Université Fédérale

Toulouse Midi-Pyrénées

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut Supérieur de l'Aéronautique et de l'Espace

Présentée et soutenue par :
Oktay KOCAN

le jeudi 17 décembre 2020

Titre :

Feedback via iterative learning control for repetitive systems

Commande par apprentissage itératif pour le contrôle des systèmes répétitifs

---

École doctorale et discipline ou spécialité :

EDSYS : Automatique et Robotique

Unité de recherche :

Équipe d'accueil ISAE-ONERA CSDV

Directeur(s) de Thèse :

M. Charles POUSSOT-VASSAL (directeur de thèse)
M. Augustin MANECY (co-directeur de thèse)

Jury :

Mme Valérie BUDINGER Professeure ISAE-SUPAERO – Présidente
M. Svante GUNNARSSON Professeur Université de Linköping Suède – Rapporteur
M. Marc JUNGERS Directeur de recherche CRAN Nancy - Examinateur
M. Edouard LAROCHE Professeur des Universités Strasbourg - Examinateur
M. Augustin MANECY Ingénieur de recherche ONERA - Co-directeur de thèse
M. Swann MARX Chargé de recherche LS2N Nantes - Examinateur
M. Tom OOMEN Professeur Université technique d'Eindhoven Pays-Bas - Rapporteur
M. Charles POUSSOT-VASSAL Maître de recherche ONERA - Directeur de thèse

Acknowledgements

First, I would like to thank myself for never giving up despite all the challenges I've gone through. Second, I would like to thank my tutors Charles Poussot-Vassal and Augustin Manecy for being great guides, trusting my autonomy and always keeping the motivation up in all cases. Third, I do not want to thank COVID-19 for making my manuscript process lot more difficult.

Yes, I have achieved this ; but again, I am telling myself the same words I said before starting all this : *Don't be the same, be better.*

Table of Contents

List of Figures

List of Tables

Abbreviations

ADRC	Active Disturbance Rejection Control
APC	Automatic Process Control
ASFB	Augmented State Feedback
ASHSS	Adaptive Harmonic Steady-State Control
BIBO	Bounded-Input-Bounded-Output
CD	Compact Disk
DOB	Disturbance Observer
DOBC	Disturbance Observer-based Control
D-type	Derivative-type
E-AFC	Extended Adaptive Feedforward Cancellation
ECC	European Control Conference
EID	Equivalent Input Disturbance
EIIC	Enhanced Inversion-based Iterative Learning Control
EPC	Engineering Process Control
ESO	Extended State Observer
FFT	Fast Fourier Transformation
FO-ILC	Fractional Order Iterative Learning Control
FNS	Functional Nueromuscular Stimulation
FxLMS	Filtered-x Least Mean Square
GLC	Global Lipshitz Continuous
HO-ILC	High-order Iterative Learning Control
IEEE	Institute of Electrical and Electronics Engineers
IFAC	International Federation of Automatic Control
IIC	Inversion-based Iterative Learning Control
ILC	Iterative Learning Control
IMC	Internal Model Control
IMP	Internal Model Principle
IROS	International Conference on Intelligent Robots and Systems
LBCT	Learning Based Controller Tuning
LLC	Local Lipschitz Continuous
LQ	Linear Quadratic
LQG	Linear Quadratic Gaussian
LTR	Loop Transfer Recovery
LTR	Adaptive Feedforward Control

NMP Non-minimum Phase

NO-ILC Norm Optimal Iterative Learning Control

NPZ-ignore Non-minimum Phase Zeros Ignore

PD-type Proportional-Derivative-type

PID-type Proportional-Integral-Derivative-type

PMSM Permanent-magnet Synchronous Motor

PNO-ILC Predictive Norm Optimal Iterative Learning Control

PO-ILC Parameter Optimal Iterative Learning Control

PSRMC Partial State Reference Model Control

P-type Proportional-type

R2R Run-to-run Control

RC Repetitive Control

RHP Right Half-plane

SISO Single-Input-Single-Output

SMC Sliding Mode Control

SOR-ILC Supervised Output Regulation via Iterative Learning Control

SPC Statistical Process Control

SRM Switched Reluctance Motors

STR Self-tuning Regulators

SVD Singular Value Decompositions

TSMC Terminal Sliding Mode Control

UAV Unmanned Air Vehicle

VDP Van der Pole

VSC Variable Structure Control

ZPETC Zero-phase Error Tracking Controller

ZMETC Zero-Magnitude-Error Tracking Controller

Resume

The work presented in this thesis has the purpose of building a bridge between data-based control and conventional control design logics for rejecting complex unknown periodic disturbances acting on linear or a class of nonlinear systems. The main contribution of the thesis is the *triple layer control approach* which shows how to combine learning-type (data-based) control and non-learning-type (conventional) control by means of an identification process as connecting tool in between. This approach is explained in Chapter 5 of the thesis. A particularly motivating point of the proposed approach is that it can properly function under complex periodic unknown disturbances both when the system is linear and nonlinear (performance in linear case is much higher from the one in nonlinear case for which the reasons will be explained in Chapter 5 and the general conclusion). The feasibility of this approach is proven with simulation results.

