# Robust iterative learning control

# Robust Iterative Learning Control

## A Literature Survey

Tijs Donkers

DCT 2007-21

Literature Survey

Supervisor: prof. ir. Okko Bosgra
ir. Jeroen van de Wijdeven

Technische Universiteit Eindhoven
Department Mechanical Engineering
Control Systems Technology Group

Eindhoven, Februari 2007

# Table Of Contents

# Nomenclature

### Abbreviations

| | |
|---|---|
| AIU | Allowable Interval Uncertainty |
| HJB | Hamilton-Jacobi Bellman (system) |
| ILC | Iterative Learning Control |
| IMP | Internal Model Principle |
| LFT | Linear fractional transformation |
| LTI | Linear Time Invariant |
| LTV | Linear Time Varying |
| MIMO | Multiple Input Multiple Output |
| NMPhZ | Non-Minimum Phase Zeros |
| NP | Nominal Performance |
| RP | Robust Performance |
| RS | Robust Stability |
| SISO | Single Input Single Output |
| SSV | Structured Singular Value |

### General notation

| | |
|---|---|
| $\equiv$ | equal by definition |
| $:=$ | redefinition |
| $A, B, C, D$ | Matrices belonging to a state space description |
| $\lambda$ | Eigenvalue |
| $\sigma$ | Singular value |
| $\Sigma, U, V$ | Matrices belonging to the singular value decomposition |
| $\rho(\bullet)$ | Spectral radius |
| $\|\| \bullet \|\|_i$ | Norm of a vector, matrix, signal, or system |
| $\langle \bullet \rangle$ | Bigger norm value between a matrix and its transpose |
| tr | Trance operator, sum of the diagonal element of a matrix |
| $\mathcal{J}$ | Cost functional |

## Notation used for ILC

| | |
|---|---|
| $J$ | ILC system (often closed loop process sensitivity function) |
| $L$ | learning filter |
| $N$ | trial length |
| $Q$ | robustness filter |
| $e, E(z)$ | tracking error signal |
| $f, F(z)$ | command signal for ILC |
| $k$ | trial number |
| $m$ | delay |
| $r, ref$ | reference trajectory for ILC |
| $u$ | output of the learning filter |
| $y, Y(z)$ | system output |

## Notation used for feedback control

| | |
|---|---|
| $D$ | uncertainty scale |
| $G$ | open loop system |
| $K$ | controller |
| $L$ | Loop gain |
| $P$ | generalised plant |
| $S$ | closed loop sensitivity function |
| $W$ | uncertainty weight |
| $X, Y$ | solutions to a Riccati equation |
| $u$ | controlled input |
| $w$ | disturbance |
| $x$ | state |
| $y$ | measured output |
| $z$ | performance output |
| $\Delta$ | (Complex) perturbation matrix |
| $\gamma$ | bound on the $\infty$-norm |
| $\delta$ | real valued perturbation |
| $\lambda$ | costate |
| $\mathcal{F}_l$ | lower Linear Fractional Transformation |
| $\mathcal{F}_u$ | upper Linear Fractional Transformation |
| $\mathcal{H}_\infty$ | Banach space of matrix valued functions that are analytic outside the unit circle and have a bounded $\infty$-norm. |

## Subscripts

| | |
|---|---|
| $i$ | general index |
| $k$ | trial index |
| $p$ | perturbed |

# Chapter 1 Introduction

In many applications, a system has to perform the same task over and over again. Examples of such systems are robotic manipulators, (e.g., a pick and place machine, or a wafer-stage), and chemical batch processes. Since the task is repetitive in nature, any feedback controller applied to these systems will result in a error signal that is equal for every repetition (trial). If this error does not meet the required performance specifications, a solution has to be found that is able to meet the specifications.

One interesting solution is given by Iterative Learning Control (ILC). ILC is a control strategy based on the idea that the performance of a system that executes a repeated task can be improved by learning from previous trials. When properly designed, the ILC controller iteratively finds a command signal that results in high system performance, despite model uncertainty and unknown, repeating disturbances.

## 1.1 Concept of ILC

The idea of using an iterative method for dealing with repetitive disturbances is almost 40 years old. In 1967, a patent was filed with the title *'Learning Control of Actuators in Control Systems'*. One of the first academic papers in English that discusses ILC is [5], which was published in 1984. The term *'Iterative Learning Control'* was introduced by the same author later that year.

The general idea of ILC is illustrated in figure 1.1 [32]. All signals shown are defined on a finite time interval $t \in [0, t_f]$. The subscript $k$ denotes the trial or repetition number. The algorithm operates as follows: during trial $k$ an input $f_k$ is applied to the system $J$. This input is stored in a memory together with the resulting output $y_k$. After the trial has finished, an error $e_k$ is calculated, based on a desired trajectory $r$. (There is no subscript $k$, since $r$ is equal for all trials.) Both error and system input can then be processed by the ILC algorithm: $f_{k+1} = ILC(f_k, e_k)$, resulting in an input $f_{k+1}$, as function of $f_k$ and $e_k$. The input is subsequently applied during the next trial. When properly designed, the error $e_{k+1}$, resulting from input $f_{k+1}$, will be smaller than $e_k$. This process can then be repeated, making the error smaller and smaller.

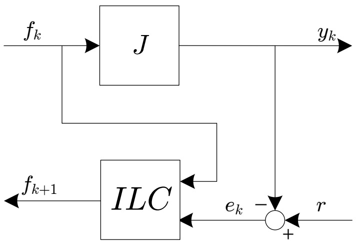


Figure 1.1: ILC block diagram

This idea is further illustrated by figure 1.2, where it can be seen that the error between the desired trajectory (dashed) and the measured output (solid) is getting smaller from iteration to iteration. After each trial $k$ is ended, the system is reset to its initial position.

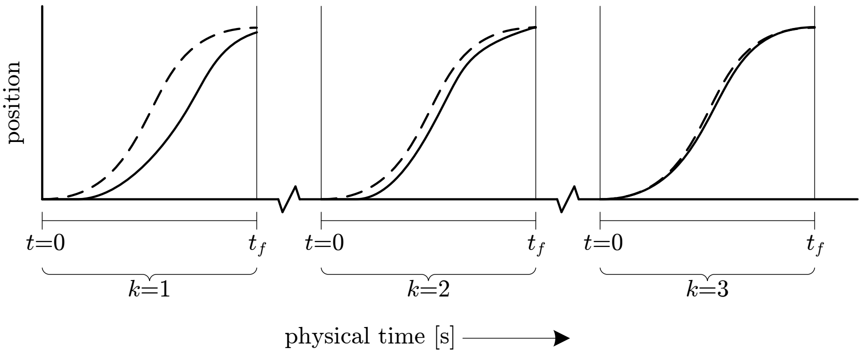


Figure 1.2: ILC as it progresses in time. The reference trajectory (dashed) and the measured output (solid).

## 1.2 ILC in relation to other techniques

ILC has some advantages over a *well-designed feedback and feedforward controller* (see figure 1.3) even though both can yield good performance. A feedback controller $K_{\text{fb}}$ reacts to an error $e$, introduced by a desired trajectory or disturbances and the system output $y$, and therefore always has a lag. Feedforward $K_{\text{ff}}$ can be used to compensate for this lag, however, only for known signals, such a reference trajectory, and not for unknown disturbances. ILC is capable of compensating for unknown disturbances, using information from previous trials. The only requirement for ILC is that the reference trajectory and disturbances repeat from iteration to iteration.

While a feedback controller can handle uncertainties in the system model, a feedforward controller only performs well using an exact system model. Because ILC learns its control signal from measurement data, it is robust against unmodelled dynamics and nonlinear behaviour. Furthermore, because all calculations are done between consecutive trials, ILC allows for using advanced filtering, such as zero-phase filtering [22].

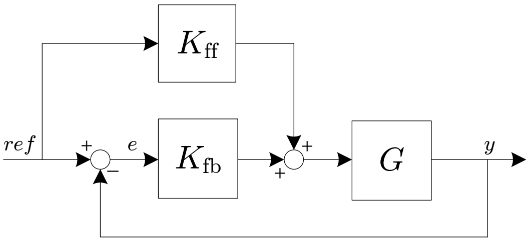


Figure 1.3: Block diagram of feedback control in conjunction with feedforward control

One control strategy that can handle an uncertain system model is *Adaptive Control* (see, e.g., [44, Chapter 8]). In this strategy, it is assumed that the structure of the system is known and the exact values of the parameters are unknown. These are then determined using an online identification procedure. Using these parameters, a feedforward signal is calculated. This gives good performance, under varying conditions, if the structure of the system is correct and the control signal sufficiently excites the system. Note that, contrary to ILC, the controller parameters are adapted. In ILC, the command signal itself is adapted.

ILC is perhaps most related to *Repetitive Control*, which is based on similar theory [17]. Both control laws are used to deal with repetitive disturbances. The difference lies in the fact that repetitive control is intended for continuous operations, whereas ILC is used for batch repetitive tasks. This means that ILC has the same initial conditions for all iterations, revisit figure 1.2. On the contrary, repetitive control assumes conditions at a beginning of a period that are the result of the previous condition.

## 1.3 Problem formulation

In ILC research, the focus has always been on stability of the ILC controlled systems. If the ILC controlled system is unstable, it is useless. In practice, we might even require an ILC controlled system to have good learning transients. This means that we require a system to perform *'better and better'*, and not *'better-worse-better'*. (In [22], an example of a stable ILC controlled system is given, that reaches an error of about $10^{40}$ times the initial error before converging to 0.)

The notion of stability of ILC controlled systems in presence of model uncertainties, i.e., robust ILC, is less studied. Although we already mentioned that ILC is robust for unmodelled dynamics and nonlinearities, not many ILC control laws explicitly incorporate uncertainties in their designs. Therefore, it is impossible to make solid statements on robust stability/performance. For widespread use of ILC, these statements are essential.

With this is mind, we can formulate the following problem statement for this literature survey:

> **Give a critical overview of the current state of art of model uncertainty handling in the Iterative Learning Control framework, and investigate possibilities for future research.**

In order to do so, several questions can be distinguished with respect to this problem statement:

- What are the basics of ILC?
- What is the current state of art of robust ILC?
- How can, in general, model uncertainties be described?
- (Where) Do ILC and (robust) feedback control intersect?

## 1.4 Outline of this report

The rest of this report is organised as follows: in chapter 2 some basics of ILC are discussed. Next, two ingredients of robust feedback control are discussed, namely: representing model uncertainty in chapter 3, and robust feedback controller synthesis in chapter 4. In chapter 5, knowledge from both ILC and robust feedback control are joined in the discussion on robust ILC. Finally, some conclusions are drawn and recommendations for future research are made. This report also contains two appendices. The first gives a comprehensive treatment of the singular value decomposition and of norms for vectors, matrices, signals, and systems. The second contains an extended derivation of an equation used in chapter 4.

Chapter 2

# ILC Basics

In the previous chapter, we introduced ILC as a method for improving performance of a system, which is performing a repetitive task. Before studying robustness of ILC controlled systems, we need to define the class of systems and ILC controllers under study. This is the subject of the next section, in which we will mostly determine what we will not study, and why not. Subsequently, some notations are introduced and analysis tools are developed that enables us to study ILC in a context we prefer. Finally, we discuss existing methods for ILC controller synthesis.

## 2.1 Research boundaries

In section 1.1, we already sketched the basic idea of ILC, in which was mentioned that ILC uses information from previous trials: $f_{k+1} = ILC(f_k, e_k)$. This is referred to as first order ILC. When more previous trials are used, $f_{k+1} = ILC(f_k, f_{k-1}, \ldots, e_k, e_{k-1}, \ldots)$, this is called higher order ILC. However, when all disturbances are trial-invariant, higher order does not offer benefits in terms of performance ([37] and [49, Section 2.5]). Since we only consider trial invariant disturbances, we will focus on first order ILC only. (In [13] and [49, Section 3.6] it is shown that higher order ILC can offer benefits in presence of trial varying disturbances.)

Some researchers incorporate the measurement data from the current cycle to determine the output at that same trial (e.g., [3, 46]), e.g.: $f_{k+1} = ILC(f_k, e_{k+1}, e_k)$. Hence the name, current cycle ILC. Current cycle ILC considers simultaneous optimal feedback and ILC design in an attempt to improve the behaviour of the ILC controlled system. If we assume the system to be stable, or already stabilised by a feedback controller (see figure 2.1), incorporation of the current cycle is not necessary, and only complicates design. Therefore, current cycle ILC is beyond the scope of this literature survey.

Most of the results of ILC research were derived for linear time invariant (LTI) systems. However, various classes of nonlinear systems have been considered [51]. Of primarily interest are nonlinear systems representative for robotic manipulators. In this literature survey, we do not discuss nonlinear ILC control, for it is assumed that we can apply linear control theory on nonlinear systems. In feedback control design, linear theory is sometimes successfully applied

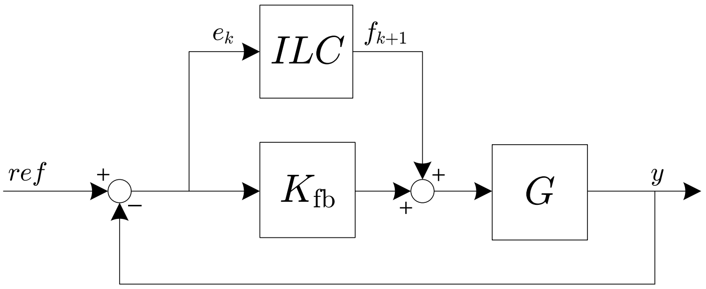


Figure 2.1: Block diagram of feedback control in conjunction with ILC

to nonlinear systems. Furthermore, in many cases, the ILC controller operates in conjunction with an existing feedback controller (see, e.g., [18, 22, 39]), which already gets the system output in the neighbourhood of the desired trajectory. This way, we can linearise the system around this trajectory, creating a linear time varying (LTV) system.

Another subject that has been left out of this literature survey is theory developed for continuous time. Since ILC requires a command signal and an error signal to be stored in a memory, a digital implementation is obvious. The design of the control law might as well acknowledge this for the start [8, Chapter 7], by being designed in discrete time.

Thus, in this literature survey, we will focus on the theory developed for stable, discrete time systems, both linear time-varying (LTV) and linear time-invariant (LTI). Moreover, we will study first order ILC only, and we do not to incorporate feedback control design into the ILC control design problem.

## 2.2 System description

In the previous section, we motivated for a specific branch of ILC systems operating on a finite time interval. Furthermore, all initial conditions are assumed to be 0. Throughout this remainder of this chapter, we consider the system:

$$J : \left\{ \begin{array}{rcl} x\left(T_s(t+1)\right) & = & Ax(T_st) + Bf(T_st) \\ y(T_st) & = & Cx(T_st) + Df(T_st) \end{array} \right. \tag{2.1}$$

where $T_s$ denotes the sample time. For simplicity of notation, $T_s$ is omitted in the rest of this document. The input-output behaviour can be described using a convolution of the input with the impulse response of the system:

$$y(t) = \sum_{\tau=0}^{t-1} CA^{t-\tau-1}Bf(\tau) + Df(t) \tag{2.2}$$

### 2.2.1 Lifted system description

Since we are dealing with a finite trial length $N$, thus $t \in \{0, 1, 2, \ldots, N-1\}$, we can evaluate (2.2) for every $t$:

$$\begin{array}{rl} y(0) & = Df(0) \\ y(1) & = CBf(0) + Df(1) \\ y(2) & = CABf(0) + CBf(1) + Df(2) \\ \vdots & \quad \vdots \\ y(N-1) & = CA^{N-2}Bf(0) + CA^{N-3}Bf(1) + \ldots + CBf(N-2) + Df(N-1) \end{array} \tag{2.3}$$

If we form a column vector from the time signal $y_k(t)$ at trial $k$, as first proposed in [42], we obtain a 'lifted' notation for $y_k$:

$$y_k = \left[ y_k^T(0), \quad y_k^T(1), \quad \ldots \quad , \quad y_k^T(N-1) \right]^T$$

We can define $f_k$ in a similar fashion. For a system having $l$ inputs and $n$ outputs, we can write (2.3) as a $(nN) \times (lN)$ dimensional linear system:

$$\begin{bmatrix} y_k(0) \\ y_k(1) \\ \vdots \\ y_k(N-1) \end{bmatrix} = \begin{bmatrix} D & 0 & \ldots & 0 \\ CB & D & & 0 \\ \vdots & \ddots & \ddots & \vdots \\ CA^{(N-2)}B & \ldots & CB & D \end{bmatrix} \begin{bmatrix} f_k(0) \\ f_k(1) \\ \vdots \\ f_k(N-1) \end{bmatrix} \tag{2.4}$$

Which has the form of $y_k = Jf_k$. We will refer to this system description as the 'lifted system description'. Note that $J$ has a block Toeplitz structure (see, e.g., [11]), and because $J$ is causal, has a lower triangular form. This notation allows us to use concepts from linear algebra to analyse ILC and design ILC controllers. Note that the first $l$ columns of the matrix $J$ contains the system's Markov parameters.

### 2.2.2 System description using $z$-transform

Sometimes, it can be more convenient to do the analysis of an ILC controlled system in the frequency domain. For that, we introduce the $z$-transform, which exists only for LTI systems:

$$Y(z) = \sum_{t=-\infty}^{\infty} y(t) z^{-t}$$

For causal signals and systems, $y(t) = 0$ for $t < 0$. In order to apply the $z$-transform, $N = \infty$ must be assumed, meaning that the signal is known over a infinite interval. Since ILC assumes a repetitive task, it is logical to assume a finite time interval. Therefore, an analysis in the $z$-domain yields an (conservative) approximation [40]. To obtain a frequency domain description of a $z$-transformed time signal or a transfer function, $z$ should be replaced by $e^{j\vartheta}$, with $\vartheta \in [-\pi, \pi]$.

Since a convolution in the time domain is equivalent to a multiplication in the $z$-domain, the output of the system of (2.2) at trial $k$ becomes:

$$Y_k(z) = J(z)F_k(z)$$

Note that, in order to make a proper distinction, the argument $z$ (or $e^{j\vartheta}$) is not omitted. Moreover, all $z$-transformed signals are written as capitals. Finally, the $z$-transform of (2.1) is given by:

$$J(z) = C(zI - A)^{-1}B + D$$

### 2.2.3 Singular Systems

Like any matrix, the lifted system description $J$ can be singular. This will occur if the system has delays or a relative degree unequal to zero. In discrete time, a relative degree of $m_r$ means that it takes $m_r$ steps before the result of an input is visible at the output. So, including a delay of $m_d$ steps, it takes $m = m_r + m_d$ steps before the output responds to an input. For the system matrix of (2.4), this means that the entries on the diagonal and on $m - 1$ subdiagonals of this matrix will become zero, resulting in a singular matrix.

The problem with a singular $J$ matrix is that the ILC controlled systems it is not fully observable. It is thus not possible to make the error $e(t) = 0 \ \forall \ t \in [0, N-1]$. Moreover, as we will see in the next section, $J$ needs to have full column rank in order to make the ILC controlled system asymptotically stable. We can solve this problem by redefining the system as follows:

$$J := J \begin{bmatrix} I \\ O \end{bmatrix}$$

where $I$ denotes an identity matrix of size $(nN - nm) \times (lN - lm)$ and $O$ a zero matrix of size $nm \times (lN - lm)$. Note that the redefined system $J$ is non-square, which can be done without loss of generality.

Another reason the system matrix $J$ becomes (nearly) singular, is when it has non-minimum phase zeros (NMPhZ) [48]. By the definition of zeros, there exist a initial condition $x(0)$ and a signal $f_k$ such that the output equals:

$$y_k = Jf_k + \mathcal{O}x(0) = 0$$

where $\mathcal{O}$ denotes the observability matrix $\mathcal{O} = \begin{bmatrix} C^T & (CA)^T & (CA^2)^T & \ldots \end{bmatrix}^T$ of the system (2.1). Let the singular value decomposition of $J$ be given by $U\Sigma V^T$ (see appendix A), then the singular value $\sigma$ associated to this zero is given by:

$$\min_i \sigma_i(J) < \frac{\|\mathcal{O}x_0\|_2}{\|f_k\|_2}$$

In case of a NMPhZ, $f_k$ corresponds to an exponential signal with the zero as exponential factor, whose amplitude will become larger as the trial length $N$ increases. As $N \to \infty$, $f_k(N) \to \infty$, and thus $\|f_k\|_2 \to \infty$. Therefore, the system matrix becomes nearly singular for longer trajectories, since:

$$\lim_{N\to\infty} \frac{\|\mathcal{O}x_0\|_2}{\|f_k\|_2} = 0$$

A general approach to solve the problem of a singular $J$ matrix is by considering the singular value decomposition, see appendix A. If $\Sigma_2$ corresponds to the singular values associated to the NMPhZ and delays, we can redefine $J$ so that it has full column rank:

$$J := JV_1$$

Because of this, the Toeplitz structure of $J$ is lost, which can be done without loss of generality.

