivalent in the sense of Definition 3.4.3, corresponded to different left-coprime factorization of the same Equivalent Controller.

We proved this Equivalent Controller to be stabilizing (Theorem 3.4.7) and showed the converse result to hold as well (Theorem 3.4.8). That is to say, we showed: (a) that each admissible pair defines a particular, stabilizing controller, and (b) that every stabilizing controller admits a factorization in terms of admissible pairs (and more than one factorization at that). By restricting attention to a class of representatives, we could establish a bijection between this smaller set of admissible pairs and the set of stabilizing controllers.

This showed that the respective methods of Causal ILC and conventional feedback control are essentially equivalent; indeed, that Causal ILC is feedback control in disguise. However, we were unable to justify this claim by any other than purely theoretical means. Our experimental results (Section 3.6.1) showed a small, but persistent discrepency in performance, in favor of ILC. There is nothing conclusive at this point, but our best guess is that this phenomenon is caused by noise (measurement noise, quantization noise), see also our discussion in Sections 3.4.5 and 3.6.4. We did confirm the Equivalent Controller to be stabilizing in all cases considered.

Extensions to CCF-ILC

These results were readily extended to the case of CCF-ILC (Section 3.5, the result on Equivalent Feedback is in Theorem 3.5.2). But with one exception: it turned out that in CCF-ILC, the set of Equivalent Controllers is just a subset of all stabilizing controllers. These sets do not coincide, unless the current-cycle term (the nominal feedback controller) is strongly stabilizing (i.e. both stable and stabilizing), see Theorem 3.5.3.

Noncausal ILC

Then we turned to Noncausal ILC. Conscious of the historical emphasis on causal algorithms, our first concern was to make clear that Noncausal ILC is no mere extension of causal ILC; that it is in fact more natural to consider noncausal algorithms, than it is to consider causal algorithms.

To support our argument, we brought in some optimization problem. The objective of this problem was to find the minimizing argument of some cost functional $J(\Delta u)$, with Δu the control increment. We showed that the solution would involve the use of noncausal operators (refer to Equation (4.3)).

In the course of the chapter (which dealt exclusively with the Standard ILC case) we put forth a number of additional arguments as to why we would want to consider noncausal update laws. Most of these arguments related to performance, or rather the lack therof in case we would insist on causality. We showed that by imposing causality constraints, we would seriously impair the achievable performance, particularly so when the plant: (a) has relative degree greater than one (Lemma 4.2.2) and (b) exhibits non-minimum phase behavior (Section 4.2.4).

In this respect, zero and first-order minimum-phase systems were shown to form an exception. With these systems we can attain perfect tracking even with causal algorithms (Lemma 4.2.3). The corresponding, causal, operators would not be admissible in the sense of Definition 3.3.1, as they do not guarantee boundedness of the input. But that, in fact, has more to do with perfect tracking than with causality. For had we used noncausal operators they would still not be admissible.

In constrast, we showed that, regardless whether the system is minimum phase or non-minimum phase, there is always a bounded *noncausal* operator pair (I, L) such that $I - LP$ is less than unity for almost all frequencies (Lemma 4.2.4). And this condition implies that, in Noncausal ILC: (a) none of the aforementioned constraints (Bode's Sensitivity Integral, Poisson's Integral Relation) applies and (b) perfect tracking can always be achieved (though, again, the input need not converge if we do not constrain the output to be feasible).

If we insist on boundedness of the input (which we would do in most practical applications) we ought to consider deploying the somewhat stronger condition of admissibility. Under the condition of admissibility, perfect tracking will be hard to attain (if P is strictly proper, there is no L such that $I - LP$ is a strict contraction). But even then, there is much to be gained, performancewise.

As for Equivalent Feedback, we showed that, in general, no *causal* feedback map exists that would give the same performance without iterations. A noncausal map would do the trick, but if we would try to implement it causally, it would be destabilizing. And to implement it noncausally would be impossible in closed loop.

Finally, a remark. In our analyses, we adopted an infinite-time perspective. We showed that such analysis is conservative in the sense that if $Q-LP$ is a contraction on $\mathcal{L}_2[0,\infty)$ then it follows that its restriction onto $[0,T]$ is a contraction on $\mathcal{L}_2[0,T]$ (Lemma 4.A.2). This suggests that our approach is 'safe'. Yet, there are problems. The effects of truncation, for instance, are not well understood. Unwanted behavior may occur, especially during

the transient phase. This problem, which originates in the initialization of the noncausal filters, needs further study.