The structure of the thesis is as follows. In Chapter 1, an introduction is given in order to elaborate on the existing learning-type and non-learning type control methods and portray the development process of the thesis. In Chapter 2, a study is dedicated to the *iterative learning control (ILC)* bibliography where important design concepts for ILC as well as the main ILC method used throughout thesis (i.e. the *norm optimal ILC (NO-ILC))* are introduced. In Chapter 3, a practical application with a real indoor UAV is shown as an introductory study which was carried out for testing the feasibility of NO-ILC and determining whether or not it could be used in the rest of the thesis work. In Chapter 4, the main contribution of the thesis begins to develop with the proposal of a procedure for automatically tuning the linear feedback controllers which is given the name of *learning based controller tuning (LBCT) workflow*. This procedure can be seen as the first attempt to create a connection between data-based control and conventional control throughout the thesis. It is demonstrated with the simulation results of an *a priori unknown* periodic disturbance rejection problem that ILC can be utilised as a powerful tool for simplifying the design of the feedback controllers. This whole process actually corresponds to the transformation of the data-based ILC into a feedback law in an approximate way. Next, in Chapter 5, the ideas presented in Chapter 4 are developed further and the approach that is mentioned above as *triple layer control* is obtained as the main outcome of the thesis.

The final section of the thesis is dedicated to a conclusion that provides some suggestions about future studies in the given topic and it involves the critical view of the author on various points regarding the applications and approaches conducted during the thesis.

Introduction

Contents

One of the main issues of control design can be seen as achieving high controller performance while at the same time keeping the compromises on system stability and robustness as low as possible. The challenging part of such control objective comes into play when the system is subject to unwanted external or internal factors such as disturbances and uncertainties. Not surprisingly, main issues of engineering design have its focus on improving disturbance rejection, robustness and performance which are constantly growing requirements of more rapid and vigorous systems [65]. Handling parametric uncertainties and external disturbances in situations where the exact model of a system or disturbance is hard to be found is of fundamental interest in control system design as well as challenging for in industrial and academic applications [122]. One main source of system uncertainties originates from the modelling of the dynamics which for most real-life systems are usually governed by a set of nonlinear equations. The accuracy of these equations suffer from the lack of understanding by the designer or the available knowledge on the system's physics. Consequently, a common fact in system modelling process is to have some residual uncertainties in the end due to neglected or unmodelled dynamics.

Another source for system uncertainties can be related to the time-variance of the system dynamics. It is possible that the dynamics of the real system gradually deviates from its initial model as a result of some external effects during the operation process (e.g. plant aging). This model inaccuracy may cause an increased uncertainty over operation time leading to unwanted results. Besides modelling uncertainties, internal or external system disturbances are also crucial factors in controller performance. The problem of rejecting unknown disturbances has fundamental consideration in various applications such as nano-positioning, active noise control, sinusoidal disturbances rejection of vibrating structures, control of robotic manipulators and disturbance rejection in gyroscopes [87]. These disturbances can be classified in two groups : periodic and non-periodic (or repetitive and non-repetitive) disturbances. The effect of periodic disturbances usually appear in systems with rotary elements such as CD players and electrical machines due to any imbalance, asymmetry or friction during their operation despite the fact that they are designed to move at a constant angular speed [33]. On the other hand, non-periodic disturbances do not necessarily belong to a specific group of systems. They exist in almost all practical applications in different forms such as measurement noise, vibrations, windage

and so on. This means that, if not treated well, non-periodic disturbances can still degrade the system performance even though the system is dominated by the periodic ones [104].

The classification of disturbances can go further. The final important one that is worth to mention is the classification made according to whether the disturbance is matched or unmatched. Various practical systems involve unmatched disturbances (i.e. the control inputs and disturbances enter the system through different channels) which is still a challenging factor for the robustness of many methods [65].

1.1 Control of repetitive systems

In repetitive systems, the periodic disturbance rejection usually appears as the primary focus and the non-repetitive disturbance rejection is taken into account as an additional step for further improving the performance. Accordingly, the main focus of the sections hereafter is the periodic disturbances. To begin, one can find a notable amount of methods in the literature dedicated to suppress the effect of periodic disturbances in repetitive systems. One categorisation of these methods can be done considering whether they use the available repetitive data or not, i.e. to divide them into two categories : *non-learning-type methods* and *learning-type methods*. Table 1.1 provides vast majority of the existing methods in both of these categories (note that the presented classification is according to the author's point of view as a control engineer). Here, it can be seen that the number of methods for the non-learning-type category is larger than that of the learning-type one. This is due to the fact that non-learning-type methods are not specifically designed for repetitive systems only while the presented learning-type methods are solely developed for repetitive systems.

Let us now give details on both categories.

1.1.1 Non-learning-type methods

The first method is the most traditional one in the non-learning-type category which is called the *internal model control (IMC)*. It is based on a feedback or feedforward control design using the *internal model principle (IMP)* given in 1976 by B. A. Francis and W. M. Wonham in [43]. The idea of IMP can also be seen as a reformulation of *the good regulator theorem* by Conant and Ashby from 1970 which states that "every good regulator of a system must be a model of that system" [32]. According to the IMP the disturbance can fully be rejected if its model can be properly acquired and placed inside the controller [87]. In case of a reference tracking problem, the same logic requires the reference signal to be known. If the model of a disturbance (or a reference signal) is known, this means that it can be written by means of known differential or difference equations or it can be produced by an exo-system. In the most conventional sense, a repetitive system can be controlled by designing a state-feedback or output feedback controller based on IMP. However, the difficulty of finding the required feedback is proportional to the amount of unconsidered nonlinearity, uncertainty or process dynamics in the internal model. Therefore, the IMC