### 2.2.4 Inverses in the $z$-domain

In the lifted description, the matrix $J$ becomes singular if the systems has a relative degree unequal to zero or if it has delays, and nearly singular if the system has NMPhZs. Problems with singularities arise when an inverse is needed. In the $z$-domain, a relative degree unequal to zero and NMPhZs also give problems when trying to invert the system. The inverse of a system with a relative degree unequal to zero would be a non-proper filter, which has high gain at high frequencies. Furthermore, an NMPhZ will result in an unstable filter. Both phenomena are not desirable.

These problems can be solved by using the ZPETC algorithm [47], which calculates a stable, proper approximation of the inverse of $J(z)$, such that the phase $\angle(J(z)J^{-1}(z)) = 0$. Given a $z$-domain representation of $J(z)$:

$$J(z) = z^{-m_d}\frac{B_s(z)B_u(z)}{A(z)}$$

where $m_d$ denotes the number of pure delays in the system and $B_s(z)$ and $B_u(z)$ are chosen in such a way that $B_u(z)$ contains all NMPhZ. The ZPETC algorithm gives the following approximation of the inverse:

$$J^{-1}(z) = z^{(m_d+phd)}\frac{A(z)B_\star(z)}{B_s(z)\beta} \qquad \text{with:} \qquad \frac{1}{B_u(z)} = \frac{z^{phd}B_\star(z)}{\beta}$$

where $\beta$ is chosen such that at low frequencies $J(z)J^{-1}(z) \approx 1$.

### 2.2.5 Using feedback control together with ILC

In section 2.1, we already mentioned that ILC can be used in conjunction with feedback control, either for stabilising the system, or linearising it. This makes it legitimate to assume the system, given by (2.1), to be stable. Given a single-input-single-output (SISO) system $G$ and a feedback controller $K_{\mathrm{fb}}$, revise figure 2.1, the error at trial $k$ is given by:

$$e_k = \frac{1}{1+GK_{\mathrm{fb}}} ref + \frac{G}{1+GK_{\mathrm{fb}}} f_k$$

It is possible to put this situation into the framework of figure 1.1 by defining:

$$r = \frac{1}{1+GK_{\mathrm{fb}}} ref, \qquad \text{and:} \qquad J = \frac{G}{1+GK_{\mathrm{fb}}}$$

where $ref$ is the original reference trajectory and $r$ is the trajectory as given in section 1.1.

## 2.3 ILC Analysis

In the previous section, notations have been introduced for describing the input/output behaviour of a LTI system in both the lifted domain and the $z$-domain. In this section, we will develop a learning control law for these two domains. After that, tools are developed for analysis of stability, performance, and transient learning behaviour of an ILC controlled system.

### 2.3.1 Learning Algorithm

The goal of ILC is to minimise the tracking error:

$$e_k = r - y_k \tag{2.5}$$

This reference trajectory $r$ can be seen as a constant disturbance in the trial domain. According to the internal-model-principle (IMP) [23], a persistent disturbance can be handled by incorporating a model of this disturbance in the feedback path, and adding an controllable input to this disturbance model. In this case, the disturbance is a constant disturbance, which can be modelled as an integrator. A widely used [10, 32] ILC control law is therefore given by:

$$f_{k+1} = Q(f_k + Le_k) \tag{2.6}$$

In this equation, both $Q$ and $L$ can be full matrices, thus representing possibly noncausal filters. This control law fulfils the IMP, when $Q = I$. When the underlying system of $Q$ has, e.g., a low-pass characteristic, the IMP is only valid for the parts of the spectrum of $r$ where the magnitude of the system describing $Q$ equals 1.

In a similar fashion, an ILC control law of (2.6) can be given in the $z$-domain:

$$F_{k+1}(z) = Q(z)\left(F_k(z) + L(z)E_k(z)\right)$$

where $E(z) = R(z) - Y_k(z)$ denotes the error. There is a fundamental difference between the ILC law in the lifted domain and the $z$-domain. In the lifted domain, $Q$ and $L$ are not required to be Toeplitz, thus they can represent time varying filter operations. Since the $z$-domain only exists for LTI systems, $Q(z)$ and $L(z)$ are (possibly noncausal) time invariant filters. This implies that the design freedom in the $z$-domain is more restricted than the lifted domain.

Using (2.4), (2.5), and (2.6), a block diagram can be defined, see figure 2.2, showing that ILC can be seen as feedback control in the iteration domain. In this diagram $w^{-1}I$ denotes the one-trial-shift operator, thus: $f_k = w^{-1}f_{k+1}$. This lifted-domain equivalent of an integrator can also have an initial value $f_0$, which corresponds to a feedforward signal at trial 0.

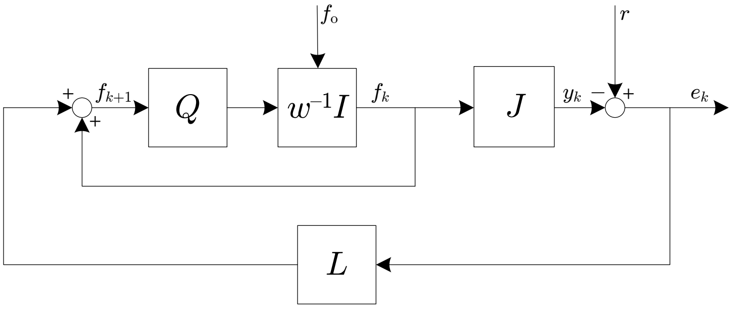


Figure 2.2: ILC as a feedback problem in the iteration domain.

### 2.3.2 Asymptotic stability and monotonic convergence

For both the lifted domain and the $z$-domain, we can obtain conditions for asymptotic stability. An ILC system is called asymptotically stable if there exists an $\tilde{f} \in \mathbb{R}$, such that:

$$\lim_{k\to\infty} |f_k(t)| \leq \hat{f}, \qquad \forall\, t \in [0, 1, \ldots N-1]$$

**Proposition 1** ([40, 10])**.** *In the lifted domain, the ILC controlled system is asymptotically stable if and only if:*

$$\rho\left(Q(I - LJ)\right) \equiv \max_i |\lambda_i(Q(I - LJ)| < 1 \tag{2.7}$$

*where $\rho$ denotes the spectral radius and $\lambda_i$ the i-th eigenvalue of the matrix $Q(I - LJ)$.*

*Proof.* Substituting (2.4) into (2.6) yields:

$$f_{k+1} = Q(f_k + Le_k) = Q(I - LJ)f_k + QLr \tag{2.8}$$

This difference equation is stable if and only if all eigenvalues of $Q(I - LJ)$ are within the unit circle. □

In section 1.3, we already mentioned that, although the ILC system is asymptotically stable, a practically undesirable learning transient can occur. Examples given in [10] and [22] show an increase of the error in the first iterations, before it starts converging to zero. To avoid this kind of behaviour, we can desire monotonic convergence of the input, i.e.:

$$\|f_\infty - f_{k+1}\|_i \leq \gamma \|f_\infty - f_k\|_i \tag{2.9}$$

where $\| \bullet \|_i$ denotes a matrix norm, see appendix A, with $i \in \{1, 2, \ldots, \infty\}$, and where $0 \leq \gamma < 1$ denotes the convergence rate. If we design the ILC controller such that $\gamma = 0$, dead-beat convergence is obtained, meaning that the error converges in one iteration.

**Proposition 2** ([40, 10, 32])**.** *The norm of $\|f_\infty - f_k\|_i$, with $i \in \{1, 2, \ldots, \infty\}$, converges monotonically with convergence rate $\gamma$, if:*

$$\|Q(I - LJ)\|_i \leq \gamma \tag{2.10}$$

*Proof.* If we evaluate (2.8) at $k+1$ and at $k \to \infty$ and subtract these from each other, this equals:

$$\begin{aligned} f_{k+1} - f_\infty &= Q(I - LJ)f_k + QLr - (Q(I - LJ)f_\infty + QLr) \quad \Leftrightarrow \\ &= Q(I - LJ)(f_k - f_\infty) \end{aligned} \tag{2.11}$$

This can be written in the form of (2.9) if we take the norm and apply the multiplicative property of matrix norms, see appendix A. □

In the $z$-domain, we can also derive a condition for stability in a similar to proposition 1. However, it is not possible to calculate the spectral radius from a transfer function. In order to make a statement, the following lemma is used.

**Lemma 1** (e.g., [43])**.** *The spectral radius $\rho$ of a (transfer function) matrix $A$ is bounded by:*

$$\rho(A) \leq \|A\|_i \tag{2.12}$$

*where $\| \cdot \|_i$ denotes any matrix norm with $i \in \{1, 2, \ldots, \infty\}$.*

*Proof.* It follow from the eigenvalue $\lambda$ and eigenvector $x$ of the matrix $A$ that:

$$\begin{aligned} \lambda x &= Ax & \Rightarrow \\ |\lambda| \|x\|_i = \|\lambda x\|_i &= \|Ax\|_i \leq \|A\|_i \|x\|_i & \Rightarrow \\ |\lambda| &\leq \|A\|_i \end{aligned}$$

Since this holds for any eigenvalue of matrix $A$, and hence also for the maximum eigenvalue, the spectral radius is bounded by any matrix norm of $A$. □

**Proposition 3** ([40, 10])**.** *In the $z$-domain, the ILC controlled system is guaranteed to be asymptotically stable if:*

$$\|Q(z)\,(1 - L(z)J(z))\,\|_i < 1 \qquad \textit{with:} \qquad i \in \{1, 2, \ldots \infty\} \tag{2.13}$$

*Proof.* Follows from proposition 1 and lemma 1. □

Note that in the $z$-domain, the sufficient condition for asymptotic stability is corresponds to the sufficient condition for monotonic convergence in the lifted domain.

### 2.3.3 Performance

The performance of an ILC system is based on the value of the error as $k \to \infty$. If the ILC system is asymptotically stable, $e_k \to e_\infty$, and $f_k, f_{k+1} \to f_\infty$, the lifted domain description (2.8) becomes:

$$\begin{aligned} f_\infty &= Q(f_\infty - LJf_\infty) + QLr \quad \Leftrightarrow \\ &= (I - Q(I - LJ))^{-1} QLr \end{aligned}$$

Substituting this into (2.5) results in an expression for the steady state error.

**Proposition 4** ([40, 10, 32])**.** *If the ILC controlled system is asymptotically stable, the error converges to:*

$$e_\infty = \left(I - J\left(I - Q(I - LJ)\right)^{-1} QL\right) r \tag{2.14}$$

*given a lifted domain description of $J$, $Q$, and $L$.*

Note that for arbitrary $r$ and (2.6), the error converges to 0 if and only if $Q = I$ and $J$ is square and of full column rank.

In the $z$-domain, we can derive an expression for the converged error, similar to proposition 4.

**Proposition 5** ([40, 10])**.** *If a SISO ILC controlled system is asymptotically stable, the error converges to:*

$$E_\infty(z) = \frac{1 - Q(z)}{1 - Q(z)\left(1 - L(z)J(z)\right)} R(z) \tag{2.15}$$

*Proof.* Can be derived from proposition 4. We can rewrite (2.14) in terms of transfer functions:

$$E_\infty(z) = \left(1 - \frac{J(z)Q(z)L(z)}{1 - Q(z)\left(1 - L(z)J(z)\right)}\right) R(z) = \frac{1 - Q(z)}{1 - Q(z)\left(1 - L(z)J(z)\right)} R(z) \qquad \square$$

## 2.4 Disturbance aspects

An ILC controller has the ability to perform satisfactory while exposed to unknown disturbances or while designed using an erroneous system model. In this section, we consider the consequences of repetitive and non-repetitive disturbances, repetitive and non-repetitive initial conditions, and model uncertainty on the performance and the stability of the ILC controlled system.

### 2.4.1 Repetitive and non-repetitive disturbances

One of the key features of ILC is the ability to deal with (unknown) repetitive disturbances. It is possible to analyse the effect of both trial-varying and trial-invariant disturbances by incorporating them into error equation (2.5). Given an input disturbance $w$ and measurement noise $v$, the error equation becomes:

$$e_k := r - Jw_k - v_k - Jf_k$$

From this equation, it stems that trial invariant disturbances have the same effect as a reference trajectory, and thus can be handled by ILC. Trial varying disturbances are more difficult to deal with, and is subject of current research. If we apply the theory developed in [31] in the lifted domain, we obtain:

$$e_k = (I-JQL)r - Jw_k + JQLJw_{k-1} - v_k + JQLv_{k-1} - \sum_{j=1}^{k-2} J[Q(I-LJ)]^{k-j-1}QL(r - Jw_j - v_j)$$

From this equation, we see that if $w_k \neq w_{k-1}$, or $v_k \neq v_{k-1}$, the error increases and, hence, performance deteriorates. In literature, proposals have been made to handle this problem. Some contributions are: [38], which proposes specific choices for the $Q$ filter, [13] which employs higher order ILC, and [36] in which Kalman filters are used.

### 2.4.2 Trial invariant initial states and trial varying initial states

So far, we assumed that all initial states are zero. However, there is no guarantee that this is always the case. If we consider a non zero initial state $x_k(0)$ at trial $k$, the lifted system description becomes of (2.4) becomes:

$$y_k = Jf_k + \mathcal{O}x_k(0)$$

where $\mathcal{O}$ denotes the observability matrix, which was defined before. We can analyse this situation by considering the error:

$$e_k = r - Jf_k - \mathcal{O}x_k(0)$$

From this equation, it stems that trial invariant initial conditions can be treated as disturbances. They can be fully dealt with if they are trial invariant and if the relative degree of the system $m = 0$. This is due to the the fact that $y_k(0)$ can only be influenced by $f_k(0)$, and hence can cancel $Cx_k(0)$ if and only if the $D$-matrix of (2.1) has full row rank. The effects of trial varying initial conditions can be analysed similar to that of trial varying disturbances.

### 2.4.3 Model uncertainty

In this section, we showed that ILC is capable of dealing with trial invariant disturbances. Trial invariant disturbances and non zero initial conditions affect the converged error, and thus performance of the ILC controlled system negatively. Although loss of performance is not desirable, stability is maintained while exposed to these effects. By giving a simple example, we can show that model uncertainty is able to make the ILC controlled system unstable. Consider a learning filter $L = J^{-1}$ (assuming it exists), which was designed using a model of the system $J$. If the real system behaves as $\hat{J} = J(I + \Delta)$, where $\Delta$ denotes an unknown but bounded perturbation, the stability condition of proposition 1 becomes:

$$\rho(Q(I - L\hat{J})) = \rho(Q\Delta) < 1 \tag{2.16}$$

which is not guaranteed to be true for a (poorly) designed $Q$ and $L$ given a $\Delta$. This is why model uncertainty should be incorporated in the synthesis of an ILC controller.

## 2.5 Learning controller synthesis

In the previous sections, the required notation and analysis tools have been introduced in both lifted and $z$-domain. In this section, some commonly used learning filters will be discussed, together with their stability properties. We restrict ourselves to the design of learning controllers in the lifted domain, because this is the preferred domain.

### 2.5.1 Arimoto-like learning filter

The most simple learning filter is the algorithm originally proposed by Arimoto [5]. An implementation in discrete time is discussed in, e.g., [10, 22, 39]. The key issue in this method is that it requires only limited knowledge of the system. For a system with relative degree $m$, the following learning algorithm is proposed:

$$f_{k+1}(t) = f_k(t) + \varphi e_k(t + m)$$

where $\varphi$ is a learning gain. In the lifted framework, this corresponds to $Q = I$, $J$ defined according to section 2.2.3, and $L$ a matrix with entries on the $m$-th superdiagonal. For instance, if $m = 1$, the system $J$ and the learning matrix $L$ become:

$$J = \begin{bmatrix} 0 & \dots & 0 \\ CB & \ddots & \vdots \\ \vdots & \ddots & 0 \\ CA^{N-2}B & \dots & CB \end{bmatrix} \in \mathbb{R}^{N\times(N-m)}, \qquad \text{and:} \qquad L = \begin{bmatrix} 0 & \varphi & & \\ & \ddots & \ddots & \\ & & 0 & \varphi \end{bmatrix} \in \mathbb{R}^{(N-m)\times N}$$

**Asymptotic stability and monotonic convergence:** Because $J$ is lower triangular Toeplitz, with the first $m-1$ subdiagonals parameters equal to zero, and $L$ has entries only at the $m$-th superdiagonal, the product $LJ \in \mathbb{R}^{(N-m)\times(N-m)}$ is lower triangular. Therefore, the ILC system is asymptotically stable if and only if:

$$\rho(I - LJ) = \begin{cases} |1 - CA^{m-1}B\varphi| < 1 & \text{for:} \quad m = 1, 2, \dots \\ |1 - D\varphi| < 1 & \text{for:} \quad m = 0 \end{cases} \tag{2.17}$$

A problem with the algorithm is that monotonic convergence is in general not guaranteed. In [33] it is shown that for specific systems monotonic convergence can be guaranteed. Given a $\varphi$ that stabilises the ILC system, the $\infty$-norm of the error converges monotonically if:

$$|1 - \varphi CA^{m-1}B| < 1 - |\varphi| \sum_{j=m}^{N-2} |CA^j B| \tag{2.18}$$

Other possible solutions to obtain monotonic convergence are: adding a (possibly zero-phase) $Q$ filter [39], using a lower sampling rate, so that the system has the properties of (2.18) [52], or adding more learning gains, [22]:

$$u_{k+1}(t) = u_k(t) + \varphi e_k(t+m) + \psi e_k(t+m-1) + \dots$$

### 2.5.2 Inverted plant learning filter

If a lifted domain model of the system is available, it is possible to use that model for designing a learning filter. Ideally, the learning filter would be the inverse of the system, but the system can be singular. This can be solved as described in section 2.2.3. The resulting learning filter becomes the pseudo-inverse (Moore-Penrose inverse) [6], given by:

$$L = \alpha(J^T J)^{-1} J^T, \qquad \text{with:} \qquad 0 < \alpha < 2 \tag{2.19}$$

**Asymptotic stability and monotonic convergence:** Proposition 2 gives a sufficient condition for monotonic convergence, substituting (2.19) gives:

$$\|Q\left(I - \alpha(J^T J)^{-1} J^T J\right)\|_i = |1 - \alpha| \|Q\|_i < 1$$

for any norm $i \in \{1, 2, \dots, \infty\}$. For $\alpha = 1$, the convergence rate $\gamma = 0$. This means that the ILC controlled system converges in one iteration.

### 2.5.3 Norm-optimal learning filter

ILC controller design can be approached using optimal control theory in the lifted domain, see [3, 18, 25, 28], using a quadratic criterion:

$$\mathcal{J} = e_{k+1}^T W_e e_{k+1} + f_{k+1}^T W_f f_{k+1} \tag{2.20}$$

where $W_e$ and $W_f$ are symmetric, positive semi-definite weighting matrices. Using these matrices, a trade-off can be made between performance (weight on $e_{k+1}$), and control effort (weight on $f_{k+1}$). This criterion can be subject to the constraint: $(f_{k+1} - f_k)^T(f_{k+1} - f_k) \leq \epsilon$. This way, we can limit the change of controller output for iteration to iteration. This constraint can be added to the criterion by introducing the Lagrange multiplier $\beta$. Furthermore, using (2.4) and (2.5), this cost function becomes:

$$\mathcal{J}' = (r - Jf_{k+1})^T W_e (r - Jf_{k+1}) + f_{k+1}^T W_f f_{k+1} + \beta((f_{k+1} - f_k)^T(f_{k+1} - f_k) - \epsilon)$$

This quadratic criterion has a global minimum where:

$$\frac{\partial \mathcal{J}'}{\partial f} = 2\left(-J^T W_e r + (J^T W_e J + W_f) f_{k+1} + \beta(f_{k+1} - f_k)\right) = 0$$

Rewriting this and using the fact that $r = e_k + Jf_k$, yields:

$$f_{k+1} = (J^T W_e J + W_f + \beta I)^{-1} \left((J^T W_e J + \beta I) f_k + J^T W_e e_k\right)$$

This can be put in the framework of (2.6), if we define:

$$Q = (J^T W_e J + W_f + \beta I)^{-1}(J^T W_e J + \beta I) \tag{2.21}$$

and:

$$L = (J^T W_e J + \beta I)^{-1} J^T W_e \tag{2.22}$$

In literature, several choices for $W_f$, $W_e$ and $\beta$ have been proposed. Choosing $W_f = 0$, implying $Q = I$, results in the solution found in [25]. According to proposition 4, this can yield convergence to zero error. Furthermore, if we also choose $W_e = I$, the result is equivalent to that discussed in [18]. Choosing $W_e = I$ and $\beta = 0$, yields: $L = (J^T J)^{-1} J^T$, which is the same as the learning filter discussed in the previous section.