Trial-dependent update laws

In Chapter 5 we looked into a class of trial-dependent update laws. This was after we had concluded that, by construction, trial-independent update rules would not be able to accommodate even the simplest forms of adaptation. It was not *performance* we were after. In that respect, the class of noncausal algorithms would do perfectly well. What we did expect to find was a potential for *adaptation*, an ability to enhance performance robustness.

We would start from the same basic algorithm, augment it so as to accommodate different types of trial-dependent behavior, and then study the results. As a first step we replaced the fixed operator pair (Q, L) with a sequence of operator pairs $\{(Q_k, L_k)\}$, to obtain algorithms of the form

$$u_{k+1} = Q_{k+1}u_k + L_{k+1}e_k. \tag{6.4}$$

At first, we did not allow any of the operators to depend on trial data: we simply assumed (Q_k, L_k) to be given for all k.

We derived a condition on $\{(Q_k, L_k)\}$ under which the sequence of inputs induced by (6.4) would converge to a unique fixed point, independent of the initial condition. This involved extending some known results in fixed point theory (Theorem 5.3.4). Basically, $Q_k - L_kP$ had to be contractive for almost all k, with a finite number of exceptions at most.

Next we proved the above class of trial-dependent update laws, (6.4), and the next class of trial-independent update laws,

$$u_{k+1} = Qu_k + Le_k \tag{6.5}$$

to be equivalent. More precisely, assuming $\{Q_k, L_k\}$ to be contractive and *strongly converging* (Definition 5.3.1), we showed that there exist Q and L, with (Q, L) admissible such that (6.4) and (6.5) both converge to the same fixed point.

Led by this result, we decided to look into a class of *non-contractive* update laws:

$$u_{k+1} = u_k + L_{k+1}e_k. \tag{6.6}$$

The adjective 'non-contractive' refers to the fact that $I - L_kP$ is not assumed to be contractive for (almost) all k; that condition would be impossible to satisfy anyway. Instead, we assumed L_k to strongly converge to zero. We proved that, provided $\|L_k\|$ is summable (Definition 5.3.5), the sequence of

inputs induced by (6.6) would always converge to some bounded solution (Theorem 5.3.6).

We had argued that since this class does not allow for Equivalent Feedback, it might be of interest. We were wrong. The one interesting thing about algorithms of the form (6.6) is that it is impossible to tell where it will converge to without actually running the scheme. The iteration cannot be reduced to a competitive scheme of lower complexity. To put it differently: each and every parameter L_k matters; we cannot, as we could before, replace the whole sequence of operators with a single operator and get the same outcome. All the same, this class was not going to give us what we were looking for. It could do little to *enhance* performance (Theorem 5.3.8), and for sure, it would not make it more robust. Besides, its effectiveness would depend on the initial condition, which, from a generic stand, is unacceptable.

So what we learned is that: (a) there is little point in considering trial-dependent update laws if we do not allow the operators to 'adapt'; (b) there is no point whatsoever in considering trial-dependent update laws for reasons of performance. For in general trial-dependent, and trial-independent classes are subject to the same constraints.

Led by these observations we decided to do a case study on two potentially adaptive algorithms. The first algorithm (Algorithm 5.3.9, Example 5.3.10) was based on a simple scheme that would tune a particular gain parameter to a 'safe' value. The purpose of this example was to show that by very elementary means, we can break the tradeoff between performance and stability robustness. The use of adaptation allowed for a competitive (i.e. non-cautious) design that would have surely compromised some stability margins in the nominal setup.

The next case (Algorithm 5.3.11, Example 5.3.12) was still more interesting, at least from a performance point of view. For despite its other virtues, the previous algorithm did little to enhance performance robustness. This time, our starting point was a noncausal update law. As in the previous example, a simple scheme was to adapt a single scalar parameter. The effectiveness of the algorithm was remarkable. Performance was not only good (perfect tracking), but also robust. This succes (which is purely theoretical at this point) was due to a combination of using what we know (exploiting the fact that the system is LTI) and knowing what we do not know (the system gain).