TABLE 1.1: Methods classification for repetitive system control

Control of repetitive systems	
Non-learning-type methods	Learning-type methods
Internal Model Control	Repetitive Control (RC)
- Forward model IMC	Iterative Learning Control (ILC)
- Inverse model IMC	Run-to-run Control (R2R)
Robust Control	
- High gain feedback control	
- H_∞ loop shaping	
- Variable structure control (VSC)	
- Sliding mode control (SMC)	
- Terminal SMC (TSMC)	
- Loop transfer recovery (LTR)	
- Probabilistic robust control	
Adaptive control	
- Adaptive Feedforward Cancellation(AFC)	
- Model Reference Adaptive Control (MRAC)	
Disturbance Observer (DOB)	
- Active Disturbance Rejection Control (ADRC)	
- Extended State Observer (ESO)	
- Equivalent Input Disturbance (EID)	
Backstepping	

design can be quite a tedious process under complex situations. Also, the IMC can yield both the asymptotic rejection of the disturbances and the asymptotic tracking of reference signals which are alike and this feature can bring about unwanted side effects in case of joint disturbance and reference signals [50]. This method family will play an important role in this manuscript later and the details will be given.

The second method in this category is the *robust control* that has been developing since 50s. The main idea of robust control is to design static control policies that are able to deal with uncertainties and disturbances until a certain bound. This means assuming in the control design stage that the controller will undergo some limit of parameter variation due to internal/external unknown effects. The earliest attempts of robust approach began by trying to improve stability margins of *linear quadratic Gaussian (LQG)* controller using *loop transfer recovery (LTR)* [65]. Then, the robustness of the LTR method was found to be insufficient in case of nonminimum phase systems which lead to the H_∞ optimisation approach [65]. The method is also called H_∞ loop shaping by some and it is based on lowering the sensitivity of the system over its frequency spectrum such that it will be less effected by any uncertainty or unknown disturbance entering the system later on. However, H_∞ optimisation approach suffer from algebraic and analytic constraints (e.g. limited information on uncertainties and disturbances, unmodeled dynamics and right half-plane (RHP) zeros) which together reduce the attainable robustness and performance

requirements of the control system [65]. Another robust control method that dates back to early 80s is the *high-gain feedback control* (see [111, 112]). It is a method that stems from the idea that any adverse effect or variation on the system can be diminished applying a sufficiently high feedback gain. Yet, such approach has significant drawbacks in real-life applications since it becomes a challenge to avoid control saturation (and thus closed-loop stability) and excitation of high-order unmodelled dynamics [27]. In addition, it can usually make it difficult to deal with the noise by amplifying it.

The third method in this category is the *variable structure control (VSC)* that is based on discontinuous nonlinear control. It is a very old method invented in 50s by Emelyanov in the Soviet Union (see [41]). The fundamental idea behind this approach is to design a control law that changes its structure depending on the system's position in the state-trajectory. This induces a quick switching behaviour between multiple smooth control laws with respect to the position in state-trajectory. The most unique property of VSC can be seen as yielding very robust control systems that are almost fully invariant to external disturbances and parametric uncertainty [53]. In the literature, the applications of VSC appear generally under the extension of *sliding mode control (SMC)* and in more recent approaches via *terminal sliding mode control (TSMC)*. An SMC has two modes : (1) the *reaching mode (nonsliding mode)* which brings the trajectories starting from anywhere in the state-space to a switching line in finite time and (2) the *sliding mode* which moves the trajectory asymptotically to the origin [106]. Besides the praises, SMC is usually critiqued because of the chattering phenomenon (high-frequency oscillations) due to switching nonidealities and the need of high gain for unknown disturbances (see [120] for details).

The fourth method in this category is the *adaptive control* which came from the idea of creating controllers that modify themselves with the changing system parameters or the varying external effects. The adaptive control branch has been developing since 50s bringing out many different adaptation strategies. The earliest approaches include *gain scheduling*, *model reference adaptive control (MRAC)* and *self-tuning regulators (STR)* which have a common framework (a normal feedback loop plus a regulator with adjustable parameters) that only differs in the way they adjust parameters of the regulator [12]. The challenge of adaptive control is thus to find an efficient update law for changing the regulator parameters in response to variations in plant and/or disturbance dynamics. The main idea of gain scheduling is to find a set of gains that will produce the correct controller for each operating condition. The advantage of gain scheduling is that it allows a very rapid controller adjustment in response to quick process variations, but it still has some drawbacks such as having a time consuming design procedure where the gains are calculated via extensive simulations at each operating condition and being an open-loop compensation that means an incorrect parameter adjustment can not be corrected [12]. Moreover, the MRAC and the STR approaches are similar in ultimate goals but they differ in their functioning. The MRAC is based on direct parameter adjustment for the regulator via processing the plant's input, output and a reference model signal representing the desired behaviour in a deterministic manner. On the other hand, the STR is based on controller parameter adjustment based on processing the online recursive estimation of the plant current plant parameters. The detailed background on both methods can be found in [12, 39, 51].