**Asymptotic stability and monotonic convergence:** Using proposition 2, and both (2.21) and (2.22), a sufficient condition for monotonic convergence is obtained:

$$\| \left(J^T W_e J + W_f + \beta I\right)^{-1} \beta I \|_2 < 1 \tag{2.23}$$

Because $J^T W_e J + W_f$ is symmetric, positive definite (eigenvalues equal singular values), we can write the singular value decomposition (SVD) of:

$$U \Sigma U^T = J^T W_e J + W_f$$

Substituting this into (2.23) yields:

$$\| \left(U \Sigma U^T + \beta I\right)^{-1} \beta I \|_2 = \| (\Sigma + \beta I)^{-1} \beta I \|_2 < 1$$

Because $\Sigma$ and $\beta I$ are diagonal, this condition is equivalent to:

$$\frac{\beta}{\beta + \sigma_i} < 1 \quad \forall\, i \tag{2.24}$$

where $\sigma_i$ is the $i$th diagonal element of $\Sigma$. Because $\sigma_i(J^T W_e J + W_f)$ is positive, and $\beta$ is positive, (2.24) is always fulfilled.

What can be seen from (2.24), is that when $W_e = I$, $W_f = 0$, the learning filter becomes an approximate inverse of the system. For the singular values of the system $\sigma_i(J) \gg \beta$, the convergence rate $\gamma \approx 0$. Therefore, the part of the error which belongs to these singular values converges almost to zero in the first iteration. For singular values $\sigma_i(J) \ll \beta$, the convergence rate $\gamma \approx 1$. The part of the error associated to these singular values converge more slowly.

### 2.5.4 Low Order learning filter

The solutions found in the previous two sections have a major drawback. For a trajectory of length $N$, the learning filter becomes a $N \times N$ matrix. Hence, for large trajectories, the calculations become numerically extensive. In [19, 20], an alternative is presented, which is discussed in more detail in [18, Section 4.7]. This method is based on the fact the output of the learning filter $u = Le$ can also be obtained by simulation using the underlying difference equation.

If we consider the solution of the previous section and choose $W_e = I$ and $W_f = 0$, the optimal solution is given by:

$$u = (J^T J + \beta I)^{-1} J^T e$$

The output $u$ can be obtained in two steps, see figure 2.3. The first one is calculating $\tilde{e} = J^T e$ and the second is calculating $u = (J^T J + \beta I)^{-1} \tilde{e}$.

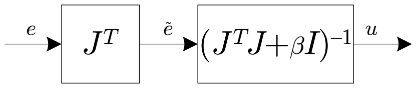


Figure 2.3: Graphical representation of $u = (J^T J + \beta I)^{-1} J^T e$

Revisit (2.1), if we assume the system $J$ to be strictly proper, it can be described using the following difference equation:

$$J : \begin{cases} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{cases} \tag{2.25}$$

On the finite interval $t \in \{0, 1, \ldots, N-1\}$, $\tilde{e}$ can be obtained by solving the difference equation describing $J^T$:

$$J^T : \begin{cases} \mu(t) &= A^T \mu(t+1) + C^T e(t) \\ \tilde{e}(t) &= B^T \mu(t+1) \end{cases} \tag{2.26}$$

given a final value $\mu(N) = 0$. Note that this is an anti-causal recursion, which can be solved, since all calculations are done after each trial and $e(t)$ is known for the entire interval.

**The Hamiltonian system**

The second step, solving $u = (J^TJ + \beta I)^{-1}\tilde{e}$ is more cumbersome. Considering (2.25) and (2.26), an expression for $(J^TJ + \beta I)^{-1}$ can be obtained:

$$(J^TJ+\beta I)^{-1}\begin{cases} x(t+1) &= Ax(t) - \beta^{-1}BB^T\gamma(t+1) + \beta^{-1}B\tilde{e}(t) \\ \gamma(t) &= C^TCx(t) + A^T\gamma(t+1) \\ u(t) &= -\beta^{-1}B^T\gamma(t+1) + \beta^{-1}\tilde{e}(t) \end{cases} \tag{2.27}$$

In this expression, we find the structure of a Hamilton-Jacobi-Bellman (HJB) equation, which is also found in optimal feedback control theory (see, e.g., [12, 24]). It consists of a set of two linked difference equations, with a causal recursion in the state $x$, and an anticausal recursion in the costate $\gamma$. If we define $R = \beta I$, the state propagation part of (2.27) becomes:

$$\begin{bmatrix} x(t+1) \\ \gamma(t) \end{bmatrix} = \begin{bmatrix} A & -BR^{-1}B^T \\ C^TQC & A^T \end{bmatrix} \begin{bmatrix} x(t) \\ \gamma(t+1) \end{bmatrix}$$

which is exactly the HJB-equation belonging to the LQ optimal state feedback control problem. This raises questions on the equivalences and differences between feedback control and ILC control. In this literature survey, no attempts will be made to answers such questions, but it could be a direction of future research.

If we rewrite (2.27) as a forward recursion, we obtain:

$$\begin{aligned} \begin{bmatrix} x(t+1) \\ \gamma(t+1) \end{bmatrix} &= \begin{bmatrix} A+\beta^{-1}BB^TA^{-T}C^TC & -\beta^{-1}BB^TA^{-T} \\ -A^{-T}C^TC & A^{-T} \end{bmatrix} \begin{bmatrix} x(t) \\ \gamma(t) \end{bmatrix} + \begin{bmatrix} \beta^{-1}B \\ 0 \end{bmatrix} \tilde{e}(t) \\ u(t) &= \begin{bmatrix} \beta^{-1}B^TA^{-T}C^TC & -\beta^{-1}B^TA^{-T} \end{bmatrix} \begin{bmatrix} x(t) \\ \gamma(t) \end{bmatrix} + \beta^{-1}\tilde{e}(t) \end{aligned} \tag{2.28}$$

Since all eigenvalues of $A$ are within the unit circle, (2.28) has all eigenvalues $|\lambda_i| < 1$ in the state equation $x$, and all eigenvalues $|\lambda_i| > 1$ in the costate $\gamma$. It is possible to decouple the stable and unstable part of (2.28) by transforming the system under similarity using the following transformation matrix:

$$T = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} = \begin{bmatrix} I & O \\ \Upsilon & I \end{bmatrix} \begin{bmatrix} I & P \\ O & I \end{bmatrix} \quad \text{and:} \quad T^{-1} = \begin{bmatrix} I & -P \\ O & I \end{bmatrix} \begin{bmatrix} I & O \\ -\Upsilon & I \end{bmatrix} \tag{2.29}$$

where $\Upsilon$ is given by the following algebraic Riccati equation:

$$\Upsilon = A^T\Upsilon A - A^T\Upsilon B(\beta - B^T\Upsilon B)^{-1}B^T\Upsilon A + C^TC$$

and $P$ by solving the following Sylvester equation:

$$Z_{11}P - PZ_{22} + Z_{12} = 0$$

where:

$$Z_{11} = (I + \beta^{-1}BB^T\Upsilon)^{-1}A, \qquad Z_{22} = Z_{11}^{-T}, \qquad Z_{12} = -\beta^{-1}BB^TA^{-T}$$

Using the transformation matrix of (2.29), (2.28) becomes:

$$\begin{aligned} \begin{bmatrix} z(t+1) \\ \tilde{z}(t+1) \end{bmatrix} &= \begin{bmatrix} Z_{11} & 0 \\ 0 & Z_{22} \end{bmatrix} \begin{bmatrix} z(t) \\ \tilde{z}(t) \end{bmatrix} + T^{-1} \begin{bmatrix} \beta^{-1}B \\ 0 \end{bmatrix} \tilde{e}(t) \\ u(t) &= \begin{bmatrix} \beta^{-1}B^TA^{-T}C^TC & -\beta^{-1}B^TA^{-T} \end{bmatrix} T \begin{bmatrix} z(t) \\ \tilde{z}(t) \end{bmatrix} + \beta^{-1}\tilde{e}(t) \end{aligned}$$

with $Z_{11}$ and $Z_{22}$ as defined above. This decoupled system can be simulated by solving a forward recursing in $z$ and a backward recursing in $\tilde{z}$, after selection of proper initial conditions.

### Initial conditions of the transformed Hamiltonian system

In order to simulate the transformed Hamiltonian system, proper initial conditions are required. Since the transformed state $z$ is simulated forward in time, an initial condition $z(0)$ is needed. Similarly, for the transformed costate a final condition $\tilde{z}(N)$ is needed. Since $x(0)$ and $\gamma(N)$ are given, the transformed boundary conditions are given by:

$$x(0) = T_{11}z(0) + T_{12}\tilde{z}(0), \tag{2.30}$$

$$\gamma(N) = T_{21}z(N) + T_{22}\tilde{z}(N) \tag{2.31}$$

In order to calculate $z(0)$ and $\tilde{z}(N)$ from these equations, we need expressions for $z(N)$ and $\tilde{z}(0)$. These can be obtained by:

$$z(N) = Z_{11}^N z(0) + \Phi_z(N) \tag{2.32}$$

$$\tilde{z}(0) = Z_{22}^{-N}\tilde{z}(N) + \Phi_{\tilde{z}}(0) \tag{2.33}$$

where:

$$\begin{aligned} \Phi_z(t) &= \sum_{\tau=0}^{N-1} Z_{11}^{t-\tau-1}(I + P\Upsilon)\beta^{-1}Be(\tau) \\ \Phi_{\tilde{z}}(t) &= -\sum_{\tau=0}^{N-1} Z_{22}^{t-\tau-1}\Upsilon\beta^{-1}Be(\tau) \end{aligned}$$

Combining (2.30), (2.31), (2.32), and (2.33) yields the following system of equations:

$$\begin{bmatrix} x(0) \\ \gamma(N) \\ \Phi_z(N) \\ \Phi_{\tilde{z}}(0) \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} & 0 & 0 \\ 0 & 0 & T_{21} & T_{22} \\ -Z_{11}^N & 0 & I & 0 \\ 0 & I & 0 & -Z_{22}^{-N} \end{bmatrix} \begin{bmatrix} z(0) \\ \tilde{z}(0) \\ z(N) \\ \tilde{z}(N) \end{bmatrix}$$

Inversion of this system gives the appropriate boundary conditions, which can be used to simulate the transformed Hamiltonian system. Note that the boundary conditions have to be determined between every two trials, since $\Phi_z(t)$ and $\Phi_{\tilde{z}}(t)$ are a function of $e_k(t)$.

### Stability and convergence

Given a value $\beta$, the output of the low order learning filter is the same as the output of the norm optimal learning filter with $W_e = I$ and $W_f = 0$. Therefore, the low order learning filter has the same stability and convergence properties as the norm optimal learning filter of the previous section.

## 2.6 Closure

In this chapter, we studied a specific class of ILC controlled systems. For that, notations have been introduced and analysis tools have been developed. Moreover, we showed that, contrary to other disturbance sources, model uncertainty can make the ILC controlled system unstable. Finally, we discussed some frequently used control strategies, together with their stability properties.

Chapter 3

# Modelling Uncertainty

In this chapter, methods for modelling uncertainty are discussed. Uncertainty modelling is a matured field of research, mostly applied in robust feedback controller design. Most of the information used in this chapter is gathered from [43, 53]. Besides methods for representing uncertainties, this chapter also discusses how to formulate a control problem involving uncertain models, and how to analyse stability and performance for these uncertain systems.

## 3.1 Uncertain models

In controller design, a model is often used for controller synthesis. Since a model provides a map between inputs and outputs, its quality is determined by how closely its response matches the response of the real system. In practice, it is never possible to describe a system using a finite set of differential equations. Some reasons for this are:

- System parameters are only known by approximation;
- Some parameters change when operating conditions change, due to nonlinearity or due to wear;
- Fast dynamics are difficult to model and are therefore neglected;
- An output measurement is never perfect and always contains noise.

Therefore, it is more likely that the behaviour of the real system is not described by a single model, but by a set of models. We call a controlled system robust stable (RS) if and only if every model in the set is stable. The problem of robust performance (RP) is investigating whether performance requirements are satisfied for every model in the set. The problem of RP will discussed later using the Structured Singular Value.

The notion of uncertainty is used to express the difference between the real system and its model. The complete set of models can be described using a combination of a nominal model and an uncertainty model. The way this uncertainty model is described, depends on the amount of structure in this uncertainty model. Basically, we can distinguish two classes of uncertainty, namely:

- **Parametric uncertainty:** Here the model structure (including the order) is known, but its parameters are unknown;
- **Unstructured (dynamic) uncertainty:** Here the uncertainty is described in the frequency domain, and can be used to describe missing dynamics.

Both of these will be discussed in this section. However, first we discuss the small gain theorem, since it is needed to derive conditions for RS.

**Theorem 1** (**Small Gain Theorem** (see, e.g., [53, 43]))**.** *Given a stable loop gain $L(z)$, the closed loop is stable if:*

$$\|L(e^{j\vartheta})\|_i < 1 \quad \forall\ \vartheta \in [-\pi, \pi] \tag{3.1}$$

*where $\|\bullet\|_i$ denotes any matrix norm, and hence, satisfies the multiplicative property of matrix norms.*

*Proof.* The Nyquist theorem states that if $L(z)$ is stable, the closed loop is stable if and only if the Nyquist plot of $\det(I + L(z))$ does not encircle the origin. This theorem is proven by showing the opposite: if $\det(I+L(z))$ encircles the origin, then there exists a $\epsilon \in [-1, 1]\backslash\{0\}$, such that $\det(I + \epsilon L(e^{j\vartheta'})) = 0$ at frequency $\vartheta'$. Since the determinant equals the product of all eigenvalues, there exists an eigenvalue $\lambda_i$:

$$\begin{array}{rcll}
\lambda_i(I + \epsilon L(e^{j\vartheta'}) & = & 0 & \Leftrightarrow \\
\lambda_i(L(e^{j\vartheta'})) & = & \frac{-1}{\epsilon} & \Rightarrow \\
|\lambda_i(L(e^{j\vartheta'}))| & \geq & 1 & \Leftrightarrow \\
\rho(L(e^{j\vartheta'})) & \geq & 1 &
\end{array}$$

Thus, if the spectral radius $\rho(L(e^{j\vartheta})) \leq 1$, at some $\vartheta'$, the Nyquist plot of $\det(I + L(z))$ could encircle the origin. Hence, if the spectral radius $\rho(L(z)) < 1\ \forall\ z = e^{j\vartheta}$ then $\det(I + L(z))$ cannot encircle the origin, thereby providing a sufficient condition for stability. Using lemma 1 expression (3.1) is obtained. □

## 3.2 Unstructured uncertainty

Consider a set of possible perturbed system models $\Pi$, with a nominal design model $G(z) \in \Pi$ and a perturbed model $G_p(z) \in \Pi$. This perturbed model can be described using a combination of a nominal model and the uncertainty description using a perturbation:

$$\|\Delta(z)\|_\infty \leq 1$$

where $\Delta(z)$ is stable and of appropriate dimensions. Using stable, minimum phase, weighting filters $W_1(z)$, and $W_2(z)$, the uncertainty can be scaled appropriately. In figure 3.1, several

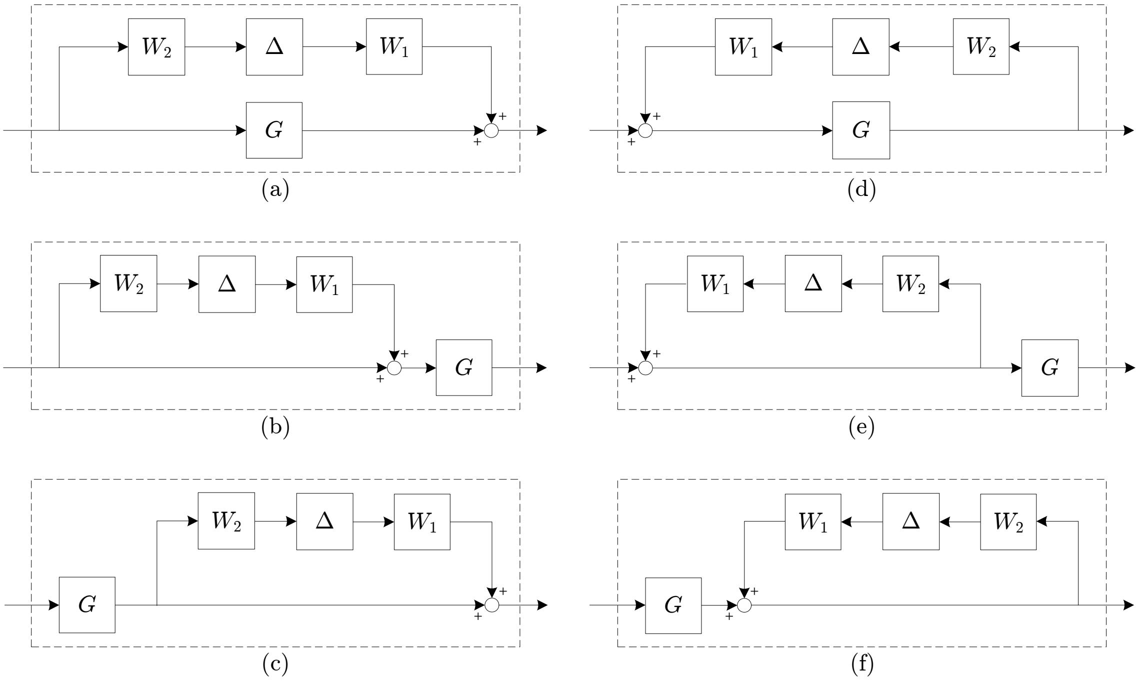


Figure 3.1: Forms of unstructured uncertainty: (a) additive uncertainty, (b) multiplicative input uncertainty, (c) multiplicative output uncertainty, (d) inverse additive uncertainty, (e) inverse multiplicative input uncertainty, (f) inverse multiplicative output uncertainty

forms of uncertain models are given schematically, which will be discussed below. Each form suits specific kinds of uncertainty as will be discussed below. If the system $G(z)$ is a SISO system, the multiplications commute, and we can simply take either $W_1(z) = 1$ or $W_2(z) = 1$.

For each form of uncertainty, we can derive a condition for RS. Given a controller $K(z)$ that stabilises the nominal system $P(z)$, let $M(z)$ denote the transfer between between the input and output of the perturbation $\Delta(z)$. According to the small gain theorem, the closed loop with loop gain $L(z) = M(z)\Delta(z)$ is stable if:

$$\|M(z)\Delta(z)\|_\infty \leq \|M(z)\|_\infty < 1$$

because $\|\Delta(z)\|_\infty < 1$ by definition. This gives a sufficient condition for RS. It can be shown that this condition is in fact also necessary, for a derivation see [43, 53]. Note that the multiplicative property of matrix norms is used here, which motivates why the $\mathcal{H}_\infty$ norm is used in robust control synthesis and analysis. The $\mathcal{H}_2$-norm is not induced and therefore, does not fulfil the multiplicative property (see appendix A). Therefore, it is not possible to judge how an interconnection of systems will behave, by evaluating the $\mathcal{H}_2$-norm of the individual components [43, Section 4.10.3].

### 3.2.1 Additive uncertainty

In figure 3.1(a) the perturbed system is in the form:

$$G_p = G + W_1 \Delta W_2$$

which explains the term 'additive uncertainty'. This form of uncertainty is often used to represent neglected high frequency dynamics. By choosing $W_1$ and $W_2$ appropriately, a greatest lower bound can be given to what extend the model is certain. Moreover, this form of uncertainty can be used to represent an uncertain NMPhZ. The necessary and sufficient condition for RS is given by

$$\|W_2 K S_o W_1\|_\infty < 1$$

where $S_o = (I + GK)^{-1}$ denotes the output sensitivity function.