6.2.2. Conclusions

How does Iterative Learning Control compare with conventional control? And more in particular, what is the use of iteration, both in relation to performance enhancement and in dealing with uncertainty? These are the questions we started out with. Now, finally, the time has come to provide some answers.

The need for iteration

First, let us try to explicate the *need* for iteration. Basically there are two scenarios in which such need arises. In the first scenario, we have a competent model of our system and presumably, an equally competent controller. Yet, because of some limitations inherent to any feedback design, performance is not quite what we want it to be. In this scenario, iteration is to deliver that little bit of extra performance. In the second scenario, the model we have is not at all that good. In fact, due to a lack of information we are forced to opt for a conservative design. As a result, our controller is far from optimal. In this scenario, iteration is to compensate for the lack of a priori knowledge, by generating data from which additional information may be extracted.

In the first scenario, the objective is to get the absolute best achievable performance. In the second, we would settle for less (say, just 'good' performance), provided this same performance could be guaranteed for all systems within a specific range—a range determined by the extent of uncertainty.

We concluded that, as it is, Iterative Learning Control answers to the first scenario and to the first scenario only. The uncertainty issue (as illustrated by the second scenario) is often ignored. The great majority of algorithms is designed with the single purpose of delivering perfect tracking. And many do a good job at that. So good in fact, that we tend to forget how they pull the trick. Let us be mindful of the fact that this success critically depends on the availability of prior knowledge.

Causal ILC and Equivalent Feedback

This became poignantly clear when we studied a class of causal algorithms and found the use of iterations artificial and superfluous. And at this point, we can safely say that Causal ILC is, in essence, noniterative. Not because there happens to exist some feedback map that delivers the exact same performance. For by itself, this does not prove anything. But rather because of the fact that this very controller does not depend on anything but the

ILC parameters. For this conclusively shows that, despite their apparent differences, Causal ILC and conventional feedback control are nothing but different manifestations of the same idea.

If Causal ILC is taken to be representative of all learning algorithms, then one of two conclusions must apply: (a) either ILC does not learn, or (b) conventional feedback control is more intelligent than we believed it to be. Fortunately, there is no reason to confine ourselves to causal algorithms.

Noncausal ILC

And indeed, it turns out that much can be gained by considering noncausal algorithms. At least performancewise, that is. This of course makes sense, as most of the limitations inherent in Causal ILC and conventional feedback control ultimately pertain to causality related constraints. Not bound by any such constraints, noncausal update laws can effect a significant increase in performance, particularly when applied to systems exhibiting nonminimum phase behavior and high relative degree. What is more, on every SISO LTI system there is at least one noncausal update law that will have the system's output converge to a desired output, no matter what that output be. This property makes a good starting point as we take on the uncertainty issue.

The uncertainty issue

Whether they be causal or noncausal, as long as these update laws are trial-independentg, their effectiveness is constrained by how much we know. And the general rule is: the more we ask, the more we need to know. So obviously, to effect perfect tracking, we need to know a lot, though we do not have a conclusive answer as to say how much that is. In case of Arimoto's D-type algorithm: at least the sign of the first (nonzero) Markov parameter. And we will not be far from the truth by supposing the same is true for the general class of (non)causal update laws.

By construction, trial-independent update laws will never be able to adapt. These schemes are so rigid that even if the model is falsified by the data, it will not affect execution. Of course, in practice, iteration will be terminated long before things get out of hand. Yet, this simple argument does reveal a serious shortcoming.

There are only two possible ways in which ILC can distinguish itself from conventional control. One has to do with performance proper, the other with performance robustness. Though there is more to be said, at this point, it appears the performance issue is settled in favor of ILC. With

ILC, we can attain good performance for a large variety of plants (and that includes the kind of plants we did not discuss in this thesis, such as those with nonlinear, time-varying dynamics, etc.). Yet little is known about how to shield performance from uncertainty.

We believe that the Learning Control community would do good if it would make this issue its focus. We suggested that trial-dependent update laws would make a good starting point and provided two examples. There are other ways. In fact there is no reason to confine oneself to algorithms of any particular kind. As far as uncertainty is concerned, iteration is just a means of acquiring data. The problem of Learning Control is to combine this data with whatever prior knowledge one has, so as to enhance a system's performance. In this sense, Learning Control for LTI systems may be a non-issue, but for less trivial systems, the problem is largely unsolved.