Apart from these methods, the *adaptive feedforward cancellation (AFC)* is a very popular approach extended to unknown periodic disturbances. It is based on estimating the magnitudes and phases of the periodic disturbance harmonics and adding the negatives of them at the input of the plant such that they do not appear in the output [20] (see also [18, 19, 33, 58] for more details). A more developed version of AFC called the *extended AFC (E-AFC)* designed for unknown and time-varying periodic disturbances can be found in [72] where also various other adaptive algorithms such as *filtered-x least mean square (FxLMS)-based algorithms* and *adaptive harmonic steady-state control (ADHSS)* are also mentioned. The main drawback in AFC algorithms is usually the difficulty of the tuning of the gain values for unknown multi-sunisoidal time-varying disturbances [72]. Although one can find many more adaptive rejection algorithms such as the *adaptive control with Q-parametrization* and *partial state reference model control (PSRMC)* as given in [63], they are left to the reader's curiosity.

The fifth method in this category is the *disturbance observation/estimation-based control (DOBC)* (see [50]). The idea of using DOBC came up with an aim of establishing linear control laws using disturbance estimates [122]. One of the most popular DOBC approaches can be seen as the *active disturbance rejection control (ADRC)* which was proposed in late 90s. ADRC is a model free approach that can inherently deal with attenuating unknown disturbances, uncertainties and nonlineaties which was supported by simulation and experimental results in various applications [87]. The idea behind ADRC is the extension of the system model with a state-variable that puts together all the unconsidered disturbances and uncertainties inside the system model which is then used in a nonlinear feedback to provide rejection [89]. The estimation component of ADRC is called the *extended state observer (ESO)* which observes both the states and external disturbances without needing a model. Another promising disturbance estimation-based method is the *equivalent input disturbance (EID)* (e.g. [97] and [96]) that is also capable of rejecting challenging unknown disturbances by only using the input and output of the plant without any a priori information of the disturbance. The fundamental idea behind EID is based on estimating the disturbance on the control input channel rather than directly estimating it which seems more practical since in the end it is the control signal that is used for producing required rejection [97].

As the final non-data based method, one can count the famous *backstepping* method which is a recursive technique for stabilising usually specific class of nonlinear systems. The main idea is to reduce an initial system until a stable subsystem is reached and then to design backwards controllers for each outer system until the controller for the initial system is reached. The main disadvantage of backstepping control can be seen as the lack of robustness against system parameter variations (seethe conclusion of [40]). Furthermore, an example of a robust backstepping algorithm that can reject bounded uncertainties with unknown periodicity in the presence of external disturbances can be seen in [62]. Moreover, it is also possible to encounter in some works such as [45, 57, 113] the combination of backstepping and ADRC approaches. However, further details are beyond the scope of this thesis work.

1.1.2 Learning-type methods

> The learning-type control methods should generally be the first preferred option for repetitive (periodic) and/or cyclic processes [110]. The reason behind such choice is the large amount of already available data acquired during the operation of these repetitive processes. Such data contains a rather useful knowledge about all the repeating behaviour within the process which can be used to create a system that anticipates for them in the following repetitions. Neglecting this type of advantage in repetitive systems can actually be as throwing over a huge precision performance, especially in a time when the data-storage and powerful data manipulation are widely accessible.

The control tasks of many practical applications have a periodic nature such as CD or magnetic disk drives, machine tools, the operations of picking, placing or painting via industrial machines etc. where the goal is to reject or track periodic exogenous signals while at the same time keeping the performance high in terms of productivity, quality and precision [35]. In the literature, this control problem has been so far addressed by three different learning-type control methods : *repetitive control (RC)*, *iterative learning control (ILC)* and *run-to-run control (R2R)*. Each of these methods were initially proposed by different authors to solve different problems in different fields such that their characteristics, specialisations and formulations are different despite the fact that each method is based on enhancing the control performance via learning from past experience [110].

The nature of the repetitive process in terms of production or functioning needs is one of the main characteristics defining the distinctions between the learning-type methods. An industrial process is usually defined as either a continuous process or a batch process. Originally, the RC and the R2R methods were designed for continuous processes whereas the ILC method was aimed for batch processes.

> In a continuous process the product is delivered continuously which is in contrast with a batch process where the product comes out discontinuously in groups. For instance, the batch processes are the most convenient processes for the discontinuous production of high-value and low-volume products while the continuous processes befit better the continuous production of high-volume products [110]. The term 'continuous process' should not be mixed hereafter with the term 'continuous system (or continuous-time system)' which is a totally different classification meaning that the time variables of the system have continuous values. A brief explanation and comparison of RC, ILC and R2R is given below.

The repetitive control (RC) technique was first proposed in 1981 by the work [54] where the authors achieved a high tracking precision of a periodic reference within 16 cycles of pulsed operation. Later many other researchers kept utilising the method also for the rejection of periodic disturbances. The earliest versions of RC are based on internal model principle (IMP) and they are made of two components : a periodic signal generator and a stabilising controller for the closed loop system (the initial methods used in the stabilising controller include stable pole/zero cancellation, zero phase error method, low-pass FIR filter and pole placement) [118]. It is clear that the initial attempts of RC suffer from the necessity of knowing the period of the periodic signals. In most scenarios, this

information may not be available a priori or simply it can be slowly varying which caused researchers to search for adaptive RC methods that recursively identify the period of the periodic signal and re-tunes the signal generator to make it possible to reject the unknown periodic disturbance [118]. The main difference of RC from ILC and R2R is that it is a feedback-based approach that can be directly integrated inside a pre-designed feedback loop [49]. This feature allows RC to be easily applicable to continuous processes. The theoretical foundation of RC concept, which comes from IMP, is based on the fact that one can generate any periodic signal with period T via utilising a free time-delay system with a proper initial function [110]. The general form of RC for a discrete-time system is given in the z-domain as below,