### 3.2.2 Multiplicative Uncertainty

In figure 3.1(b,c) the perturbed system is represented in a multiplicative form, at either the input:

$$G_p = G(I + W_1 \Delta W_2)$$

or the output:

$$G_p = (I + W_1 \Delta W_2)G$$

This form can be used to represent uncertainties that happen specifically at the input or output of the system. Examples of these are actuator errors at the input and sensor (measurement) errors at the output. Note that if we are dealing with a SISO system, the multiplications commute, and the input and output multiplicative uncertainty are equivalent.

It is possible to convert an additive uncertainty description into a multiplicative uncertainty, and vise versa, be redefining the weighting functions. Given a weighting function for an additive uncertainty description $W_{1,A}$ and a weighting function for an input multiplicative uncertainty $W_{1,I}$, both the uncertainty descriptions are equivalent if:

$$|W_{1,I}| = |G^{-1}| \ |W_{1,A}|$$

where $| \bullet |$ denotes the magnitude of the frequency response of the transfer function. Equivalently, given a weighting function for an additive uncertainty description $W_{2,A}$ and a weighting function for an output multiplicative uncertainty $W_{2,O}$, these are also equivalent if:

$$|W_{2,O}| = |W_{2,A}| \ |G^{-1}|$$

Although both uncertainty descriptions are equivalent, there can be reasons to prefer one above the other. Sometimes, inverses of the weighting functions are needed and this requires the weighting functions to be proper but not strictly proper.

For an input multiplicative uncertainty description the necessary and sufficient condition for RS is given by:

$$\|W_2 K S_o G W_1\|_\infty < 1$$

And for the output multiplicative uncertainty, it is given by:

$$\|W_2 G K S_o W_1\|_\infty < 1$$

Note that these conditions confirm the aforementioned relation between additive and multiplicative uncertainty.

### 3.2.3 Inverse uncertainty

The last forms of unstructured uncertainty discussed in this section, shown in figure 3.1(c,d,e), are referred to as inverse uncertainty. We can distinguish inverse additive and inverse multiplicative forms. The inverse additive perturbed system is given by:

$$G_p = G(I - W_1 \Delta W_2 G)^{-1} \tag{3.2}$$

Its necessary and sufficient condition for RS is given by:

$$\|W_2 S_o G W_1\|_\infty < 1$$

The inverse multiplicative uncertainty description at the input and the output are given by:

$$G_p = G(I - W_1 \Delta W_2)^{-1} \qquad \text{and:} \qquad G_p = (I - W_1 \Delta W_2)^{-1} G$$

respectively. It corresponding RS conditions are:

$$\|W_2 S_i W_1\|_\infty < 1 \qquad \text{and:} \qquad \|W_2 S_o W_1\|_\infty < 1$$

for the input and output inverse multiplicative uncertainty respectively, where $S_i = (I + KG)^{-1}$ denotes the input sensitivity function.

Whereas additive and multiplicative uncertainty descriptions can be used to cover uncertainty in high frequency dynamics and uncertain zeros, the inverse forms can be used to describe low frequency errors and uncertain (unstable) poles. For example, consider a first order SISO system with an uncertain pole somewhere within a circle of radius $r$ around point $p$. This form of uncertainty can be conveniently written with a inverse additive uncertainty:

$$G_p = \frac{1}{z + p + r\Delta} = \frac{1}{z + p}\left(1 + \frac{r}{z + p}\Delta\right)^{-1}$$

Since this is a SISO system, all multiplications commute, and considering (3.2), the weighting functions for the inverse additive uncertainty become: $W_1 = r$ and $W_2 = 1$.

## 3.3 Parametric uncertainty

Parametric uncertainties can be modelled in a similar way as the unstructured uncertainty: using a nominal model and an uncertainty description. So far, we assumed nothing more than that the perturbation $\Delta$ is a complex matrix with a bounded infinity norm. However, using a complex perturbation is not realistic when describing uncertainty in terms of system parameters. In a real system, the system parameters are real-valued. Hence, instead of a complex matrix, a perturbation block is needed in the form:

$$\Delta = \begin{bmatrix} \delta_1 & 0 & \dots \\ 0 & \delta_2 & \ddots \\ \vdots & \ddots & \ddots \end{bmatrix} \tag{3.3}$$

where $\delta_i$, with $i = 1, 2, \dots$, is a possibly repeating, real-valued perturbations, satisfying $|\delta_i| \leq 1$. Thus, parametric uncertainty requires a more structured perturbation block. Therefore, parametric uncertainty a is form of structured uncertainty. In this section, we discuss how parametric uncertainty can be used to describe uncertain parametric models, such as transfer functions and state space models.

### 3.3.1 Transfer function form

Parametric uncertainty of transfer functions can be described using (a combination of) uncertainty descriptions of figure 3.1 with the perturbation according to (3.3). If we reconsider the example of the pole uncertainty of section 3.2.3, and we restrict the $\Delta$ to be real valued, this is parametric uncertainty. One can think of a water vessel with an outflow, in which we want to control the water level. In this SISO system, the pole can be uncertain due to nonlinear fluid behaviour.

Describing parametric uncertainty using transfer functions works well in the aforementioned example and in some other simple situations. However, if the system has multiple sources of uncertainties or is a multiple-input-multiple-output (MIMO) system, using a transfer function form can be quite cumbersome. In this case, using a state-space description of the system can be more convenient.

### 3.3.2 State space form

Consider the following discrete-time, uncertain state space model:

$$G \; : \begin{cases} x(t+1) & = \; A_p x(t) + B_p u(t) \\ y(t) & = \; C_p x(t) + D_p u(t) \end{cases}$$

where $(A_p, B_p, C_p, D_p)$ represent the perturbed system matrices. The input-output behaviour is given the following transfer function matrix:

$$G_p(z) = C_p(zI - A_p)^{-1}B_p + D_p$$

In figure 3.2, a block diagram is shown of the perturbed system, assuming all system matrices can be parameterised in the following form:

$$A_p = A + \delta A, \qquad B_p = B + \delta B, \qquad C_p = C + \delta C, \qquad D_p = D + \delta D$$

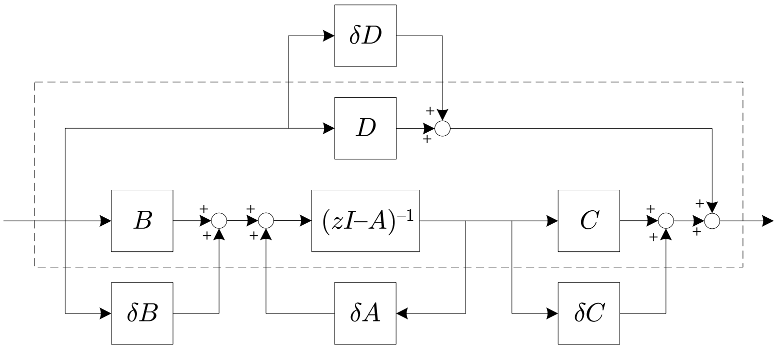


Figure 3.2: Parametric state space uncertainty

Using this parametrisation, it is possible to have uncertainty in each element of each system matrix. For example, consider a second order system, with a perturbed $A$-matrix, such that:

$$A_p = \begin{bmatrix} a_{11} + k_{11}\delta_1 & a_{12} + k_{12}\delta_2 \\ a_{21} + k_{21}\delta_3 & a_{22} + k_{22}\delta_4 \end{bmatrix}$$

where $a_{11}, a_{12}, a_{21}$, and $a_{22}$ denote the nominal values of matrix $A$, and the factors $k_{11}, k_{12}, k_{21}$, and $k_{22}$ are used to scale the real-valued perturbations $|\delta_1|, |\delta_2|, |\delta_3|, |\delta_4| < 1$. This can be put into a form of figure 3.1, by defining:

$$A_p = A + \underbrace{\begin{bmatrix} k_{11} & k_{12} & 0 & 0 \\ 0 & 0 & k_{21} & k_{22} \end{bmatrix}}_{W_1} \underbrace{\begin{bmatrix} \delta_1 & 0 & 0 & 0 \\ 0 & \delta_2 & 0 & 0 \\ 0 & 0 & \delta_3 & 0 \\ 0 & 0 & 0 & \delta_4 \end{bmatrix}}_{\Delta} \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}}_{W_2}$$

Perturbations of the $B_p$, $C_p$, and $D_p$ can be parameterised in a similar fashion. Furthermore, it is also possible to handle seemingly nonlinear parameters, such as $k\delta^2$, as shown in figure 3.3.

Obtaining weighing filters $W_1$ and $W_2$ and a perturbation block $\Delta$ for the complete perturbed model may seem complicated, certainly when dealing with large systems with lots of uncertain parameters. However, this procedure can be automated in MATLAB, using routines such as:

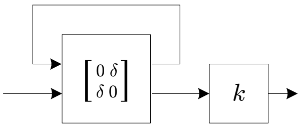


Figure 3.3: Block diagram for $k\delta^2$

`ureal, lftdata`. The routine `ureal` is used to create uncertain parameters, which can be used to form a so-called 'uncertain state-space model'. Then, the routine `lftdata` can be used to write the uncertain model as an upper linear fractional transformation of a $\Delta$-block and a state-space model that includes the nominal model and the uncertainty weight. The linear fractional transformation is discussed in the next section.

## 3.4 Linear Fractional Transformations

In robust feedback control, linear fractional transformations (LFTs) play an important role. Using LFTs, it is possible to to provide a framework for analysis and syntheses of feedback controllers, applicable to a variety of situations.

### 3.4.1 Definition of Linear Fractional Transformations

According to [21], let $M$ be a complex matrix partitioned as:

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \in \mathbb{C}^{(n_1+n_2)\times(m_1+m_2)}$$

and let $\Delta_u \in \mathbb{C}^{n_1\times m_1}$ and $\Delta_l \in \mathbb{C}^{n_2\times m_2}$. Then, the *upper* LFT is defined as the following map:

$$\mathcal{F}_u(M, \bullet) : \mathbb{C}^{n_1\times m_1} \to \mathbb{C}^{n_2\times m_2}$$

with:

$$\mathcal{F}_u(M, \Delta_u) \equiv M_{22} + M_{21}\Delta_u(I - M_{11}\Delta_u)^{-1}M_{12}$$

given the fact that the matrix $(I - M_{11}\Delta_u)$ is nonsingular. Furthermore, the *lower* LFT is defined as:

$$\mathcal{F}_l(M, \bullet) : \mathbb{C}^{n_2\times m_2} \to \mathbb{C}^{n_1\times m_1}$$

with:

$$\mathcal{F}_l(M, \Delta_l) \equiv M_{11} + M_{12}\Delta_l(I - M_{22}\Delta_l)^{-1}M_{21}$$

provided that the matrix $(I - M_{22}\Delta_l)$ is nonsingular. The formulae of the LFTs arise when describing a feedback system. The notions of upper and lower LFTs become clear when considering figure 3.4.

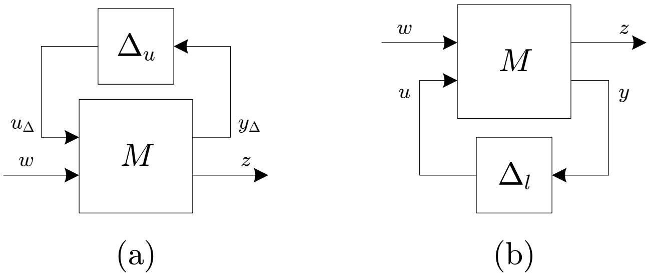


Figure 3.4: (a) Upper Linear Fractional Transformation, (b) Lower Linear Fractional Transformation

### 3.4.2 LFTs in feedback control and the generalised plant

Using LFTs it is possible to formulate robust control analysis and synthesis problems by using the so-called generalised plant, denoted $P(z)$. This generalised plant incorporates the system $G$ including performance weights, and uncertainty weights. In figure 3.5, a block diagram of the generalised plant is shown, which is clearly a combination of both a lower and an upper LFT.

In this diagram, the signals $u_\Delta$, $w$, $u$, $y_\Delta$, $z$, and $y$ denote vector valued functions of time or frequency. The exogenous inputs $w$ represents setpoints, disturbances, and measurement noise, and the outputs $z$ denote the performance variables, such as: output, error, etc. The overall control objective is to minimise the $\infty$-norm of closed loop transfer between $w$ and $z$, denoted $\|T_{zw}\|_\infty$, by finding a controller $K$, which uses measured outputs $y$ and calculates the input $u$.

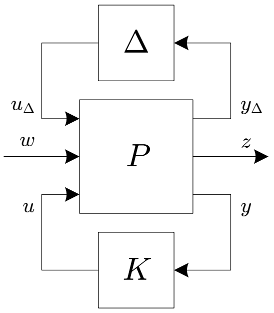


Figure 3.5: Generalised plant

The controller is RS if and only if the uncertainty block $\Delta$ does not destabilise the closed loop $T_{zw} = \mathcal{F}_l(P, K)$. This uncertainty block can be either unstructured (see section 3.2) or parametric (see section 3.3) of a combination of both.

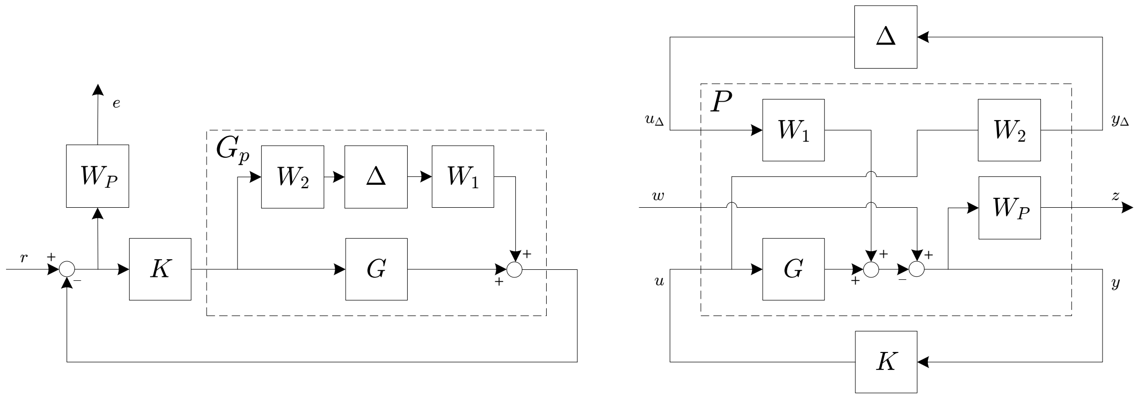
Given a generalised plant, simple MATLAB routines, such as `hinfsyn`, `dksyn`, can calculate a $\mathcal{H}_\infty$ optimal controller. The designers problem is to obtain a generalised plant, by selecting performance measures, weighting functions for these performance variables, and a proper uncertainty description including its weights. Since each uncertainty description and each control objective results in a different generalised plant, we illustrate the procedure by giving an example.

### Example: Servo control problem with additive uncertainty

Given an perturbed system $G_p(z)$ with an additive uncertainty description, revisit figure 3.1, with properly selected weights $W_1(z)$ and $W_2(z)$. Let's assume that the system is a servo system which has to track a reference trajectory with a minimal error. In terms of the generalised plant, the control objective is to minimise the weighted closed loop output sensitivity function:

$$\min_K \|W_P(I+GK)^{-1}\|_\infty$$

with $W_p(z)$ chosen to be the inverse of the desired closed loop output sensitivity. A block scheme of this control problem is drawn in figure 3.6(a).



(a) (b)

Figure 3.6: (a) Servo problem with an uncertain system, (b) The same problem formulated as an LFT

To obtain the generalised plant of figure 3.5, the elements of this block diagram have to be rearranged, see figure 3.6(b), which leaves the system unaltered. The transfer function matrix belonging to this generalised plant is given by:

$$P(z) = \left[\begin{array}{cc:c} 0 & 0 & W_2 \\ -W_pW_1 & W_p & -W_pG \\ \hdashline -W_1 & I & -G \end{array}\right]$$

With this generalised plant, an $\mathcal{H}_\infty$ optimal controller can be obtained using the standard MATLAB routine `hinfsyn`. The resulting closed loop can be calculated using a lower LFT:

$$\mathcal{F}_l(P,K) = \begin{bmatrix} 0 & 0 \\ -W_pW_1 & W_p \end{bmatrix} + \begin{bmatrix} W_2 \\ -W_pG \end{bmatrix} KS \begin{bmatrix} -W_1 & I \end{bmatrix} = \begin{bmatrix} -W_2KSW_1 & W_2KS \\ -W_pSW_1 & W_pS \end{bmatrix}$$

where $S = (I + GK)^{-1}$ denotes the output sensitivity function. From this closed loop, we can draw conclusions regarding nominal performance (NP) and robust stability (RS):

$$\begin{aligned} \text{NP:} \quad & \|W_p S\|_\infty < 1 \\ \text{RS:} \quad & \|W_2 K S W_1\|_\infty < 1 \end{aligned}$$

which is exactly as stated in section 3.2. Moreover, if we can model the uncertainty such that $W_1(z) = I$, the system has robust performance (RP) if:

$$\left\| \begin{matrix} W_2 K S \\ W_p S \end{matrix} \right\|_\infty < 1$$

## 3.5 The Structured Singular Value ($\mu$)

In section 3.2, we obtained conditions for RS for each form of unstructured uncertainty. In cases where there is structure in the perturbation matrix $\Delta$, e.g., when dealing with parametric uncertainty or when combining several forms of unstructured uncertainty, these conditions will be conservative. To reduce conservatism, we need a generalisation of the singular value, referred to as the Structured Singular Value (SSV) $\mu$. According to [21, 41], given a matrix $M \subset \mathbb{C}^{n \times n}$, and let $S$ and $F$ be a vector of nonnegative integers, and:

$$\boldsymbol{\Delta} = \{\mathrm{diag}[\delta_1 I_{k_1}, \ldots, \delta_S I_{k_S}, \Delta_1, \ldots, \Delta_F] \ : \ \delta_i \in \mathbb{C}, \ \Delta_j \in \mathbb{C}^{k_j \times k_j}\} \tag{3.4}$$

Then, the SSV is defined as:

$$\mu_\Delta(M) \equiv \frac{1}{\min\{\max_i \sigma_i(\Delta) \ : \ \Delta \in \boldsymbol{\Delta} \ : \ \det(I - M\Delta) = 0\}} \tag{3.5}$$

The SSV is the smallest possible largest singular value of $\Delta \in \boldsymbol{\Delta}$ that is able to destabilise the closed loop $I - M(z)\,\Delta(z)$. Note that the SSV is defined for a constant matrix $M$. For a transfer function matrix $M(z)$, the SSV has to be calculated for each frequency. If no $\Delta \in \boldsymbol{\Delta}$ makes $\det(I - M\Delta) = 0$, then $\mu_\Delta(M) \equiv 0$.

### 3.5.1 Some properties of the structured singular value

Before discussing applications of the SSV, two useful properties of the SSV are given in the form of propositions.

**Proposition 6** ([53]). *The SSV of a matrix $M$ is bounded by it spectral radius and its maximal singular value:*

$$\rho(M) \leq \mu_\Delta(M) \leq \max_i \sigma_i(M) \tag{3.6}$$

*Proof.* The first step is to prove the lower bound. For that, consider the extreme case that $\mathbf{\Delta} = \{\delta I \ : \ \delta \in \mathbb{C}\}$, hence, the set consists only of diagonal matrices. The only $\Delta$'s in the set that are able to make $\det(I - M\Delta) = 0$ are the reciprocals of the eigenvalues of $M$. For a diagonal matrix, the singular values are the absolute values of the eigenvalues. In this case, (3.5) becomes:

$$\mu_\Delta(M) = \frac{1}{\min_i |\lambda_i^{-1}(M)|} = \max_i |\lambda_i(M)| \equiv \rho(M)$$

The second step is to prove the upper bound. For that, consider the extreme case that: $\mathbf{\Delta} = \{\Delta \ : \ \Delta \in \mathbb{C}^{n \times n}\}$. From the Small Gain Theorem follows that $\det(I - M\Delta) \neq 0$ if $\max_i \sigma_i(M\Delta) < 1 \Rightarrow \max \sigma(\Delta) < \frac{1}{\max \sigma(M)}$. Using this, (3.5) becomes:

$$\mu_\Delta(M) = \frac{1}{\min\{\max_i \sigma_i(\Delta)\}} < \frac{1}{\min_j \sigma_j^{-1}(M)} = \max_j \sigma_j(M)$$

which proves the upper bound. □

This proposition gives bounds for the SSV. These bounds can however be large. In the following proposition, it is shown that the upper bound can be tightened.