A

Bibliography

Ravi P. Agarwal, Maria Meehan, and Donal O'Regan. *Fixed Point Theory and Applications*. Number 141 in Cambridge Tracts in Mathematics. Cambridge University Press, 2001.

N. Amann, D. H. Owens, and E. Rogers. 2D systems theory applied to learning control systems. In *Proc. of the 33rd Conf. on Decision and Control*, pages 985–986, Lake Bruena, FL, USA, Dec. 1994.

N. Amann, D. H. Owens, and E. Rogers. Iterative learning control using optimal feedback and feedforward actions. *International Journal of Control*, 65(2):277–293, September 1996a.

N. Amann, D. H. Owens, and E. Rogers. Robustness of norm-optimal iterative learning control. In *Proceedings of International Conference on Control*, volume 2, pages 1119–1124, Exeter, UK, September 1996b.

N. Amann, D. H. Owens, E. Rogers, and A. Wahl. An H_∞ approach to linear iterative learning control design. *International Journal of Adaptive Control and Signal Processing*, 10(6):767–781, November-December 1996c.

S. Arimoto, S. Kawamura, and F. Miyazaki. Bettering operation of robots by learning. *J. of Robotic Systems*, 1(2):123–140, 1984.

S. Arimoto, S. Kawamura, and F. Miyazaki. Convergence, stability and robustness of learning control schemes for robot manipulators. In M. J. Jamishidi, L. Y. Luh, and M. Shahinpoor, editors, *Recent Trends in Robotics: Modelling, Control, and Education*, pages 307–316. Elsevier, NY, USA, 1986.

S. Bennett. *A History of Control Engineering 1800-1930*. The Institution of Electrical Engineers (IEE), 1979.

Dennis S. Bernstein. Feedback control: an invisible thread in the history of technology. *Control Systems Magazine*, 22(2):53–68, April 2002.

Zeungnam Bien and Jian-Xin Xu. *Iterative Learning Control: analysis, design, integration and applications*. Kluwer Academic Publishers, 1998.

Okko H. Bosgra and Huibert Kwakernaak. *Design methods for control systems*. Lecture Notes, Dutch Institute of Systems and Control, 2001.

Y. Chen, C. Wen, and M. Sun. A robust high-order P-type iterative learning controller using current-iteration tracking error. *Int. J. of Control*, 68(2): 331–342, Sept. 1997a.

Y. Chen, J.-X. Xu, and T. H. Lee. Current-iteration tracking error assisted iterative learning control of uncertain nonlinear discrete-time systems. In *Proc. of the 35th IEEE Conference on Decision and Control*, pages 3040–5, Kobe, Japan, Dec. 1996a.

Y. Chen, J.-X. Xu, and T. H. Lee. An iterative learning controller using current-iteration tracking error information and initial state learning. In *Proc. of the 35-th IEEE CDC*, pages 3064–9, Kobe, Japan, Dec. 1996b.

Y. Chen, J.-X. Xu, and T.H. Lee. Current-iteration tracking error assisted high-order iterative learning control of discrete-time uncertain nonlinear systems. In *Proc. of the 2nd Asian Control Conference (ASCC'97), Vol. I*, pages 573–6, Seoul, Korea, Jul. 1997b.

Yangquan Chen and Changyun Wen. *Iterative Learning Control: convergence, robustness and applications.* Lecture Notes in Control and Information Sciences. Springer-Verlag, 1999.

R.F. Curtain and H.J. Zwart. *An Introduction to Infinite-Dimensional Linear Systems Theory*. Springer-Verlag, 1995.

B.J. de Kruif. *Function Approximation for Learning Control—a Key Sample Based Approach*. PhD thesis, University of Twente, 2004.

Dick de Roover. Synthesis of a robust iterative learning controller using an H_∞ approach. *Proceedings of the 35th Conference on Decision and Control*, pages 3044–3049, December 1996.

Dick de Roover, Okko H. Bogra, and Maarten Steinbuch. Internal-model-based design of repetitive and iterative learning controllers for linear multivariable systems. *International Journal of Control*, 73(10):914–929, 2000.

Dick de Roover and Okko H. Bosgra. Synthesis of robust multivariable iterative learning controllers with application to a wafer stage motion system. *International Journal of Control*, 73(10):968–979, 2000.

J. B. Edwards and D. H. Owens. *Analysis and Control of Multipass Processes*. Research Studies Press, Taunton, Chichester, 1982.