$$U(z^{-1}) = \frac{z^{-T} L_{RC}(z^{-1})}{1 - z^{-T} Q_{RC}(z^{-1})} E(z^{-1}) \tag{1.1}$$

where z^{-1} is the backward shift operator in z-domain, z^{-1} is the backward shift by the amount of the period T, $E(z^{-1}) = R(z^{-1}) - Y(z^{-1})$ is the system error ($R(z^{-1})$ and $Y(z^{-1})$ are the system reference and the system output, respectively), $U(z^{-1})$ is the system input, $L_{RC}(z^{-1})$ is the *learning filter (or L-filter)* and $Q_{RC}(z^{-1})$ is the *Q-filter*. The block diagram of this general RC system is illustrated in Figure 1.1. The control objective of RC is thus to find proper $L_{RC}(z^{-1})$ and $Q_{RC}(z^{-1})$ values or forms such that the tracking error $E(z^{-1})$ converges to zero as the time goes to infinity ($t \to \infty$). This is another distinction between RC and the other two learning methods. Its design is done in frequency-domain and the stability and convergence performance analysis are assessed according to process duration in time axis. Without going into further details, some commonly utilised RC filters (see [110]) can be provided below :

$$L_{RC}(z^{-1}) = K_{RC} \quad or \quad L_{RC}(z^{-1}) = K_{RC}(1 - z^{-T}) \tag{1.2}$$

$$Q_{RC}(z^{-1}) = \alpha_1 z + \alpha_0 + \alpha_1 z^{-1} \tag{1.3}$$

where $K_{RC} \in \mathbb{R}$ is a gain value and $\alpha_0 \in \mathbb{R}^+$ and $\alpha_1 \in \mathbb{R}^+$ are filter coefficients with $2\alpha_1 + \alpha_0 = 1$. The L-filter is used for filtering the error (i.e. determines RC's convergence rate) while the Q-filter has the purpose of increasing the RC's robustness. In the following sections, it will be seen that the same logic exists for the aim of ILC filters.

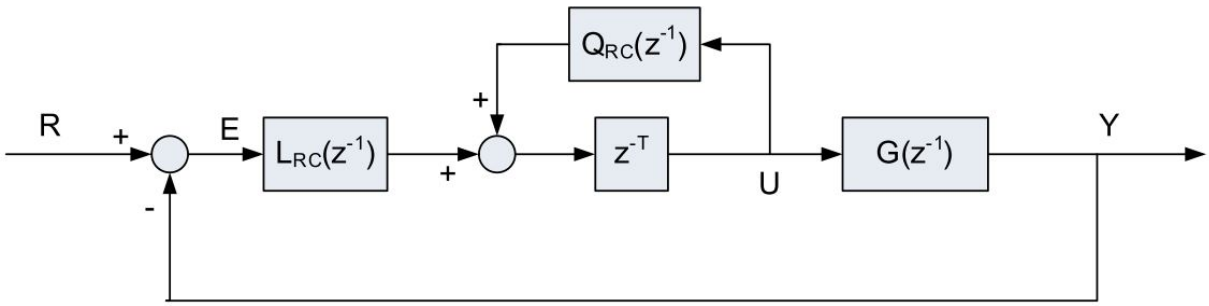

FIGURE 1.1: An example of a repetitive controller [110]

The run-to-run control (R2R) is a control technique that emerged in 90s specifically for the batch processes in the semiconductor industry after the work of Sachs et al. in [94].

R2R (or *run-by-run*) process control is a term used for naming the methods that combine both the *statistical process control (SPC)* and *engineering process control (EPC)* in the semiconductor industry [37]. SPC is a method that is based on process output monitoring, *out of control* process detection and causal attribution (e.g. a process is out of control if the output variation can be assigned to a cause such as a disturbance) [25]. The regulation of the output variation is obtained by varying the input (or set-point) with a control action and this task is not included in SPC, i.e. left to the control engineer. The widespread applications of SPC can be observed in discrete parts manufacturing where the processes in general have natural variability and high repetitiveness. On the other hand, the EPC (or *automatic process control (APC)*) is a widely used method in chemical processing that is based on transforming the output variability into the input control variable utilising the measurements of crucial process variables within a feedback law [25]. The main aim of R2R control is to benefit from the features of both SPC and EPC (i.e. process monitoring and measurement (observation) based control) in order to regulate the output variability. R2R controllers are used to determine the recipe for the batch processes (i.e. how the process should be updated) and they are usually model-based controllers with an observer integration [25]. The main motivation behind R2R was the difficulty of having constant *in situ* measurements (i.e. the measurements done while the process is running) in semiconductor manufacturing such as the film thickness of wafers, the gas flow rate of the equipment etc. [94]. Similar to ILC, R2R updates the recipe between the runs (or batches/trials) utilising a unified approach of feedforward, feedback and local optimisation (note that here updating the recipe can be interpreted as updating the process input). The feedforward feature is used for adjusting the recipe for the next process run based on the previous process measurement data which is analogous to ILC. Nevertheless, in R2R applications, sparsely available output measurements are sufficient for performing the control update whereas in ILC, multiple measurements are needed [110]. Furthermore, the feedback feature of R2R helps to maintain the system at the local optimum of the recipe (i.e within some desired performance) under the effect of disturbances. The general structure of an R2R can be observed in Figure 1.2 and the more details regarding the method can be found in [73] which is a the recent survey.