**Proposition 7** ([53]). *Given a set $\mathcal{D} = \{D \ : \ D\Delta = \Delta D \ : \ \Delta \in \mathbf{\Delta}\}$, the upper bound of proposition 6 can be tightened to:*

$$\mu_\Delta(M) \leq \inf_{D \in \mathcal{D}} \max_i \sigma_i(DMD^{-1}) \tag{3.7}$$

*Proof.* Since the matrix $D \in \mathcal{D}$ commutes with $\Delta$, the following is true:

$$\Delta = D^{-1}D\Delta = D^{-1}\Delta D$$

This implies:

$$\det(I - M\Delta) = \det(I - MD^{-1}\Delta D) = \det(I - DMD^{-1}\Delta)$$

Hence, $\mu_\Delta(M) = \mu_\Delta(DMD^{-1})$. Since $\max_i \sigma_i(M) \neq \max_i \sigma_i(DMD^{-1})$, by changing $D \in \mathcal{D}$, the bound can be tightened. □

### 3.5.2 Robust stability analysis for structured uncertainty

Revisit the generalised plant from figure 3.5, in this sequel, the controlled system is defined by:

$$N = \mathcal{F}_l(P, K)$$

Moreover, let us assume that this controlled system has a parametric uncertainty description, see figure 3.7. In this case, $\mathbf{\Delta}$ is a diagonal matrix. By definition of the SSV, $\mu_\Delta \leq 1$ implies that the maximum singular value of $\Delta \in \mathbf{\Delta}$ has to be larger than one in order to destabilise the loop. Recall that the uncertainty descriptions in the generalised plant are such that the absolute value of their elements is bounded by one. Hence, if the SSV is smaller than one for all frequencies, the perturbation block of the generalised plant is not able to destabilise the loop.

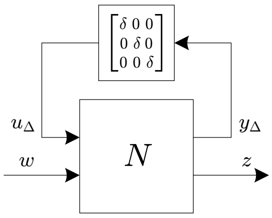

$\begin{bmatrix}\delta & 0 & 0\\ 0 & \delta & 0\\ 0 & 0 & \delta\end{bmatrix}$
$u_\Delta$
$y_\Delta$
$w$
$N$
$z$

Figure 3.7: Controlled system with structured uncertainty

If we combine the results of the reasoning with the results from proposition 6, we can make the following statement. Let $N_{11}$ denote the of the transfer function from $u_\Delta$ to $y_\Delta$. The controller system is RS if and only if:

$$\sup_{\vartheta} \rho\left(N_{11}(e^{j\vartheta})\right) < 1$$

### 3.5.3 Robust performance analysis

Formally, the RP analysis answers the question whether performance specifications are met for all possible perturbed systems. Given a controller $K$ and a generalised plant $P$, the $\infty$-norm of transfer function between $w$ and $z$, $N_{zw}$, has to be smaller than one for all possible $\Delta$'s. It is possible to pose this problem as a structured form of uncertainty, see figure 3.8, by including the performance-part in the uncertainty representation.

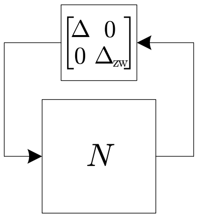


Figure 3.8: Robust performance analysis problem

Let the structure of the uncertainty description be defined by (3.4), and let:

$$\mathbf{\Delta}_P = \left\{ \begin{bmatrix} \Delta & 0 \\ 0 & \Delta_{zw} \end{bmatrix} \; : \; \Delta \in \mathbf{\Delta}, \Delta_{zw} \in \mathbb{C}^{q \times p} \right\}$$

where $\Delta$ can be further structured, since a structured form of model uncertainty can be used. $\Delta_{zw}$ is a full block perturbation which represents the requirement that the closed loop between $w$ and $z$ should smaller than one. The system is RP if and only if:

$$\sup_{\vartheta} \mu_{\Delta_P} \left( N(e^{j\vartheta}) \right) < 1$$

### 3.5.4 Robust performance synthesis

After showing how the notion of the SSV can be used to analyse RS and RP, this final part presents a method that can be used to synthese controller that has RP. Solving the RP problem directly using the SSV is, however, is not yet fully understood. A reasonable approach combines the results from proposition 7 and $\mathcal{H}_\infty$ synthesis, which will be discussed thoroughly in the next chapter.

In figure 3.9, the RP problem is posed using $D$-scales. Since the matrices $D \in \mathcal{D}$ are defined such that they commute with $\Delta$, the uncertainty representation in this figure is the same for all $D \in \mathcal{D}$. Changing the $D$-scales, however, changes the control synthesis problem. Since the SSV is bounded according to proposition 7, a reasonable solution to the $\mu$ synthesis problem is to solve:

$$\min_{K} \inf_{D \in \mathcal{D}} \| D \mathcal{F}_l(P, K) D^{-1} \|_\infty$$

by alternating solving for $K$ and $D$. This is called $DK$-iteration. For a fixed $D$, this is a standard $\mathcal{H}_\infty$ minimisation problem. Given a $K$, the minimisation problem can be solved pointwise in the frequency domain. Although both optimisations are convex, it is not guaranteed that the joint optimisation converges to a fixed point.

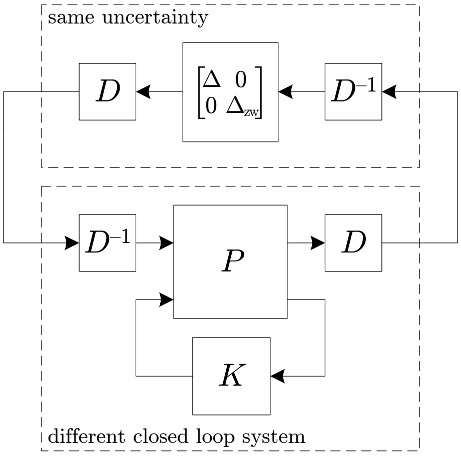


Figure 3.9: $\mu$-synthesis problem using $D$ scaling

## 3.6 Closure

This chapter discussed two forms of representing uncertainty and derived conditions for robust stability for each class. Moreover, it was shown how a proper control problem can be formulated using the generalised plant paradigm.

Although, unstructured uncertainty is modelled in the frequency domain, it can be used in the time domain. A state space realisation of the weighting filters can be easily obtained.

Chapter 4

# Robust Feedback Control

In the previous chapter, it was discussed how model uncertainties can be represented using LFTs, and that LFTs can provide a paradigm for formulating a control problem in a very general way. This chapter takes the next step in actually solving a robust feedback control problem. Note that the focus is on feedback control problem, because this branch of control theory has matured and elements of this could be used when solving a robust ILC control problem.

In this chapter, an introduction to optimal control is given. Subsequently, the $\mathcal{H}_\infty$ control problem is treated, which is an optimal control problem. In the previous chapter, we reasoned that, for robustness, $\mathcal{H}_\infty$ is preferred over $\mathcal{H}_2$, because of the small gain theorem and the multiplicative property of matrix norms.

Throughout this chapter, the following discrete-time system is considered, working on a finite time interval $t \in [0, N-1]$:

$$\left\{\begin{array}{lllll} x(t+1) & = & Ax(t) & +B_1w(t) & +B_2u(t) \\ z(t) & = & C_1x(t) & & +D_{12}u(t) \\ y(t) & = & C_2x(t) & +D_{21}w(t) & \end{array}\right. \tag{4.1}$$

where $x$, $w$, $u$, $z$, $y$ are defined in the same way as in the previous chapters. In order to simplify the expressions, some generality is sacrificed and it is assumed that:

$$\begin{bmatrix} B_1 & D_{21} \end{bmatrix} D_{21}^T = \begin{bmatrix} 0 & I \end{bmatrix} \qquad \text{and:} \qquad D_{12}^T \begin{bmatrix} C_1 & D_{12} \end{bmatrix} = \begin{bmatrix} 0 & I \end{bmatrix} \tag{4.2}$$

## 4.1 Introduction to optimal control

Optimal control, building on the optimal Wiener-Hopf filtering work, defines control problems as an optimisation problem, in which competing objectives are traded-off using a cost functional. Designing an optimal controller can be done in two steps, which we will discuss next. Moreover, we wills show how a control problem can be formulated as a constrained optimisation problem.

### 4.1.1 State feedback

The first step in designing an optimal controller is designing state feedback that is optimal with respect to a certain criterion. State feedback assumes all the states to be available for feedback. Hence, the controller has the form: $u(t) = -K(t)x(t)$. The closed loop is obtained by substituting this equation into (4.1), This yields:

$$\left\{\begin{array}{lcl} x(t+1) & = & (A - B_2K(t))\, x(t) \;+\; B_1w(t) \\ z(t) & = & (C_1 - D_{12}K(t))\, x(t) \end{array}\right. \tag{4.3}$$

The goal of controller design is the choose $K(t)$ such that the closed loop is stable and the transfer between $w(t)$ and $z(t)$ is minimal using some suitable notion of minimality.

### 4.1.2 Observers

Since state feedback requires all the states of the system to be available for feedback, a state observer is needed to make an estimation of the states. Such a state observer is given by:

$$\left\{\begin{array}{lcl} \hat{x}(t+1) & = & A\hat{x}(t) \;+\; B_2u(t) \;+\; L(t)\,(y(t) - C_2\hat{x}(t)) \\ \hat{z}(t) & = & C_1\hat{x}(t) \end{array}\right. \tag{4.4}$$

where $\hat{x}(t)$ denote the estimated values of the states and $\hat{z}(t)$ a linear combination of the states. The objective is to minimise the distance between $z$ and $\hat{z}$ using an appropriate measure of distance. Substracting (4.1) from (4.4) yields:

$$\left\{\begin{array}{lcll} (x-\hat{x})(t+1) & = & A(x-\hat{x})(t) & + B_1w(t) \;-\; L(t)C_2\,(x-\hat{x})\,(t) \;-\; L(t)D_{21}w(t) \\ (z-\hat{z})(t) & = & C_1(x-\hat{x})(t) & - D_{12}u(t) \end{array}\right.$$

For the moment, the influence of $D_{12}$ on the output is neglected. We can rewrite this equation as:

$$\left\{\begin{array}{lcll} (x-\hat{x})(t+1) & = & (A - L(t)C_2)(x-\hat{x})(t) & + \;(B_1 - L(t)D_{21})\,w(t) \\ (z-\hat{z})(t) & = & C_1(x-\hat{x})(t) & \end{array}\right.$$

From this equation, it can be observed that the minimisation of $\hat{z} - z$ is a pole placement problem. We define the adjoint of this system as:

$$\left\{\begin{array}{lcll} \tilde{x}(k) & = & \left(A^T - C_2^TL^T(t)\right)\tilde{x}(t+1) & + \;C_1^T\tilde{z}(t+1) \\ \tilde{w}(k) & = & \left(B_1^T - D_{21}^TL^T(t)\right)\tilde{x}(t+1) & \end{array}\right. \tag{4.5}$$

This is a slightly modified definition of the adjoint, since the input $\tilde{z}$ has the index $t+1$ instead of $t$. We can observe that the observer problem is equivalent to a state feedback problem of the adjoint system (4.5), if we define an input: $\tilde{y}(t) = -L^T(t)\tilde{x}(t)$. With this input, (4.5) becomes:

$$\left\{ \begin{array}{lcllll} \tilde{x}(k) & = & A^T\tilde{x}(t+1) & + & C_1^T\tilde{z}(t+1) & + C_2^T\tilde{y}(t+1) \\ \tilde{w}(k) & = & B_1^T\tilde{x}(t+1) & & & + D_{21}^T\tilde{y}(t+1) \end{array} \right. \tag{4.6}$$

This means that an optimal observer can be designed in the same way as state feedback, using the adjoint system.

### 4.1.3 Linear-Quadratic criteria

Considering the system of (4.1), we wish to use an input $u(k)$ such that the cost functional:

$$\mathcal{J} = \tfrac{1}{2} \sum_{t=0}^{N-1} \left[ x^T(t)Qx(t) + u^T(u)Ru(t) \right] \tag{4.7}$$

is minimised. This criteria is quadratic and by choosing $Q = Q^T \geq 0$, and $R = R^T > 0$, it is guaranteed that (4.7) has a global minimum.

The criterion of (4.7) is subject to a constraint, namely the system dynamics. In literature, this is called a constrained minima problem, which can be solved by adding the constraint to the cost functional, using a Lagrange multiplier $\lambda$ (see, e.g., [9]). If it is assumed that $B_1 = 0$, the cost functional becomes:

$$\mathcal{J}' = \sum_{t=0}^{N-1} \left[ \tfrac{1}{2}x^T(t)Qx(t) + \tfrac{1}{2}u^T(u)Ru(t) + \lambda^T(t+1)\left\{ Ax(t) + B_2u(t) - x(t+1) \right\} \right] \tag{4.8}$$

Although the index of the Lagrange multiplier could be defined at $t$ instead of $t+1$, the resulting solution would become less elegant than it is now [24]. Since (4.8) is a quadratic criterion, the minimum is achieved where its gradient equals zero.

Since this section only serves as an introduction to optimal control, the rest of the derivation is not given. It is however interesting to consider the partial derivative to $u(t)$:

$$\tfrac{\partial \mathcal{J}'}{\partial u(t)} = Ru(t) + B_2^T\lambda(t+1) = 0 \qquad \Leftrightarrow \qquad u(t) = -R^{-1}B_2^T\lambda(t+1)$$

From this equation, two things can be observed. First, if we assume $\lambda(t) = X(t)x(t)$, the controller output has the form of state feedback. That this assumption is in fact legitimate is shown in [12]. Second, for a strictly proper system, $u(N-1)$ affects $y(N)$, which lies outside the time interval $t \in [0, N-1]$. Therefore, we can set $u(N-1) = 0$, and hence, $\lambda(N) = 0$.

## 4.2 $\mathcal{H}_\infty$ optimal control

In the previous chapter, it was motivated that in order to obtain a controller such that the closed loop has RS and RP, the $\mathcal{H}_\infty$ norm of specific closed loop transfer functions should be smaller than 1. Using LFTs, the problem is converted to one in which the $\mathcal{H}_\infty$ the closed loop transfer between $w$ and $z$ should be smaller than 1. In the previous section, it was shown that designing an optimal controller boils down to the design of an optimal state feedback and a state observer, and that this can be done by using a quadratic criterion. In this section, we use elements from game theory to obtain quadratic criterions that can be used to obtain state feedback and a state observer which are optimal in the $\mathcal{H}_\infty$-norm.

### 4.2.1 Difference games and the mini-max problem

An $\mathcal{H}_\infty$ controller minimises the worst-case gain (see [7, 12, 30]). This problem can be seen as a game of two participants, namely, the controller which is trying to minimise the gain and the disturbance who is seeking the maximum gain. In a discrete time context, these kind of games are called difference games.

A difference game is usually described by game dynamics and an objective functional. The game dynamics in this case are the dynamics of the system (4.1), and the objective is the following mini-max criterion:

$$\min_u \max_w \mathcal{J}(x(t), w(t), u(t))$$

If we consider the definition of the $\mathcal{H}_\infty$ norm, and square it, we can obtain a suitable cost functional:

$$\mathcal{J} = \|P_{zw}\|_\infty^2 = \sup_{w \neq 0} \frac{\|z(t)\|_2^2}{\|w(t)\|_2^2} < \gamma^2$$

It is not possible to do any calculations with this criterion because a supremum denotes a lowest upper bound. We can however approach this bound to an arbitrary close distance $\epsilon$:

$$\frac{\|z(t)\|_2^2}{\|w(t)\|_2^2} = \gamma^2 - \epsilon^2$$

Since the supremum is never achieved, the optimal $\mathcal{H}_\infty$ controller does not exist and we must be satisfied with a suboptimal controller. Multiplying both sides by the denominator, and grouping the terms, yields:

$$\|z(t)\|_2^2 - \gamma^2 \|w(t)\|_2^2 = -\epsilon^2 \|w(t)\|_2^2$$

The left hand side of this equation can be used as a cost functional. This cost functional is quadratic and has a saddle point where its Jacobian equals zero. This is where the control is such that is minimises the maximum disturbance.

### 4.2.2 Full information control ($\mathcal{H}_\infty$ state feedback)

An optimal full information controller minimises the worst case gain between $w$ and $z$ assuming that all the states are available for construction of optimal feedback. The $\mathcal{H}_\infty$ optimal feedback is obtained by considering the following criterion:

$$\mathcal{J}' = \|z(t)\|_2^2 - \gamma^2 \|w(t)\|_2^2$$

for $t \in [0, N-1]$. The definition of the 2-norm for signals is given in appendix A. Similar to the previous section, the optimisation is constrained, and this constraint can be added to the cost functional using a Lagrange multiplier $\lambda$. Substituting (4.1) into the cost functional yields:

$$\begin{aligned}\mathcal{J}' = \sum_{t=0}^{N-1} & \tfrac{1}{2} x^T(t) C_1^T C_1 x(t) + \tfrac{1}{2} u^T(t) u(t) - \tfrac{1}{2}\gamma^2 w^T(t) w(t) \\ & \qquad + \lambda^T(t+1)\left\{Ax(t) + B_1 w(t) + B_2 u(t) - x(t+1)\right\}\end{aligned}$$

Note that the result is elegant, because of the conditions of (4.2). This functional has a saddle point where all partial derivatives are equal to zero:

$$\begin{aligned}\tfrac{\partial \mathcal{J}'}{\partial \lambda(t+1)} &= Ax(t) + B_1 w(t) + B_2 u(t) - x(t+1) = 0 && (4.9)\\ \tfrac{\partial \mathcal{J}'}{\partial x(t)} &= C_1^T C_1 x(t) + A^T \lambda(t+1) - \lambda(t) = 0 && (4.10)\\ \tfrac{\partial \mathcal{J}'}{\partial w(t)} &= -\gamma^2 w(t) + B_1^T \lambda(t+1) = 0 && (4.11)\\ \tfrac{\partial \mathcal{J}'}{\partial u(t)} &= u(t) + B_2^T \lambda(t+1) = 0 && (4.12)\end{aligned}$$

Substituting (4.11) and (4.12) into (4.9) and combining it with (4.10) yields:

$$\begin{bmatrix} x(t+1) \\ \lambda(t) \end{bmatrix} = \begin{bmatrix} A & -BR^{-1}B^T \\ C_1^T C_1 & A^T \end{bmatrix} \begin{bmatrix} x(t) \\ \lambda(t+1) \end{bmatrix} \tag{4.13}$$

with:

$$B = \begin{bmatrix} B_1 & B_2 \end{bmatrix}, \qquad \text{and:} \qquad R = \begin{bmatrix} -\gamma^2 I & 0 \\ 0 & I \end{bmatrix}$$

This system is the Hamilton-Jacobi-Bellman (HJB) system for the $\mathcal{H}_\infty$ full information control problem. This equation is similar to that of section 2.5.4, besides the fact that (2.27) was posed for an LQ optimal control problem, and had an external input. Thus, the difference between conventional optimal control and ILC is inherently posed as a tracking problem, instead of a regulator problem.