Engell. Optimale lineare regelung: Grenzen der erreichbaren regelgüte in linearen zeitinvarianten regelkreisen. *Fachberichte Messen-Steuern-Regeln*, 18, 1988.

B.A. Francis. *A Course in $\mathcal{H}_\infty$ Control Theory*, volume 88 of *Lecture Notes in Control and Informations Sciences*. 1987.

B.A. Francis and W.M. Wonham. The internal model principle for linear multivariable regulators. *Applied mathematics and optimization*, 2:175–194, 1975.

J.S. Freudenberg and D.P. Looze. Right half plane poles and zeros and design tradeoffs in feedback systems. *IEEE Transactions on Automatic Control*, 30:555–565, 1985.

Peter B. Goldsmith. The fallacy of causal iterative learning control. In *Proc. Of the 40th Conference on Decision and Control*, pages 4475–4480, Orlando, Florida, USA, 2001.

Peter B. Goldsmith. On the equivalence of causal LTI iterative learning control and feedback control. *Automatica*, 38(4):703–708, 2002.

G. Heinzinger, D. Fenwick, B. Paden, and F. Miyazaki. Stability of learning control with disturbances and uncertain initial conditions. *IEEE Trans. of Automatic Control*, 37(1):110–114, 1992.

T.-J. Jang, C.-H. Choi, and H.-S. Ahn. Iterative learning control in feedback systems. *Automatica*, 31(2):243–245, 1995.

T. Kavli. Frequency domain synthesis of trajectory learning controllers for robot manipulators. *Journal of Robotic Systems*, 9(5):663–680, 1992.

Kevin Kelly. *Out of Control: the rise of neo-biological civilization*. Perseus Publishing, 1994.

J. E. Kurek and M. B. Zaremba. Iterative learning control systhesis based on 2-D system theory. *IEEE Trans. of Automatic Control*, 38(1):121–125, 1993.

H. Kwakernaak and R. Sivan. *Modern Signals and Systems*. Information and System Sciences. Prentice-Hall, 1991.

Hak-Sung Lee and Zeungnam Bien. Study on robustness of iterative learning control with non-zero initial error. *International Journal of Control*, 64: 345–359, June 1996.

Y.-J. Liang and D. P. Looze. Performance and robustness issues in iterative learning control. In *Proc. of the 32nd Conf. on Decision and Control*, pages 1990–1995, San Antonio, Texas, USA, Dec. 1993.

A. De Luca, G. Paesano, and G. Ulivi. A frequency-domain approach to learning control: implementation for a robot manipulator. *IEEE Trans. on Industrial Electronics*, 39(1):1–10, 1992.

David G. Luenberger. *Optimization by Vector Space Methods*. John Wiley & Sons, 1969.

Pater Hagoort & Rob Maessen. *Geest, Computer, Kunst*. Stichting Grafiet, Utrecht, 1990.

Principia Cybernetica. `http://pespmc1.vub.ac.be/DEFAULT.html`, 2004.

K. L. Moore. *Iterative learning control for deterministic systems*. Advances in Industrial Control. Springer-Verlag, 1993.

K. L. Moore. Iterative learning control: an expository overview. *Applied & Computational Controls, Signal Processing, and Circuits*, 1(1):151–214, 1999.

K. L. Moore, M. Dahleh, and S. P. Bhattacharyya. Iterative learning control: a survey and new results. *J. of Robotic Systems*, 9(5):563–594, 1992.

Kevin L. Moore. A matrix fraction approach to higher-order iterative learning control: 2d dynamics through repetition-domain filtering. In *Proceedings of the 2nd International Workshop on Multidimensional (nD) Systems, 27-30 June, Czocha Castle, Lower Silesia, Poland*, pages 99–104, 2000.

Desineni Subbaram Naidu, Selahattin Ozcelik, and Kevin L. Moore. *Modeling, Sensing and Control of Gas Metal Arc Welding*. Elsevier Science Ltd., Oxford, UK, 2003.

Mikael Norrlöf. *Iterative Learning Control: analysis, design, and experiments*. PhD thesis, Department of Electrical Engineering, Linköpings Universitet, Sweden, 2000.

D. H. Owens. Iterative learning control: convergence using high gain feedback. In *Proceedings of the 1992 Conference on Decision and Control*, volume 4, page 3822, Tucson, AZ, December 1992.