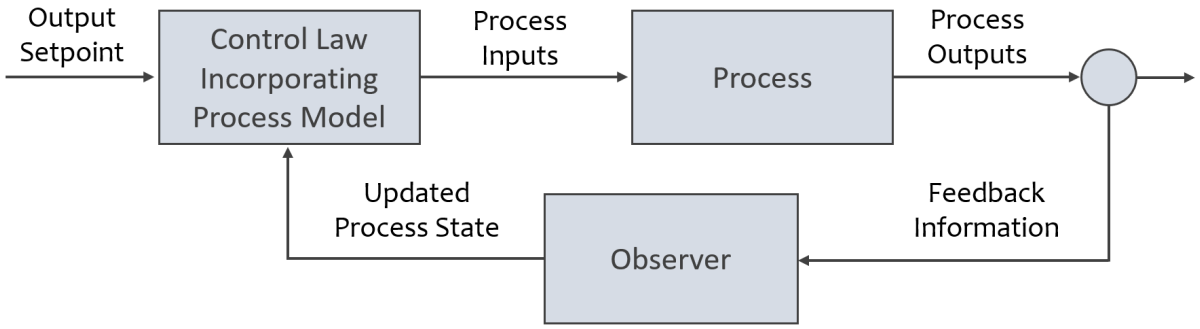

FIGURE 1.2: An example of a run-to-run controller [73]

The iterative learning control, that is the final learning-type method of the section, is the core of the thesis work and therefore, it is introduced extensively in the following sections.

1.2 Iterative learning control

The concept of iterative learning can be seen as an adaptation of human way of learning to machines. In humans, the learning of a skill is based on repetitions of the same practices or tasks over a period of time. In each repetition, the human gets to know the mistakes that are reoccurring and the improvement towards the skill depends on how these mistakes are processed by the person. In the sense of iterative learning control (ILC), the 'skill' is a specific goal of a repetitive (periodic) system such as achieving a certain level of tracking or rejection performance. The 'reoccurring mistakes' are the repeating system errors remaining after each trial of the same task. Finally, the 'improvement' (i.e. the gradual learning) is the result of ILC's performance in correcting these repeating system errors.

1.2.1 Historical background

Iterative learning control has been a topic in the literature for more than 35 years. Some authors say that the first ILC approach is even older by mentioning a U.S. patent filed by Garden in 1967 [44]. Besides, there are other authors mentioning that the idea of ILC begins by the work of Uchiyama published in 1978 [105]. Although controversial, the vast majority of researchers still accept that main attraction towards ILC started after the journal paper written by Arimoto et al. in 1984 [9]. Some other works that also popularised the ILC in the same year were the ones of Craig [34] and Casalino and Bartolini [26]. Finally, later in the same year, another work of Arimoto [61] used the name "Iterative Learning Control" for the first time and since then, this name has been used by all researchers. In 2020, the importance of ILC can easily be understood by looking at the total number of available ILC publications at the moment. For the keyword "iterative+learning+control", IEEE Xplore website currently provides 4192 publications containing conference papers, articles in journals, magazines and books. Similarly, the number of publications reaches 9780 for "iterative+learning", 20480 for "iterative+control" and 62456 for "learning+control". This clearly shows that the topic of iterative learning based control has become a rather active research area over the years. Moreover, it is worth to mention that [1, 15, 29, 78, 86, 115, 116] are some fundamental books written about ILC theory and applications between the years 1993 and 2016. Plus, several surveys such as [1] and [80] are useful sources for understanding the development of ILC approaches over the years.

1.2.2 Fields of use

ILC has been utilised in various applications in different fields over the years. The initial focus of the ILC works was to improve the performance of industrial robots in repetitive tasks. Later on, the applications branched into many other fields such as rotary systems, batch processes, factory processes, chemical processes, biomedical and bioengineering applications, actuators, semiconductors, power electronics, UAVs, high-speed trains and so on. A detailed list of ILC applications in these fields are provided below on Table 1.2. Except the applications written in 'bold', the information presented on Table 1.2 is based

on [1] which gives some paper references for each application. Therefore, here only the 'bold' applications are provided with the references.

1.3 Development of the thesis work

In non-learning-type control, the high-precision tracking and the accommodation of unknown periodic disturbances (or uncertainties) most of time require tedious analysis. The results are usually achieved with some limitations and the controller tuning can be quite conservative due to the mathematical assumptions needed by the methods. Moreover, even though the desired asymptotic performance is acquired, these methods cannot improve the transients in the output since they do not have an anticipation property. On the other hand, the iterative learning control, with its very simple update rule, is capable of providing very high-precision results in tracking of periodic trajectories and it can remove challenging periodic disturbances (including the transients) without needing to know their frequency content. This property of ILC has been one of the main motivations for beginning this thesis.

The global objective followed throughout the thesis can be seen as facilitating the conventional ILC approaches from a practical point of view. The initial preparation for the thesis was done during a master thesis study at Isae-Supaero based on a 6-month internship at Onera-Toulouse (February 2017-August 2017). Since October 2017, the thesis has led to the following contributions in the chronological order :

Contribution 1 (**Madrid, 2018 - Rank : A1**) IEEE/RJS International Conference on Intelligent Robots and Systems (IROS)
Subject : A Practical Method to Speed-Up the Experimental Procedure of Iterative Learning Controllers

Contribution 2 (**Naples, 2019 - Rank : B1**) European Control Conference (ECC)
Subject : Towards the Automatic Tuning of Linear Controllers Using Iterative Learning Control Under Repeating Disturbances

Contribution 3 (**Berlin, July 2020 - Rank : B4**) International Federation of Automatic Control (IFAC) World Congress
Subject : Supervised Output Regulation via Iterative Learning Control for Rejecting Unknown Periodic Disturbances

Contribution 4 (on-going) Nonlinear periodic disturbance rejection by internal model control : a mixed learning and data-driven approach.