The HJB system of 4.13 is a two point boundary value problem, with the following boundary values:

$$x(0) = x_0 \qquad \text{and:} \qquad \lambda(N) = 0$$

This system can be converted to a single point boundary value problem by assuming that $\lambda(t) = X(t)x(t)$ (see [12, 24]). Note that $X(N) = 0$ because $\lambda(N) = 0$ and, in general,

$x(N) \neq 0$. Using this definition, the first equation in the set (4.13) becomes:

$$\begin{aligned}
x(t+1) &= Ax(t) - BR^{-1}B^TX(t+1)x(t+1) & \Leftrightarrow \\
x(t+1) &= \left(I + BR^{-1}B^TX(t+1)\right)^{-1}Ax(t) & \Leftrightarrow \\
x(t+1) &= Ax(t) - B\left(R + B^TX(t+1)B\right)^{-1}B^TX(t+1)Ax(t) &
\end{aligned}$$

Substitution of this into (4.10) yields:

$$X(t)x(t) = \Big[A^TX(t+1)A - A^TX(t+1)B\left(R + B^TX(t+1)B\right)^{-1}B^TX(t+1)A + C_1^TC_1\Big]x(t)$$

This equation must hold for every $x(t)$ with $t \in [0, N-1]$, hence:

$$X(t) = A^TX(t+1)A - A^TX(t+1)B\left(R + B^TX(t+1)B\right)^{-1}B^TX(t+1)A + C_1^TC_1 \quad (4.14)$$

This anti-causal difference equation is a time-varying Riccati equation and can be solved for a given $X(N)$. Using the solution of this Riccati equation, the $\mathcal{H}_\infty$ (sub)optimal state feedback is given by:

$$u(t) = -B_2^TX(t+1)\left(I + BR^{-1}B^TX(t+1)\right)^{-1}Ax(t) \quad (4.15)$$

which has indeed the form of state feedback as in (4.3), if we define:

$$K(t) = B_2^TX(t+1)\left(I + BR^{-1}B^TX(t+1)\right)^{-1}A$$

### 4.2.3 $\mathcal{H}_\infty$ optimal state observer

In section 4.1.2, we concluded that designing an observer is equivalent to designing state feedback for the adjoint system of (4.6). The $\infty$-norm of the adjoint system is given by:

$$\|P^T(z^{-1})\|_\infty = \sup_{\vartheta}\max_i \sigma_i\left(P^T(e^{-j\vartheta})\right) = \sup_{\vartheta}\max_i \sigma_i\left(P(e^{j\vartheta})\right) = \|P(z)\|_\infty$$

Hence, the $\infty$-norm is invariant under the adjoint operation. This means that we can design an $\mathcal{H}_\infty$ (sub)optimal state observer using the adjoint system. Similarly to the previous section, the cost functional is:

$$\tilde{\mathcal{J}}' = \|\tilde{w}(t)\|_2^2 - \gamma^2\|\tilde{z}(t)\|_2^2$$

Again, the constraint is added to the cost functional using a Lagrange multiplier $\tilde{\lambda}$ and using the assumptions of (4.2), the cost functional becomes:

$$\begin{aligned}
\tilde{\mathcal{J}}' = \sum_{t=0}^{N-1} &\tfrac{1}{2}\tilde{x}^T(t+1)B_1B_1^T\tilde{x}(t+1) + \tfrac{1}{2}\tilde{y}^T(t+1)\tilde{y}(t+1) - \tfrac{1}{2}\gamma^2\tilde{z}^T(t+1)\tilde{z}(t+1) \\
&+ \tilde{\lambda}^T(t+1)\left\{A^T\tilde{x}(t+1) + C_1^T\tilde{z}(t+1) + C_2^T\tilde{y}(t+1) - \tilde{x}(t)\right\}
\end{aligned}$$

Since all the derivations related to the (sub)optimal state observer are similar to the state feedback case, they will not be given. Similar to (4.13), the HJB system related to the $\mathcal{H}_\infty$ state observer is given by:

$$\left\{ \begin{array}{lcl} \begin{bmatrix} \tilde{x}(t) \\ \tilde{\lambda}(t+1) \end{bmatrix} & = & \begin{bmatrix} A^T & -C^T Q^{-1} C \\ B_1 B_1^T & A \end{bmatrix} \begin{bmatrix} \tilde{x}(t+1) \\ \tilde{\lambda}(t) \end{bmatrix} \\ \hat{y} & = & -C_2 \lambda(t) \end{array} \right. \tag{4.16}$$

with:

$$C = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \qquad \text{and:} \qquad Q = \begin{bmatrix} -\gamma^2 I & 0 \\ 0 & I \end{bmatrix}$$

This is again a two point boundary value problem. However, the boundary values of this equation are given by:

$$\tilde{x}(N) = x_n \qquad \text{and:} \qquad \tilde{\lambda}(0) = 0$$

This can be converted into a single point boundary value problem by defining $\tilde{\lambda}(t) = Y(t)\, \tilde{x}(t)$, with $Y(0) = 0$. Which results in a Riccati equation:

$$Y(t+1) = AY(t)A^T - AY(t)C^T \left(Q + CY(t)C^T\right)^{-1} CY(t)A^T + B_1 B_1^T \tag{4.17}$$

This Riccati equation is a forward recursion and can be solved using $Y(0) = 0$. The Riccati equation can be used to obtain a $\mathcal{H}_\infty$ optimal state observer. An expression of the observer is obtained by considering the state feedback of the adjoint system:

$$\tilde{y}(t) = -C_2 Y(t) \left[I - C^T \left(Q + CY(t)C^T\right)^{-1} CY(t)\right] A^T \tilde{x}(t)$$

According to section 4.1.2, this gives the following observer:

$$\left\{ \begin{array}{lcl} \hat{x}(t+1) & = & A\hat{x}(t) + B_2 u(t) + L(t) \left\{ y(t) - C_2 \hat{x}(t) \right\} \\ \hat{z}(t) & = & C_1 \hat{x}(t) \end{array} \right.$$

where:

$$L(t) = AY(t) \left(I + C^T Q^{-1} C Y(t)\right)^{-1} C_2^T \tag{4.18}$$

### 4.2.4 Measurement feedback

So far, it was assumed that there are two steps in the design of an $\mathcal{H}_\infty$ optimal controller: state feedback and a state observer. At the first glace, it appears appropriate to make an estimation of the states using an observer, and use these estimated states for feedback. In [35], it was shown that a Riccati equation can also be seen as a time varying similarity transformation. As a result, the states of the observer are different from the states of the state feedback. In order to use the results from the state feedback problem in the state observer, the state feedback controller needs to be expressed in the states of the observer. (Or vise versa, transforming the state observer states to the state feedback states.)

First, a HJB system for the measurement output is obtained by adding the third equation of (4.1) to the cost functional of section 4.2.2 using a second Lagrange multiplier $\mu$. The cost functional becomes:

$$\begin{aligned}\mathcal{J}' = \sum_{t=0}^{N-1} & \tfrac{1}{2}x^T(t)C_1^TC_1x(t) + \tfrac{1}{2}u^T(t)u(t) - \tfrac{1}{2}\gamma^2w^T(t)w(t) \\ & + \lambda^T(t+1)\{Ax(t) + B_1w(t) + B_2u(t) - x(t)\} + \mu^T(t)\{C_2x(t) + D_{21}w(t) - y(t)\}\end{aligned}$$

Obtaining a HJB system for this cost functional is slightly trickier. The cost functional has a saddle point where:

$$\tfrac{\partial \mathcal{J}'}{\partial x} = C_1^TC_1x(t) + A^T\lambda(t+1) + C_2^T\mu(t) - \lambda(t) = 0 \tag{4.19}$$

$$\tfrac{\partial \mathcal{J}'}{\partial \lambda} = Ax(t) + B_1w(t) + B_2u(t) - x(t+1) = 0 \tag{4.20}$$

$$\tfrac{\partial \mathcal{J}'}{\partial \mu} = C_2x(t) + D_{21}w(t) - y(t) = 0 \tag{4.21}$$

$$\tfrac{\partial \mathcal{J}'}{\partial u} = u(t) + B_2^T\lambda(t+1) = 0 \tag{4.22}$$

$$\tfrac{\partial \mathcal{J}'}{\partial w} = -\gamma^2w(t) + B_1^T\lambda(t+1) + D_{21}^T\mu(t) = 0 \tag{4.23}$$

Substituting (4.23) into (4.21), and regrouping yields:

$$\mu(t) = -\gamma^{-2}C_2x(t) + \gamma^2y(t) \tag{4.24}$$

Note that we also used (4.2). Substitution of (4.22) and (4.23) into (4.20) and substitution of (4.24) into (4.19) yields the following HJB system:

$$\begin{bmatrix} x(t+1) \\ \lambda(t) \end{bmatrix} \begin{bmatrix} A & -BR^{-1}B^T \\ -\gamma^2C^TQ^{-1}C & A^T \end{bmatrix} \begin{bmatrix} x(t) \\ \lambda(t+1) \end{bmatrix} + \begin{bmatrix} 0 \\ \gamma^2C_2^T \end{bmatrix} y(t) \tag{4.25}$$

where $B$, $C$, $Q$, and $R$ are defined as before. This HJB system could also be obtained by making an observer from (4.13) by adding an innovation term:

$$\begin{bmatrix} x(t+1) \\ \lambda(t) \end{bmatrix} \begin{bmatrix} A & -BR^{-1}B^T \\ C_1^TC_1 & A^T \end{bmatrix} \begin{bmatrix} x(t) \\ \lambda(t+1) \end{bmatrix} + \begin{bmatrix} 0 \\ \gamma^2C_2^T \end{bmatrix} (y(t) - C_2x)$$

Or, similarly, by adding a control input to the adjoint of (4.16):

$$\begin{bmatrix} x(t+1) \\ \lambda(t) \end{bmatrix} \begin{bmatrix} A & B_1B_1^T \\ -C^TQ^{-1}C & A^T \end{bmatrix} \begin{bmatrix} x(t) \\ \lambda(t+1) \end{bmatrix} + \begin{bmatrix} -\gamma^2B_2 \\ 0 \end{bmatrix} B_2^T\lambda(t+1)$$

The difference in $\gamma$ is a matter of scaling the equations. It is possible to make (4.25) lower triangular by applying the following similarity transformation:

$$\begin{bmatrix} x(t) \\ \lambda(t) \end{bmatrix} = \begin{bmatrix} I & \gamma^{-2}Y(t) \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{x}(t) \\ \lambda(t) \end{bmatrix} \tag{4.26}$$

where $Y(t)$ is the solution (4.17). In appendix B, it is shown that this similarity transformation results in a lower triangular system:

$$\begin{cases}\begin{bmatrix}\hat{x}(t+1)\\ \lambda(t)\end{bmatrix} = & \begin{bmatrix}A\left(I+Y(t)C^TQ^{-1}C\right)^{-1} & 0\\ -\gamma^2C^T\left(Q+CY(t)C^T\right)^{-1}C & \left(I+C^TQ^{-1}CY(t)\right)^{-1}A^T\end{bmatrix}\begin{bmatrix}\hat{x}(t)\\ \lambda(t+1)\end{bmatrix}\\ & +\begin{bmatrix}B_2\\ 0\end{bmatrix}K(t)x(t)+\begin{bmatrix}\gamma^2L(t)\\ \gamma^2\left(Y(t)A\right)^{-1}L(t)\end{bmatrix}y(t)\\ u(t)=-K(t)x(t)\end{cases} \tag{4.27}$$

This way, it is possible the use the optimal state feedback controller together with the observer. The only problem is that the controller depends on the untransformed states $x(t)$ and not on $\hat{x}(t)$. Fortunately, the relation between $x(t)$ and $\hat{x}(t)$ can be derived using (4.26) and the fact that $\lambda(t) = X(t)x(t)$ (see section 4.2.2):

$$\begin{aligned}x(t) &= \hat{x}(t) + \gamma^{-2}Y(t)X(t)x(t)\\ &= (I-\gamma^{-2}Y(t)X(t))^{-1}\hat{x}(t)\end{aligned} \tag{4.28}$$

Post-multiplying the equation of the original state feedback by this equation results in a state feedback controller that can be used in conjunction with the state observer. This transformation also adds an extra condition for existence of a suboptimal $\mathcal{H}_\infty$ controller, namely: the inverse in (4.28) must exist. A sufficient condition for existence of this inverse is that:

$$\rho(Y(t)X(t)) < \gamma^2$$

An important observation is that (4.27) shows the fundamental difference between feedback control and ILC: Since the state feedback of (4.15) depends only on $x(t)$, which is now transformed to $\hat{x}(t)$, the anti-causal part of (4.27) becomes unobservable. As a result, the final controller will be causal. In ILC, the optimal state feedback has an external input and, hence, the anti-causal part cannot be made unobservable. Therefore, a norm-optimal ILC controller consists of a causal and an anti-causal part.

### 4.2.5 Summary

In the previous section, we obtained the third condition for the existence of a suboptimal $\mathcal{H}_\infty$ controller for a given $\gamma$. To obtain a suboptimal controller that approaches the optimal controller, $\gamma$ should be iteratively lowered, and the following three conditions should be checked:

- The following Riccati equation has a solution $X(t) = X^T(t) \geq 0 \; \forall \; t \in [0, N-1]$:

$$X(t) = A^TX(t+1)A - A^TX(t+1)B\left(R+B^TX(t+1)B\right)^{-1}B^TX(t+1)A + C_1^TC_1$$

given:

$$X(N-1) = 0, \qquad \text{and:} \qquad B = \begin{bmatrix} B_1 & B_2 \end{bmatrix}, \qquad \text{and:} \qquad R = \begin{bmatrix} -\gamma^2 I & 0 \\ 0 & I \end{bmatrix}$$

- The following Riccati equation has a solution $Y(t) = Y^T(t) \geq 0 \;\forall\; t \in [0, N-1]$:

$$Y(t+1) = AY(t)A^T - AY(t)C^T \left(Q + CY(t)C^T\right)^{-1} CY(t)A^T + B_1 B_1^T$$

given:

$$Y(0) = 0, \qquad \text{and:} \qquad C = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \qquad \text{and:} \qquad Q = \begin{bmatrix} -\gamma^2 I & 0 \\ 0 & I \end{bmatrix}$$

- For all $t \in [0, N-1]$ the following coupling condition is satisfied:

$$\rho\left(X(t)Y(t)\right) < \gamma^2$$

For the lowest possible $\gamma$, the suboptimal $\mathcal{H}_\infty$ controller is given by:

$$\begin{cases} x(t+1) & = \left(A - \hat{K}(t) - L(t)C_2\right) x(t) \quad + \quad \gamma^2 L(t)y(t) \\ u(t) & = -B_2^T X(t+1)\hat{K}(t)\; x(t) \end{cases}$$

with:

$$\begin{aligned} \hat{K}(t) &= \left(I + BR^{-1}B^T X(t+1)\right)^{-1} A \left(I - \gamma^{-2} Y(t)X(t)\right)^{-1} \\ L(t) &= AY(t)\left(I + C^T Q^{-1} CY(t)\right)^{-1} C_2^T \end{aligned}$$

## 4.3 Closure

In this chapter, we derived an analytical solution of a finite-time $\mathcal{H}_\infty$ controller given a discrete time system. It was noted that conventional optimal control and norm optimal ILC differ in one fundamental way: optimal control is implicitly posed as a regulator problem and ILC is a tracking problem. This difference causes feedback controllers to be causal and ILC controllers to be noncausal.

# Chapter 5 Robust ILC

In the previous chapters, we obtained basic knowledge of ILC, and the basics of robust feedback control: modelling uncertainty and controller synthesis. This provides sufficient knowledge to discuss robust ILC, which is the topic of this chapter. To be more specific: this chapter focusses on ILC methods which are robust against uncertainty in the design model. First, robustness of the linear-quadratic optimal ILC solutions of section 2.5.3 is discussed. Then, we elaborate on a method that describes uncertainty as an interval matrix. The last part of this chapter discusses methods related to $\mathcal{H}_\infty$ optimisation.

## 5.1 Robustness of the norm-optimal ILC solution

Norm optimal ILC solutions do not incorporate model uncertainty explicitly. Nevertheless, it is possible to analyse robust convergence. In [26], robust convergence is analysed for a lifted domain norm optimal learning controller with learning gain:

$$L = (J^T J + \beta I)^{-1} J^T \tag{5.1}$$

and a SISO system with an additive uncertainty:

$$J_p(z) = J(z) + W(z)\Delta(z), \qquad \text{with:} \qquad \|\Delta(z)\|_\infty \leq 1 \tag{5.2}$$

According to proposition 2, an ILC controlled system with (5.1) and (5.2) is monotonically convergent if:

$$\begin{aligned} &\|Q(z)\left(I - \left(J^T(z^{-1})J(z) + \beta\right)^{-1} J^T(z^{-1})\left(J(z) + W(z)\Delta(z)\right)\right)\|_\infty < 1 \quad \Leftrightarrow \\ &\|Q(z)\left(J^T(z^{-1})J(z) + \beta I\right)^{-1}\left(\beta - J^T(z^{-1})W(z)\Delta(z)\right)\|_\infty < 1 \end{aligned} \tag{5.3}$$

where the second step is obtained by adding the term $(J^T(z^{-1})J(z) + \beta)^{-1}(\beta - \beta)$ to the equation. We can show that this is guaranteed to be monotonically convergent if we assume that the uncertainty is bounded as follows:

$$|Q(e^{j\vartheta})J^T(e^{-j\vartheta})W(e^{j\vartheta})\Delta(e^{j\vartheta})| \;<\; |J^T(e^{-j\vartheta})J(e^{j\vartheta})| \tag{5.4}$$

To show that this statement is valid, we make use of the fact that the following inequality holds:

$$|J^T(e^{-j\vartheta})J(e^{j\vartheta})| \;<\; \beta(1-|Q(e^{j\vartheta})|)+|J^T(e^{-j\vartheta})J(e^{j\vartheta})| \tag{5.5}$$

if $Q(e^{j\vartheta})$ is zero phase filter with $|Q(e^{j\vartheta})|<1$, and $\beta>0$. Combining (5.4) and (5.5) yields:

$$\begin{aligned}
|Q(e^{j\vartheta})J^T(e^{-j\vartheta})W(e^{j\vartheta})\Delta(e^{j\vartheta})| \;&<\; \beta(I-|Q(e^{j\vartheta}|)+|J^T(e^{-j\vartheta})J(e^{j\vartheta})| && \Rightarrow\\
\beta|Q(e^{j\vartheta})|+|Q(e^{j\vartheta})J^T(e^{-j\vartheta})W(e^{j\vartheta})\Delta(e^{j\vartheta})| \;&<\; |\beta I+J^T(e^{-j\vartheta})J(e^{j\vartheta})| && \Rightarrow\\
\left|Q(e^{j\vartheta})\left(\beta I-J^T(e^{-j\vartheta})W(e^{j\vartheta})\Delta(e^{j\vartheta})\right)\right| \;&<\; |\beta I+J^T(e^{-j\vartheta})J(e^{j\vartheta})|
\end{aligned}$$

which is equivalent to (5.3). The second step in this sequence is allowed because $J^T(e^{j\vartheta})J(e^{j\vartheta})>0$. This shows sufficiently that if (5.4) is satisfied, the perturbed system converges monotonically. In [10], a similar expression is derived for systems with a multiplicative uncertainty. Both methods share the same disadvantage: since the analysis is done in the frequency domain, the result can be conservative. Moreover, according to proposition 4, introducing a $Q(e^{j\vartheta})<1$ for some $\vartheta\in[-\pi,\pi]$ means that performance is sacrificed.

## 5.2 Robust ILC controller synthesis using interval matrices

In the lifted domain, model uncertainty can also be represented directly in the system matrix $J$ [1, 2]. Consider the model set $J^I$ in which each Markov parameter has an interval uncertainty, then the interval matrix is given by:

$$\mathcal{J}^I=\{J^I: j^I_{ij}\in[\underline{j_{ij}},\overline{j_{ij}}], i,j=1,\ldots,n\}$$

where $\underline{j_{ij}}$ and $\overline{j_{ij}}$ denote the minimum and maximum extreme value of the $ij$-th element of the matrix $J^I$, respectively. The lower bound matrix $\underline{J}$ is the matrix whose element are $\underline{j_{ij}}$ and the upper bound matrix $\overline{J}$ is the matrix whose elements are $\overline{j_{ij}}$. The nominal system matrix is defined as follows:

$$J\equiv\frac{\overline{J}+\underline{J}}{2}$$

In [1], a method is proposed that yields a controller that has a maximal interval in which systems are asymptotically stable or monotonic convergent. In this section, we only elaborate on asymptotic stability. First, we define the following notation:

$$T\equiv I-LJ \qquad \text{and:} \qquad T^I\equiv I-LJ^I$$

And:

$$\Delta T\equiv T-T^I=I-LJ-I+LJ^I=L(J^I-J)=L\Delta J \tag{5.6}$$

where $\Delta T$ is the interval uncertainty of the ILC controlled system, and $\Delta J$ is the interval uncertainty of the nominal system matrix. The goal of the controller synthesis is to maximise

$\|\Delta J\|_i$ using $\Delta T$. Finally, we introduce the notion of 'bigger norm value between a matrix and its transpose', denoted $\langle\bullet\rangle$:

$$\langle\Delta T\rangle_i \equiv \max\{\|\Delta T^T\|_i, \|\Delta T\|_i\} \tag{5.7}$$

where $\|\bullet\|_i$ denotes a matrix norm.