D. H. Owens, N. Amann, and E. Rogers. Iterative learning control: an overview of recent algorithms. *Applied Mathematics and Computer Science*, 5(3):425–438, 1995.

F. Padieu and R. Su. An H_∞ approach to learning control systems. *Int. J. Adap. Contr. and Sig. Proc.*, 4, 1990.

Kevin M. Passino. Bridging the gap between conventional and intelligent control. *IEEE Control Systems Magazine*, 13(3):12–18, 1993.

E. Rogers, K. Galkowski, D.H. Owens, T. Al-Towaim, J.D. Ratcliffe, and P.L. Lewin. 2d linear control systems: from theory to experiment to theory. In *Proceedings of Int. Symp. On Mathematical Theory of Network and Systems*, 2002.

Erwin Schrijver. *Improved Robot Tracking Control for Laser Welding*. PhD thesis, Drebbel Institute for Mechatronics, University of Twente, 2002.

Robert Sedgewick and Philippe Fjajolet. *Analysis of Algorithms*. A, 1996.

Alan Titchmarsh. The Theory of Functions, 1939.

Masayoshi Tomizuka. Zero phase error tracking algorithm for digital control. *Journal of Dynamic Systems, Measurement, and Control*, 109:65–68, 1987.

Y.A. Tsypkin. *Adaptation and Learning in Automatic Systems*. Academic Press, New York, New York, 1971.

Y.A. Tsypkin. *Foundations of the Theory of Learning Systems*. Academic Press, New York, New York, 1973.

A. M. Turing. Intelligent machinery. In C. R. Evans and A. D. J. Robertson, editors, *Cybernetics: Key Papers*, number 7. University Park Press, Baltimore, 1968.

Gordon T. Uber. Clocks and Time—History of time-keeping. `http://www.ubr.com/clocks/hist/hist.html`, 2004.

W.J.R. Velthuis. *Learning Feed-Forward Control*. PhD thesis, University of Twente, Enschede, The Netherlands, 2000.

M.H.A. Verwoerd, G. Meinsma, and T.J.A. de Vries. On the use of non-causal LTI operators in iterative learning control. *Proceedings of the 41st Conference on Decision and Control*, pages 3362–3366, 2002.

M.H.A. Verwoerd, G. Meinsma, and T.J.A. de Vries. On equivalence classes in iterative learning control. *Proceedings of the American Control Conference, Denver, Colorado, USA*, pages 3632–3637, 2003.

M.H.A. Verwoerd, G. Meinsma, and T.J.A. de Vries. On the parameterization of all admissable pairs in a class of CCF-ILC algorithms. *Proceedings of the American Control Conference, Boston, Massachusetts, USA*, pages 5156–5157, 2004.

M.H.A. Verwoerd, G. Meinsma, and T.J.A. de Vries. A class of non-contractive, trial-dependent update rules for Iterative Learning Control. *Proceedings of the American Control Conference, Boston, Massachusetts, USA*, pages 5132–5137, 2004.

M.H.A. Verwoerd, G. Meinsma, and T.J.A. de Vries. On admissible pairs and Equivalent Feedback: Youla parameterization in Iterative Learning Control. Submitted for publication.

J.-X. Xu. Direct learning of control input profiles with different time scales. In *Proceedings of the 35th IEEE Conference on Decision and Control*, Kobe, Japan, December 1996.

Jian-Xin Xu and Ying Tan. *Linear and Nonlinear Iterative Learning Control*. Number 291 in Lecture Notes in Control and Information Sciences. Springer-Verlag, 2003.

Jian-Xin Xu and Tao Zhu. Dual-scale direct learning control of trajectory tracking for a class of nonlinear uncertain systems. *IEEE Transactions on automatic control*, 44(10):1884 – 1888, 1999.

D.C. Youla, H.A Jabr, and J.J. Bongiorno. Modern wiener-hopf design of optimal controllers: Part I. *IEEE Transactions on Automatic Control*, 21:3–13, 1976.

G. Zames. Feedback and optimal sensitivity: model reference transformations, multiplicative seminorms and approximate inverses. *IEEE Transactions on Automatic Control*, 26:301–320, 1981.

Kemin Zhou, John C. Doyle, and Keith Glover. *Robust and Optimal Control*. Prentice Hall, 1996.

Index