(*Note : the conference rankings are based on 'Qualis (2012)' measure given at the following website : http ://www.conferenceranks.com*)

The first contribution was based on a real application with an indoor UAV. This work had a purpose of forming a base for the other two contributions in terms of understanding the ILC theory and testing its efficiency in practice. The ILC method that was tested in the work was the norm-optimal iterative learning control given by [84] which was proven to be highly performant in learning agile repetitive trajectories. The contribution to the literature was through the proposal of a new data-flow for speeding-up the ILC experi-

mentation processes with UAVs that had an aim of changing the conventional ILC update procedure. The proposed data-flow can be considered unique in the sense that it added a step of simulation-based ILC updates that utilises a system re-identification with simple transfer functions before each real ILC iteration (i.e. one UAV flight).

The second and third contributions are oriented towards utilising the power of ILC for detecting frequency content of unknown disturbances and building feedback controllers from this information. The third contribution is a continuation of the work presented in the second contribution. To begin, the second contribution was based on using ILC to create feedback controllers for accommodating unknown periodic disturbances. This procedure is equivalent to automatising the gain tuning of linear feedback controllers. The contribution to the literature was through the introduction of a workflow called *Learning Based Controller Tuning (LBCT)* which was proven to be effective for the unknown periodic disturbances having nonlinear structure. The main motivation behind this work was to encourage an automated tuning approach in designing linear controllers against periodic disturbances by means of using ILC's two advantageous features : the model-freeness and the simple integration to existing systems without modifying their parameters. In the third contribution, the same logic was extended to the creation of a new framework that combines ILC with a robust output regulator. The main motivation of this work was the possibility of a promising match of ILC with the robust output regulator scheme given in [10]. The general idea can be seen as to improve the rejection (or tracking) performance by combining the power of both methods. ILC was simply used to improve the restrictive part of [10] that is the need of knowing the period of the periodic disturbance. On the other hand, robust output regulation is used for increasing the system robustness to nonrepetitiveness (e.g. instant perturbations, small variations around learned reference trajectory, uncertainties and noise) by using the information learned via ILC in building an internal model controller. The combination of both methods was given under the name of *Supervised Output Regulation via Iterative Learning Control (SOR-ILC)*. The proposed SOR-ILC framework is a transformation of ILC into a feedback controller with additional features of robustness and precision that were proven to be much better than the both methods performing alone or separately.

Currently in the fourth contribution, the work done in the third contribution has been applied on a nonlinear system and the idea of disturbance rejection via SOR-ILC has been put in a generic frame called *triple-layer control*. This new work aims at motivating the use of learning-type control, identification and non-learning type control in a combination. The generic presentation of triple layer approach indicates that one may also apply the same logic by using different learning-type and non-learning-type methods in combination than limiting it to the combination of ILC and robust output regulation (SOR-ILC). The numerical application of this work is based on rejecting a nonlinear Van der Pol oscillation acting on a stable bilinear nonlinear system. The chosen triple layer method is still SOR-ILC and the improvements consist of a prestabilisation step via nonlinear feedback as well as an identification method based on *Hankel* matrices that can obtain an accurate model approximation by using time domain data only (i.e. no Fast Fourier Transformation and frequency domain data needed).

A summary for the development of the thesis can be found in Figure 1.3 and the

outline of the thesis is provided as follows. **Chapter 2** is dedicated to the bibliography notes. It introduces the theory of ILC with demonstrative examples and presents some sample systems used throughout the manuscript, the main algorithms and the notations utilised in the rest of the sections where the main thesis study is shown. Thus, **Chapter 3**, **Chapter 4** and **Chapter 5** give detailed explanations for the contribution 1, contribution 2 and contribution 3, respectively (Chapter 5 also includes the unpublished results from the above mentioned on-going work, see Article 4 in Figure 1.3). Finally, the conclusion talks about how the work could be further improved while at the same time providing the critical views of the author on the results and the overall approach.

TABLE 1.2: The various fields of use of ILC

Existing ILC applications w.r.t their fields of use	
Field	**Application**
Robotics	General robotic applications (both rigid and flexible manipulators) Robot applications with adaptive learning Robot applications with Kalman filtering Arc welding process Microscale robotic deposition Impedance matching in robotics Mobile robots Underwater robots Cutting robots Acrobat robots Gantry robots Table tennis
Batch/Factory/Chemical Processes	Batch process Agile batch manufacturing processes (product quality tracking control) Chemical reactors Chemical processes Industrial extruder plant Packaging and assembly Injection molding Moving problem of liquid container Laser cutting Water heating system

Rotary Systems	Vibration suppression of rotating machinery AC servo motor Permanent-magnet synchronous motor (PMSM) Linear motors Electrostrictive servo motor (Ultrasonic) induction motor Switched reluctance motors (SRM)
Bio-applications	Biomaterial applications Human operator Artificial muscle Biomedical applications (e.g dental implants) Functional nueromuscular stimulation (FNS) Smart microwave tube Pneumatic system **Post-stroke limb rehabilitation [77]**
Actuators	Hydraulic cylinder system (Proportional-valve-controlled) Electromechanical valve Piezoelectric actuator hysterisis Linear actuators
Semiconductors	Manufacturing processes
Power electronics	Electronic/industrial power systems Inverters
Other	Identification of aerodynamic coefficients **Engines [38, 81]** Hard disk drives Visual tracking Dynamic load simulator Uniformity control of temperature quantum mechanical system Piezoelectric tube scanner Magnetic bearings **High-speed trains [52, 56, 71]** **UAVs [70, 95, 123]** **Printer precision [17, 119]**