### 5.2.1 Robust stability analysis

Given a designed $L$ for the nominal system $J$, proposition 1 of chapter 2 gives a necessary condition for stability. Fulfilling this condition also means that the discrete-time Lyapunov equation:

$$(I - LJ)^T P(I - LJ) - P = -I \tag{5.8}$$

has a symmetric positive, definite solution for $P$ (see any resource on Lyapunov stability, e.g., [29, 43]). Given this constraint, the goal is to maximise the allowable interval uncertainty (AIU), $\Delta J$, for which $I - L(J \pm \Delta J)$ is guaranteed to be asymptotically stable. Hence, an expression for $\Delta J$ is needed that can be used as a cost function for the optimisation.

Given a symmetric, positive definite solution for $P$, ILC controlled system with the perturbed system $J^I$ is still asymptotically stable if:

$$(T^I)^T P T^I - P < 0$$

Using $T^I = T - \Delta T$ and substitution of (5.8), we obtain:

$$-T^T P\Delta T - (\Delta T)^T PT + (\Delta T)^T P\Delta T < I$$

Taking the norm of this expression and using the multiplicative property of matrix norms yields:

$$\|T^T\| \; \|P\| \; \|\Delta T\| + \|(\Delta T)^T\| \; \|P\| \; \|T\| + \|(\Delta T)^T\| \; \|P\| \; \|\Delta T\| < 1$$

which is a sufficient condition. Using the definition of (5.7), and $\Delta T = L\ \Delta J$, it is possible to rewrite this as:

$$\begin{aligned} 2\langle T\rangle\|P\|\langle\Delta T\rangle + \langle\Delta T\rangle\|P\|\langle\Delta T\rangle &< 1 \qquad \Leftrightarrow \\ \left(2\langle T\rangle\langle\Delta T\rangle + \langle\Delta T\rangle^2\right)\|P\| &< 1 \qquad \Rightarrow \\ \left(2\langle T\rangle\langle L\rangle\langle\Delta J\rangle + \langle L\rangle^2\langle\Delta J\rangle^2\right)\|P\| &< 1 \end{aligned}$$

Since this is a quadratic equation, the inequality would in general be satisfied for a range of $\langle\Delta J\rangle$. Given the fact that $\langle\Delta J\rangle > 0$, the inequality is satisfied for:

$$\langle\Delta J\rangle < \frac{-\langle L\rangle\langle T\rangle + \sqrt{[\langle L\rangle\langle T\rangle]^2 + \frac{\langle L\rangle^2}{\|P\|}}}{\langle L\rangle^2} = \frac{-\langle T\rangle + \sqrt{\langle T\rangle^2 + \frac{1}{\|P\|}}}{\langle L\rangle}$$

The final step in obtaining an expression for the AIU, is substituting $\langle T\rangle = \langle I - LJ\rangle$. Then, a sufficient condition for the upper bound of the AIU is given by:

$$\langle \Delta J\rangle_{\max} \equiv \frac{-\langle I - LJ\rangle + \sqrt{\langle I - LJ\rangle^2 + \frac{1}{\|P\|}}}{\langle L\rangle} \tag{5.9}$$

Using this equation it is possible to determine the AIU for asymptotic stability, given a learning gain $L$. An expression that gives an upper bound for the AIU that ensures monotonic convergence is more cumbersome and is shown in [1]. It will not be discussed in this report.

### 5.2.2 Synthesising a robustly stabilising controller

Using (5.9), it is possible to synthesise a controller. It is the solution of the following nonlinear constrained optimisation problem:

$$\max_{L} \Delta J, \quad \text{such that:} \quad (I - LJ)^T P(I - LJ) - P = -I \tag{5.10}$$

The result of this optimisation problem will yield $L = 0$, because this results in an infinite AIU. Therefore, the required spectral radius should be fixed. This can be done by adding the following constraint to the optimisation problem:

$$\rho(I - LJ) < \rho_{\max}$$

where $\rho_{\max} < 1$. This optimisation problem can be solved with the MATLAB routine `fmincon`.

The proposed algorithm has an important disadvantage. For longer trajectories (e.g., 1 second at 1kHz, which gives 1000 samples), the interval matrices become large. Moreover, a nonlinear optimisation problem is numerically extensive. Therefore, solving this problem for realistic situations is difficult.

## 5.3 $\mathcal{H}_\infty$-like methods

In this section, methods that use $\mathcal{H}_\infty$ optimisation are discussed. As seen in the preceding chapters, $\mathcal{H}_\infty$ techniques arise naturally when dealing with robustness. Basically, two different approaches can be distinguished, namely in the $z$-domain and the lifted domain. The $z$-domain techniques can be further divided into algorithms that design an ILC controller in cooperation with feedback and algorithms that design ILC independent of feedback.

### 5.3.1 Robust current cycle ILC

Methods that discuss ILC controller design with $\mathcal{H}_\infty$-feedback control design are discussed in, e.g., [4, 46]. Both references discuss ILC in the $z$-domain, and therefore assume an infinite trial length when designing an ILC controller. In [4], the following ILC control law is chosen:

$$f_{k+1} = f_k + K_0 e_k + K_1 e_{k+1} \tag{5.11}$$

Thus, this control law incorporates feedback control design in ILC control design. In figure 5.1, a block diagram of this control law is shown. In this block scheme, $G(z)$ denotes the system to be controlled. (Do not confuse with the definition of $J$ in chapter 2.)

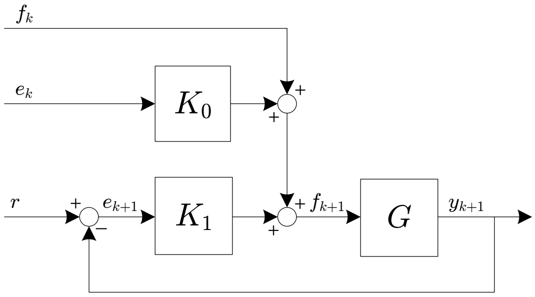


Figure 5.1: Block diagram of ILC controlled system

Designing an optimal ILC controller corresponds to design $K_0(z)$ and $K_1(z)$ in (5.1). In order to make a proper choice, we consider the error propagation from trial to trial. The error at trial $k+1$ is given as follows:

$$e_{k+1} = (I + GK_1)^{-1} r - (I + GK_1)^{-1} GK_0 e_k - (I + GK_1)^{-1} G f_k$$

If we evaluate this equation at $k+2$ and at $k+1$ and subtract these, we obtain:

$$e_{k+2} - e_{k+1} = (I + GK_1)^{-1} GK_0 (e_k - e_{k+1}) + (I + GK_1)^{-1} G (f_k - f_{k+1})$$

Substitution of (5.11) yields:

$$\begin{aligned} e_{k+2} - e_{k+1} &= -(I + GK_1)^{-1} GK_0 e_{k+1} - (I + GK_1)^{-1} GK_1 e_{k+1} \qquad \Leftrightarrow \\ e_{k+2} &= (I + GK_1)^{-1} (I - GK_0) e_{k+1} \end{aligned}$$

Which is equivalent to:

$$e_{k+1} = (I + GK_1)^{-1} (I - GK_0) e_k \tag{5.12}$$

From this equation, it stems that the optimal ILC controller is $K_0 \approx G^{-1}$, hence: the (approximate) inverse of the system. Moreover, we can observer from (5.12) that designing a high gain $K_1(z)$ also gives good performance. In [4], standard $\mathcal{H}_\infty$ optimisation routines are used to obtain an optimal $K_0(z)$ and $K_1(z)$. This is done in two steps. The first step is to find an approximate inverse of $G(z)$ by solving:

$$\min_{K_0} \| I - GW_1 K_0 \|_\infty$$

where $W_1(z)$ is chosen to be non proper, such that the product $GW_1$ is proper but not strictly proper. Typically, $W_1(z)$ has low amplification at low frequencies and high amplification at high frequencies. (How low and how high depends on the frequency content of the reference trajectory.) The second is step is a usual mixed sensitivity problem:

$$\min_{K_1} \left\| \begin{matrix} K_1(I+GK_1)^{-1}W_2 \\ (I+GK_1)^{-1}W_3 \end{matrix} \right\|_\infty$$

where $W_2(z)$ can be an upper bound with respect to additive uncertainty, and $W_3(z)$ can be chosen the inverse of the desired sensitivity function, which is common in $\mathcal{H}_\infty$ feedback controller design.

### 5.3.2 Robust ILC using the generalised plant

An alternative approach of $\mathcal{H}_\infty$ ILC controller design in the $z$-domain is given by [15, 16]. In these references, the generalised plant paradigm is used to design an optimal ILC controller. Moreover, it is assumed that the feedback controller is already designed.

Recall that a sufficient condition for monotonic convergence in the $z$-domain is:

$$\|Q(I-LJ_p)\|_\infty$$

where:

$$J_p(z) = J(z) + W(z)\Delta(z), \qquad \text{with:} \qquad \|\Delta(z)\|_\infty < 1$$

This can be put in the generalised plant paradigm, see figure 5.2. The goal is to minimise the $\infty$-norm between $f_k$ and $f_{k+1}$ by designing an $L$. Note that this requires the $Q(z)$ filter to be given. Therefore, an optimal controller is synthesised in an iterative way: first, we choose $Q(z)$ to be a Butterworth-filter with a specific cut-off frequency. Then, we synthesise a learning gain $L(z)$, given $J(z)$, $W(z)$ and $Q(z)$. If the resulting closed loop $\|Q(I-LJ_p)\|_\infty < 1$, the cut-off frequency of the $Q(z)$ filter can be increased, if the $\infty$-norm is larger than 1, the cut-off frequency should be decreased. Iteratively, the best trade-off between performance, convergence rate and robustness can be obtained.

Like the other robust ILC methods discussed in this chapter, the ILC syntheses proposed [4, 15, 16, 46] have a drawback. Since they make use of conventional $\mathcal{H}_\infty$ optimisation methods in the $z$-domain, they result in a proper or strictly-proper learning filter. This is equivalent to a causal learning filter in the time-domain. According to [27, 50], there exists a feedback equivalent for causal learning filters. Hence, in a noise free environment, it is not possible to outperform a feedback controller with ILC. For that, it is necessary that the learning filter is noncausal.

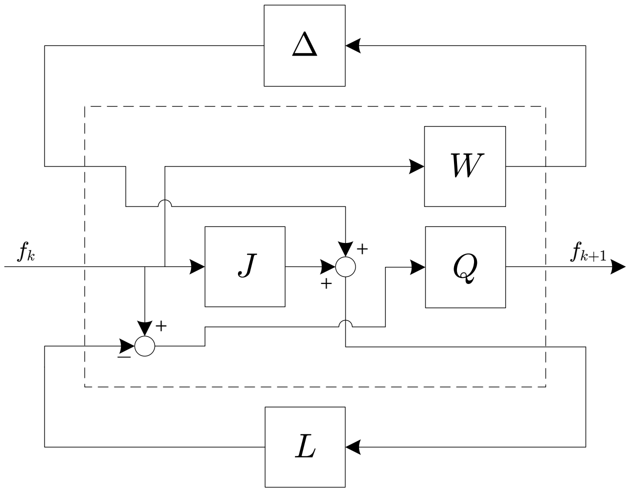


Figure 5.2: ILC control problem posed using the generalised plant

### 5.3.3 Lifted domain techniques

A different approach for synthesising ILC controllers using $\mathcal{H}_\infty$ techniques is discussed in [34]. This particular paper proposes a higher order ILC controller in the lifted domain that can be used to handle trial varying disturbances and model uncertainty. In this section, we will roughly sketch the outline of this method.

If we revisit figure 2.2 of chapter 2, and assume $Q = I$, this figure can be thought to be a large MIMO system, where we wish to apply a feedback controller (in the lifted domain). An alternative representation of 2.2 is given by figure 5.3. In this setting, the IMP-part of the ILC controller is taken as part of the system. Since all states of the ILC system are known, for they are in fact a part of the controller, we can apply state feedback. Moreover, the real system behaviour is modelled as parametric uncertainty (see section 3.3), by taking $J_p = J + \delta J$.

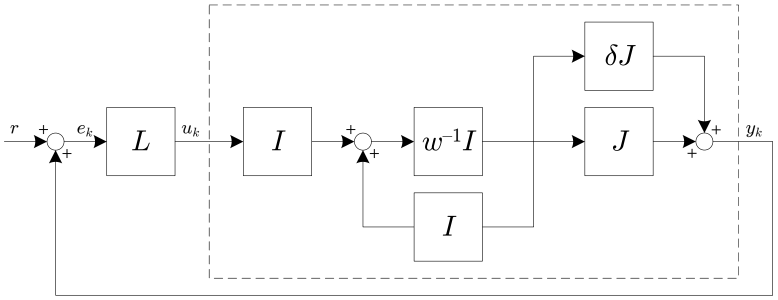


Figure 5.3: $\mathcal{H}_\infty$ lifted ILC control problem with model uncertainty

The difference equation describing this system then becomes:

$$\begin{cases} f_{k+1} &= If_k + Iu_k \\ y_k &= (J + \delta J)f_k \end{cases}$$

This difference equation can be rewritten as an upper LFT, using the Matlab routine `lftdata`. From this, and suitable performance specifications, a generalised plant can be obtained. Then, using the machinery from section 4.2.2, we can calculate the $\mathcal{H}_\infty$ optimal state feedback. These steps will not be further described in this report.

Similar to section 5.2, this method suffers from the fact that is used incredibly large uncertainty descriptions. Moreover, solving a Riccati equation for these large systems is numerically very difficult.

## 5.4 Closure

This chapter provided an overview of state-of-the-art robust ILC algorithms. Although they all have proven to work well in specific applications, they all suffer from one or more drawbacks:

- conservatism due to the usage of an infinite time analysis ($z$-domain);
- numerical difficulties because of the large matrices involved;
- causality, which does not give the best achievable performance;
- conservatism/complexity of the uncertainty model.

Future research is required to overcome these drawbacks.

Chapter 6

# Conclusions and Recommendations

The report has presented an overview of the current state of art of ILC, and how to deal with model uncertainties. Moreover, we have investigated how to represent model uncertainty and robust feedback control. In this final chapter, conclusions are drawn from the work discussed in this report and some recommendations for future research are made.

## 6.1 Conclusions

Although this report contains a summery of existing results, there are still some conclusions to be drawn.

The preferred framework for studying ILC is the lifted framework. However, we also noticed that a norm-optimal solution in the lifted domain has an equivalent solution in the finite-time domain.

Model uncertainties can be represented in a convenient way in the frequency domain. This can be used in the time domain using a state space realisation of the weighting filters. Parametric uncertainty can be directly used in the time domain. Using LFTs, a generalised plant framework can be obtained, which is a very useful way of representing control problems.

A thorough treatment of the finite-time, discrete-time $\mathcal{H}_\infty$ control problem was given. It was observed in chapter 2, that there exists an intersection between ILC and feedback control. It was noted that they are based on similar HJB-systems. However, they differ in a fundamental way. This difference is caused by the fact that optimal control is implicitly posed as a regulator problem and that ILC is a tracking problem. This causes feedback controllers to be causal and ILC controllers to be noncausal.

The state of art robust ILC methods still suffer from some deficiencies. In the lifted domain, there exist some methods for synthesising a robust ILC controller, but they are numerically demanding, due to the large scale of the problem. There are also some robust ILC methods in the frequency domain, but they all results in causal learning filter and lead to conservative results.

## 6.2 Recommendations for future research

This literature survey has shown some potential directions for future research. Robust ILC methods of chapter 5 that use results from robust feedback control have the advantage of representing model uncertainty in a convenient way. However, they make use of the $z$-domain for synthesising controllers, which assumes an infinite trial length. Furthermore, the results are causal. The low order ILC proposed in chapter 2 is related to the LQ optimal control problem in the time domain. However, the results are noncausal and explicitly deal with a finite trial length. Future research should focus on unifying robust control and low order ILC in such a way that the results are robustly stable, noncausal and act on a finite time interval.

This research can be further extended by introducing $\mu$-synthesis, which solves the RP problem for ILC. Furthermore, since we obtained a low-order ILC solution by considering the LQ optimal solution in the lifted domain, the converse should also be possible: Given a finite-time robust ILC solution, obtaining a lifted domain equivalent solution. Additionally, we can investigate the simultaneous design of feedback control and ILC control. It should be possible to pose this problem using the generalised plant, such that the resulting controller has two inputs and two outputs: one input and output for feedback control, and one input and output for ILC control. Of course, all results should be implemented on an experimental set-up.

# Appendix A Mathematical definitions

## A.1 The singular value decomposition [43, 45]

Any matrix $A \in \mathbb{C}^{n \times m}$ can be factorised using a singular values decomposition (SVD):

$$A = U \Sigma V^H$$

where $U \in \mathbb{C}^{n \times n}$ and $V \in \mathbb{C}^{m \times m}$ are unitary matrices ($UU^H = VV^H = I$) and $\Sigma \in \mathbb{R}^{n \times m}$ contains a matrix $\hat{\Sigma}$ as in:

$$\Sigma = \begin{bmatrix} \hat{\Sigma} \\ 0 \end{bmatrix}, \quad \text{for: } n \geq m \qquad \text{and:} \qquad \Sigma = \begin{bmatrix} \hat{\Sigma} & 0 \end{bmatrix}, \quad \text{for: } n \leq m$$

Matrix $\hat{\Sigma}$ is a diagonal matrix of real, nonnegative singular values $\sigma_i$, arranged in descending order:

$$\overline{\sigma} \equiv \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_k \equiv \underline{\sigma} \geq 0, \qquad k = \min\{n, m\}$$

For a matrix with $\text{rank}(A) = p < m \leq n$, matrixes $U$, $\Sigma$, and $V$ can be partitioned, such that:

$$A = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^H \\ V_2^H \end{bmatrix}$$

where:

$$\begin{array}{lll} U_1 \in \mathbb{C}^{n \times p}, & V_1 \in \mathbb{C}^{m \times p}, & \Sigma_1 \in \mathbb{R}^{p \times p}, \\ U_2 \in \mathbb{C}^{n \times (n-p)}, & V_2 \in \mathbb{C}^{m \times (m-p)}, & \Sigma_2 \approx 0 \in \mathbb{R}^{(n-p) \times m-p}, \end{array}$$

From this, it follows that:

$$A \approx U_1 \Sigma_1 V_1^H$$

## A.2 Norms for vectors, signals, matrices and systems [43, 53]

A norm is a real valued scalar function: $\| \bullet \| : \mathbb{C}^{n\times m} \rightarrow \mathcal{R}$, which gives an overall measure of 'magnitude' or size of a mathematical object, such as a vector, matrix, signal or system. It satisfies the following properties:

$$\begin{array}{lrcll} (a) & \|x\| & < & 0 & \forall\ x \neq 0 \\ (b) & \|x\| & = & 0 & \text{iff: } x = 0 \\ (c) & \|ax\| & = & |a|\|x\| & \forall\ a \in \mathbb{C} \\ (d) & \|x+y\| & \leq & \|x\| + \|y\| & \end{array}$$

where $x$ and $y$ are elements of a linear space $V \subset \mathbb{C}$. For matrices $A, B \in \mathbb{C}^{n\times m}$, also the following property holds:

$$(e) \qquad \|AB\| \ \leq \ \|A\|\|B\|$$

### A.2.1 Norms for vectors

Consider a vector $x \in \mathbb{C}^n$. The $p$-norm of this vector is defined by:

$$\|x\|_p = \left( \sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}} \qquad \text{for:} \quad 1 < p \leq \infty$$

The most frequently used norms are the 1-, 2- and the $\infty$-norm, which are given by:

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|, \qquad \|x\|_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}, \qquad \|x\|_\infty = \max_i |x_i|$$

where the $\infty$-norm can be seen as the limit case where $p \rightarrow \infty$. The 2-norm is often called Euclidean norm, because it corresponds to the notion of distance in to the Euclidean space and relates to the Euclidean inner product:

$$\langle x, x \rangle = x^H x = \|x\|_2^2$$

### A.2.2 Norms for matrices

Let matrix $A = [a_{ij}] \in \mathbb{C}^{n\times m}$, the $p$-norm of a matrix is defined by:

$$\|A\|_p = \sup_{x\neq 0} \frac{\|Ax\|_p}{\|x\|_p} \qquad \text{for:} \quad 1 < p \leq \infty$$

Such a matrix norm is said to be induced by two vector norms. Note that this intuitively corresponds to the fact that a matrix maps a vector into another vector. The most frequently used matrix norm happen to be the most easy to compute.

The 1-norm is simply the the maximum absolute column sum of the matrix:

$$\|A\|_1 = \sup_{x\neq 0} \frac{\|Ax\|_1}{\|x\|_1} = \sup_{x\neq 0} \frac{\sum_{i=1}^{n} |(Ax)_i|}{\sum_{j=1}^{m} |x_j|} = \sup_{x\neq 0} \frac{\sum_{i=1}^{n} |\sum_{j=1}^{m} a_{ij}x_j|}{\sum_{j=1}^{m} |x_j|} = \max_j \sum_{i=1}^{n} |a_{ij}|$$

Equivalently, the $\infty$-norm is the maximum absolute row sum of the matrix:

$$\|A\|_\infty = \sup_{x\neq 0} \frac{\|Ax\|_\infty}{\|x\|_\infty} = \sup_{x\neq 0} \frac{\max_i |\sum_{j=1}^{m} a_{ij}x_j|}{\max_j |x_j|} = \max_i \sum_{j=1}^{m} |a_{ij}|$$

The 2-norm can be calculated using the Euclidean inner product and the singular value decomposition, and $y = V^H x$:

$$\|A\|_2^2 = \sup_{x\neq 0} \frac{\|Ax\|_2^2}{\|x\|_2^2} = \sup_{x\neq 0} \frac{x^H V \Sigma^2 V^H x}{x^H x} = \sup_{x\neq 0} \frac{y^H \Sigma^2 y}{y^H y} = \max_{y\neq 0} \frac{\sum_{j=1}^{m} \sigma_j^2 y_j^2}{\sum_{j=1}^{m} y_j^2} = \max_i \sigma_i^2(A)$$

Furthermore, there exists norms that are not induced, but which are direct extensions of the definitions of the vector norms, namely: the sum-norm, the Frobenius-norm, and the max element norm. These are extensions of the 1-, 2-, and $\infty$-norm of vectors. These matrix norms are given by:

$$\|A\|_{\text{sum}} = \sum_{i,j} |a_{ij}|, \qquad \|A\|_{\text{fro}} = \sqrt{\sum_{i,j} |a_{ij}|} = \sqrt{\text{tr}(A^H A)}, \qquad \|A\|_{\max} = \max_{i,j} |a_{ij}|$$

where $tr(\bullet)$ denotes the trace operator, which is the sum of all diagonal terms.

### A.2.3 Norms for signals

A signal is a real-valued, vector valued function of time. Given a discrete time signal $x(t)$ defined on time set $t \in T \subset \mathbb{Z}$, we can obtain the $l_p$-norm of a signal by summing the signal over the time-axis, and summing the elements of the resulting vector. The $l_p$-norm of a time signal is defined as follows:

$$\|x(t)\|_p = \left( \sum_{t\in T} \sum_i |x_i(t)|^p \right)^{\frac{1}{p}}$$

The most frequently used norms are:

$$\|x(t)\|_1 = \sum_{t\in T} \sum_i |x_i(t)|, \qquad \|x(t)\|_2 = \sqrt{\sum_{t\in T} \sum_i |x_i(t)|^2}, \qquad \|x(t)\|_\infty = \sup_{t\in T} \max_i |x_i(t)|$$

### A.2.4 Norms for systems

In control theory, system norm are often used to assess performance. They are obtained by generalisation of matrix norms and are defined for stable systems. The most frequently used system norms the $\mathcal{H}_2$- and $\mathcal{H}_\infty$-norm. The $\mathcal{H}_\infty$-norm is given by:

$$\|G(z)\|_\infty = \sup_{u\neq 0} \frac{\|Gu\|_2}{\|u\|_2} = \sup_{u\neq 0} \max_i \sigma_i \left( G(e^{j\vartheta}) \right)$$

The $\mathcal{H}_2$-norm is obtained by taking the Frobenius-norm and integrate over all frequencies:

$$\|G(z)\|_2 = \sqrt{\frac{1}{2\pi} \int_{-\pi}^{\pi} \mathrm{tr}\left(G^H(e^{j\vartheta})G(e^{j\vartheta})\right) d\vartheta}$$

By using Parseval's theorem, the $\mathcal{H}_2$-norm of a transfer function matrix is equal to the 2-norm of its impulse response. Note that this norm is not induced and therefore, it does not fulfil the multiplicative property (e).

Finally, a less frequent used norm is the $\mathcal{H}_1$-norm, given by:

$$\|G(z)\|_1 = \sup_{u\neq 0} \frac{\|Gu\|_\infty}{\|u\|_\infty} = \max_i \sum_{j=1}^{m} \sum_{t=0}^{\infty} |g_{ij}(t)|$$

In [14], it was shown that if a system is stable in a $\mathcal{H}_\infty$-norm, it is also stable in a $\mathcal{H}_1$-norm.

# Appendix B Extended derivation of equation (4.27)

In section 4.2.4, the following HJB equation is discussed:

$$\begin{bmatrix} x(t+1) \\ \lambda(t) \end{bmatrix} \begin{bmatrix} A & -BR^{-1}B^T \\ -\gamma^2 C^T Q^{-1} C & A^T \end{bmatrix} \begin{bmatrix} x(t) \\ \lambda(t+1) \end{bmatrix} + \begin{bmatrix} 0 \\ \gamma^2 C_2^T \end{bmatrix} y(t)$$

Which is transformed using the following time varying similarity transformation:

$$\begin{bmatrix} x(t) \\ \lambda(t) \end{bmatrix} = \begin{bmatrix} I & \gamma^{-2} Y(t) \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{x}(t) \\ \lambda(t) \end{bmatrix}$$

The transformed system becomes:

$$\begin{cases} \hat{x}(t+1) & = & A\hat{x}(t) + \gamma^{-2} A Y(t) \lambda(t) - BR^{-1}B^T \lambda(t+1) - \gamma^{-2} Y(t+1) \lambda(t+1) \\ \lambda(t) & = & -C^T Q^{-1} C \left( \gamma^2 \hat{x}(t) + Y(t) \lambda(t) \right) + A^T \lambda(t+1) + \gamma^2 C_2^T y(t) \end{cases}$$

Rewriting the second equation of this set gives:

$$\lambda(t) = \left( I + C^T Q^{-1} C Y(t) \right)^{-1} \left( -\gamma^2 C^T Q^{-1} C \hat{x}(t) + A^T \lambda(t+1) + \gamma^2 C_2^T y(t) \right)$$

Substituting yields:

$$\begin{cases} \hat{x}(t+1) & = & \Big( A - AY(t) \left( I + C^T Q^{-1} C Y(t) \right)^{-1} C^T Q^{-1} C \Big) \hat{x}(t) \\ & & +\gamma^{-2} \Big( AY(t) \left( I + C^T Q^{-1} C Y(t) \right)^{-1} A - \gamma^2 BR^{-1}B^T - Y(t+1) \Big) \lambda(t+1) \\ & & \quad +AY(t) \left( I + C^T Q^{-1} C Y(t) \right)^{-1} \gamma^2 C_2^T y(t) \\ & & \\ \lambda(t) & = & \left( I + C^T Q^{-1} C Y(t) \right)^{-1} \left( -\gamma^2 C^T Q^{-1} C \hat{x}(t) + A^T \lambda(t+1) + \gamma^2 C_2^T y(t) \right) \end{cases} \tag{B.1}$$

The part depending on $\hat{x}(t)$ of the first equation in (B.1) can be written more conveniently:

$$\Big( A - AY(t) \left( I + C^T Q^{-1} C Y(t) \right)^{-1} C^T Q^{-1} C \Big) \hat{x}(t) = A(I + Y(t) C^T Q^{-1} C)^{-1} \hat{x}(t) \tag{B.2}$$

Moreover, consider the part depending on $\lambda(t+1)$ in the first equation of (B.1):

$$\gamma^{-2}\left(AY(t)\left(I+C^TQ^{-1}CY(t)\right)^{-1}A^T-Y(t+1)+B_1B_1^T-\gamma^2B_2B_2^T\right)\lambda(t+1)=$$
$$\gamma^{-2}\left(AY(t)A^T-AY(t)C^T\left(Q+CY(t)C^T\right)^{-1}CY(t)A^T-Y(t+1)+B_1B_1^T-\gamma B_2B_2^T\right)\lambda(t+1)$$

Because $Y(t)$ is a solution to the Riccati equation of (4.17), this expression simplifies to:

$$B_2B_2^T\lambda(t+1) \tag{B.3}$$

Substituting (B.2) and (B.3) into (B.1), assuming $A$ to be full rank, and recalling (4.18):

$$L(t)=AY(t)\left(I+C^TQ^{-1}CY(t)\right)^{-1}C_2^T$$

gives:

$$\begin{bmatrix}\hat{x}(t+1)\\ \lambda(t)\end{bmatrix}=\begin{bmatrix}A\left(I+Y(t)C^TQ^{-1}C\right)^{-1} & B_2B_2^T\\ -\gamma^2C^T\left(Q+CY(t)C^T\right)^{-1}C & \left(I+C^TQ^{-1}CY(t)\right)^{-1}A^T\end{bmatrix}\begin{bmatrix}\hat{x}(t)\\ \lambda(t+1)\end{bmatrix}$$
$$+\begin{bmatrix}\gamma^2L(t)\\ \gamma^2\left(Y(t)A\right)^{-1}L(t)\end{bmatrix}y(t)$$

Using (4.22), a control input can be given to this system:

$$\begin{bmatrix}\hat{x}(t+1)\\ \lambda(t)\end{bmatrix}=\begin{bmatrix}A\left(I+Y(t)C^TQ^{-1}C\right)^{-1} & 0\\ -\gamma^2C^T\left(Q+CY(t)C^T\right)^{-1}C & \left(I+C^TQ^{-1}CY(t)\right)^{-1}A^T\end{bmatrix}\begin{bmatrix}\hat{x}(t)\\ \lambda(t+1)\end{bmatrix}$$
$$-\begin{bmatrix}B_2\\ 0\end{bmatrix}u(t)+\begin{bmatrix}\gamma^2L(t)\\ \gamma^2\left(Y(t)A\right)^{-1}L(t)\end{bmatrix}y(t) \tag{B.4}$$

# Bibliography

[1] Hyo-Sung Ahn, Kevin Moore, and YangQuan Chen. Schur stability radius bounds for robust iterative learning controller design. In *Proceedings of the American Control Conference*, pages 178–183, Portland, Orlando, USA, June 2005.

[2] Hyo-Sung Ahn, Kevin L. Moore, and YangQuan Chen. Monotonic convergent iterative learning controller design based on interval model conversion. *IEEE Transactions on Automatic Control*, 51(2):366–371, 2006.

[3] Notker Amann, David H. Owens, and Eric Rogers. Robustness of norm - optimal iterative learning control. In *UKACC International Conference on Control*, pages 1119–1124, Exeter, UK, 1996.

[4] Notker Amann, David H. Owens, Eric Rogers, and Anja Wahl. An $\mathcal{H}_\infty$ approach to linear iterative learning control. *International Journal of Adaptive Systems and Signal Processing*, 10:767–781, 1996.

[5] Suguru Arimoto, Sadao Kawamura, and Fumio Miyazaki. Bettering operation of robots by learning. *Journal of Robotic Systems*, 1(2):123–140, 1984.

[6] Konstatin E. Avrachenkov and Richard W. Longman. Iterative learning control for over-determined, under-determined, and ill-conditioned systems. *International Journal of Mathematics and Computer Science*, 13(1):113–122, 2003.

[7] Tamer Başar and Pierre Bernhard. *$\mathcal{H}_\infty$ Optimal Control and Related Minimax Design Problems – A Dynamic Game Approach.* Birkhauser, Boston, USA, 1991.

[8] Zeungnam Bien and Jian-Xin Xu. *Iterative Learning Control, Analysis, Design, Integration and Applications.* Kluwer Academic Publishers, Boston, Massachusetts, USA, 1998.

[9] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[10] Douglas A. Bristow, Marina Thatayil, and Andrew G. Alleyne. A survey of iterative learning control; a learning-based method for high-performance tracking control. *IEEE Control Systems Magazine*, 26(3):96–114, 2006.

[11] Albrecht Böttcher and Sergei M. Grudsky. *Toeplitz Matrices, Asymptotic Linear Algebra and Functional Analysis*. Birkhäuser Verlag, Basel, Switzerland, 2000.

[12] Jeffrey B. Burl. *Linear Optimal Control, $\mathcal{H}_2$ and $\mathcal{H}_\infty$ Methods*. Addison Wesley Longman, Inc., Menlo Park, CA, USA, 1999.

[13] YangQuan Chen and Kevin L. Moore. Harnassing the nonrepetitiveness in iterative learning control. In *Proceedings of the 41st IEEE Conference on Decision and Control*, pages 3350–3355, Las Vegas, Nevada, USA, December 2002.

[14] Mohammed Dahleh, Alexandre Magretski, and Bassam Bamieh. On $l_\infty$ robust stability and performance with $l_2$ and $l_\infty$ pertubations. *Systems and Control Letters*, 28(1):1–6, 1996.

[15] Dick de Roover. Synthesis of a robust iterative learning controller using an $\mathcal{H}_\infty$ approach. In *Proceedings of the 35th Conference on Decision and Control*, pages 3044–3049, Kobe, Japan, December 1996.

[16] Dick de Roover and Okko H. Bosgra. Synthesis of a robust multivariable iterative learning controllers with application to a wafer stage motion system. *Internation Journal of Control*, 73(10):968–979, 2000.

[17] Dick de Roover, Okko H. Bosgra, and Maarten Steinbuch. Internal-model-based design of repetitive and iterative learning controller for linear mulivariable systems. *International Journal of Control*, 73(10):914–929, 2000.

[18] Branko G. Dijkstra. *Iterative Learning Control, with applications to a wafer-stage*. PhD thesis, Delft University of Technology, The Netherlands, 2004.

[19] Branko G. Dijkstra and Okko H. Bosgra. Convergence design considerations of low order Q-ILC for closed loop systems, implemented on a high precision wafer-stage. In *Proceedings of the 41st IEEE Conference on Decision and Control*, pages 2494–2499, Las Vegas, Nevada, USA, December 2002.

[20] Branko G. Dijkstra and Okko H. Bosgra. Extrapolation of optimal lifted system ILC solution, with application tot a wafer-stage. In *Proceedings of the American Control Conference*, pages 2595–2600, Anchorage, Alaska, USA, May 2002.

[21] John Doyle, Andy Packard, and Kemin Zhou. Review of LFTs, LMIs and $\mu$. In *Proceedings of the 30th Conference on Decision and Control*, pages 1227–1232, Brighton, England, December 1991.

[22] Haluk Elce, Richard W. Longman, Minh Q. Phan, Jer-Nan Juang, and Roberto Ugoletti. Simple learning control made practical by zero phase filtering: Applications to robotics. *IEEE Transactions on Circuits and Systems*, 49(6):753–767, 2002.

[23] B.A. Francis and W.M. Wonham. The internal model principle of control theory. *Automatica*, 12(5):457–465, 1976.

[24] Gene F. Franklin, J. David Powell, and Michael Workman. *Digital Control of Dynamic Systems*. Addison Wesley Longman, Inc., Menlo Park, CA, USA, 1998.

[25] James A. Frueh and Minh Q. Phan. Linear quadratic optimal learning control. *International Journal of Control*, 73(10):832–839, 2000.

[26] Jayati Ghosh and Brad Paden. Pseudo-inverse based iterative learning control for plant with unmodelled dynamics. In *Proceedings of the American Control Conference*, pages 472–476, Chicago, Illinois, USA, June 2000.

[27] Peter B. Goldsmith. On the equivalence of causal LTI iterative learning control and feedback control. *Automatica*, 38(4):703–708, 2002.

[28] Svante Gunnarsson and Mikael Norrlöf. On the design of ILC algorithms using optimisation. *Automatica*, 37(12):2011–2016, 2001.

[29] Hassan K. Khalil. *Nonlinear Systems.* Prentice Hall, Upper Sadle River, New Jersey, USA, 2002.

[30] D.J.N. Limebeer, M.Green, and D. Walker. Discrete time $\mathcal{H}_\infty$ control. In *Proceedings of the 28th Conference on Decision and Control*, pages 392–396, Tampa, Florida, December 1989.

[31] Roel Merry, René van de Molengraft, and Maarten Steinbuch. The influence of disturbances in iterative learning control. In *Conference on Control Applications*, pages 974–979, Toronto, Canada, 2005.

[32] Kevin L. Moore. *Iterative Learning Control for Deterministic Systems.* Advances in Industrial Control. Springer Verlag, London, UK, 1993.

[33] Kevin L. Moore. An observation about monotonic convergence in discrete time, p-type iterative learning control. In *Proceedings of 2001 IEEE International Symposium on Intelligent Control*, pages 45–49, México City, México, 2001.

[34] Kevin L. Moore, YangQuan Chen, and Hyo-Sung Ahn. Algebraic $\mathcal{H}_\infty$ desing of higher order iterative learning controllers. In *Proceedings of the 2005 IEEE International Symposium on Intelligent Control*, pages 1213–1218, Limassol, Cyprus, June 2005.

[35] Desineni S. Naidu. *Optimal Control Systems.* CRC Press, London, UK., 2003.

[36] Mikael Norrlöf. An adaptive iterative learning control algorithm with experiments on an industrial robot. *IEEE Transactions on Robotics and Automation*, 18(2):245–251, 2002.

[37] Mikael Norrlöf. Comparative study on first and second order ILC - frequency demain analysis and experiments. In *Proceedings of the 39st IEEE Conference on Decision and Control*, pages 3415–3420, Sydney, Australia, December 2000.

[38] Mikael Norrlöf and Svante Gimmarsson. Disturbance aspects of iterative learning control. *Engineering Applications of Artificial Intelligence*, 14(1):87–94, 2001.

[39] Mikael Norrlöf and Svante Gunnarsson. Experimental comparison of some classical iterative learning control algorithms. *IEEE Transactions on Robotics and Automation*, 18(4):636–641, 2002.

[40] Mikael Norrlöf and Svante Gunnarsson. Time and frequency convergence properties in iterative learning control. *International Journal of Control*, 75(14):1114–1126, 2002.

[41] Andy Packard and John Doyle. The complex structured singular value. *Automatica*, 29(1):71–109, 1993.

[42] Minh Q. Phan and Richard W. Longman. A mathematical theory of learning control for linear discrete multivariable systems. In *Proceedings of the AIAA/AAS Astrodynamics Conference*, pages 740–746, Minneapolis, Minnesota, USA, 1988.

[43] Sigur Skogestad and Ian Postlethwaite. *Multivariable Feedback Control*. John Wiley & Sons, Ltd., 2005.

[44] Jean-Jacques E. Slotine and Weiping Li. *Applied Nonlinear Control*. Prentice-Hall, Inc., Upper Saddle River, New Jersey, USA, 1991.

[45] G.W. Steward. The decompositional approach to matrix computation. *Computing in Science and Engineering*, 2(1):50–59, 2000.

[46] A. Tayebi, S. Andul, and M.B. Zaremba. Robust iterative learning control design via $\mu$-synthesis. In *Proceedings of the 2005 IEEE Conference on Control Applications*, pages 416–421, Toronto, Canada, August 2005.

[47] Masayoshi Tomizuka. Zero phase tracking error algorithm for digital control. *Journal of Dynamic Systems, Measurements, and Controls*, 109(1):65–68, 1987.

[48] Rob Tousain, Eduard van der Meché, and Okko H. Bosgra. Design strategy for iterative learning control based on optimal control. In *Proceedings of the 40th IEEE Conference on Decision and Control*, pages 4463–4468, Orlando, Florida, USA, December 2001.

[49] Casper van Oosten. Robustness for iterative learning control. Master's thesis, Delft University of Technology, The Netherlands, 2003.

[50] Mark H.A. Verwoerd, Gerrit Meinstra, and Theo J.A. de Vries. On the use of noncausal LTI operators in iterative learning control. In *Proceedings of the 41th IEEE Conference on Decision and Control*, pages 3362–3366, Las Vegas, Nevada, USA, December 2002.

[51] Jian-Xin Xu and Ying Tan. *Linear and Nonlinear Iterative Learning Control*. Lecture Notes in Control and Information Sciences. Springer Verlag, Berlin, Germany, 2003.

[52] Bin Zhang, Danwei Wang, Yongqiang Ye, Yigang Wang, and Keliang Zhou. Experimental study on pseudo-downsampled ILC with time shift along iteration axis. In *Proceedings of the American Control Conference*, pages 244–249, Minneapolis, Minnesota, USA, June 2006.

[53] Kemin Zhou, John C. Doyle, and Keith Glover. *Robust and Optimal Control*. Prentice Hall, Upper Sadle River, New Jersey, USA, 1996.