Development of the Thesis

February 2017

MSc Internship (6 months)

PhD Studies (3 years)

October 2020

	ARTICLE 1	ARTICLE 2	ARTICLE 3	ARTICLE 4 (On-going)
CONTENT	Application of a existing ILC algorithm to an indoor UAV in order to learn agile reference tracking.	Using ILC to make a linear feedback controller (***automatic*** gain tuning of a linear feedback controller) to reject **complex unknown disturbances**.	Integration of ILC to IMC-based output regulation given by **[10]** in order to reject ***unknown nonlinear disturbances*** *and* ***uncertainties***.	Combination of learning-type control, identification and non-learning-type control methods in order to reject ***unknown nonlinear disturbances***.
CONTRIBUTION	New experimental procedure for speeding up ILC application to UAVs.	***Learning-based Controller Tuning (LBCT)*** workflow in order to *automatically tune* the linear feedback controllers.	***Supervised Output Regulation via ILC (SOR-ILC)*** for combining the precision of ILC with robustness of the feedback scheme in **[10]** **.**	Reframing the SOR-ILC approach and proposing it under a more generic and improved frame called the ***triple layer*** control approach.
SPECIFICATIONS				
CONTROL OBJECTIVE	Tracking control of agile trajectories for a ***real*** indoor UAV.	Rejection control of unknown linear and nonlinear disturbances acting on a ***linear*** system.	Rejection control of ***nonlinear lumped disturbances*** under ***parametric uncertainty*** acting on a ***linear*** system	Rejection of ***nonlinear disturbances*** (e.g. ***Van der Pole Oscillations***) acting on a ***nonlinear*** system.
DISTINCTION	ILC is applied in open-loop to a UAV system which initially has a stabilizing closed loop controller.	A new workflow (LCBT) showing how to utilise ILC to built ***automatically*** tuned linear controllers is presented.	The missing part of **[10]** , i.e. requirement of ***a priori*** knowledge of the disturbance frequency, is completed via the integration of ILC	SOR-ILC approach is generalised allowing the possibility of using other learning-type and non-learning-type control methods and a new identification method using only time domain data is integrated.
RESULT	Efficiency of the optimization-based ILC (***Norm Optimal ILC***) with the proposed experimental procedure is demonstrated on a real UAV system.	Possibility of transforming ILC into an feedback controller is demonstrated.	Efficiency of turning ILC to feedback controller via the new SOR-ILC is demonstrated by comparing results to ILC-only case.	SOR-ILC is proven to be applicable to a ***nonlinear*** system under nonlinear disturbance and it is shown that an identification method based on ***Hankel*** matrices can provide precise model approximation without needing a Fast Fourier transformation.

FIGURE 1.3: Development of the thesis work through the contributions

Chapter 2

Iterative Learning Control

Contents

The purpose of this chapter is to provide the reader with some insight on the basic linear ILC theory. The topics mostly consist of different ILC representations, ILC stability analysis, ILC integration, popular ILC methods and some demonstrative numerical applications. It should be noted that the given content is not a deep review of ILC but rather an introduction to essential ILC design elements and spractical points.

2.1 Basic ILC architecture

An ILC can be defined both in continuous-time and discrete-time. However, since all real-life applications are performed using computers that work in discrete-time, the main preference in ILC literature is towards the discrete-time. Accordingly, let us consider the following discrete-time, linear time-invariant (LTI), single-input-single-output (SISO) system :

$$y_i(t) = P(q)u_i(t) + d(t) \tag{2.1}$$

where

- t is the time index,
- i is the iteration index,
- q is the forward time-shift operator $qf(t) = f(t+1)$,
- u_i is the control input or the system input at i-t iteration,
- y_i is the system output at i-t iteration,
- $d(t)$ is a signal for including repeating elements at each iteration (e.g. periodic disturbances, non-zero initial conditions, augmented feedback and feedforward controllers [22]),
- $P(q)$ is a proper rational function of q which has a relative degree of m (i.e. a delay of m samples).

It can be seen that for each iteration $i \in \mathbb{Z}^+$ the system (2.1) has a new input and a new output. If a sequence with $M \in \mathbb{Z}^+$ samples is considered, $t \in \{0, 1, ..., M-1\}$ and $u_i(t)$ is defined with the following sequence :

$$u_i(t) := \{u_i(0), u_i(1), u_i(2), ..., u_i(M-1)\} \tag{2.2}$$

Since the plant $P(q)$ has a relative degree of m, it causes a delay of m samples in the sequences of $y_i(t)$ and $d_i(t)$ such that

$$y_i(t) := \{y_i(m), y_i(m+1), y_i(m+2), ..., y_i(M+m-1)\} \tag{2.3}$$
$$d_i(t) := \{d_i(m), d_i(m+1), d_i(m+2), ..., d_i(M+m-1)\} \tag{2.4}$$

This also means that one should take into account the delay m when choosing the sequence for the system reference $r(t)$ (or equivalently the system's desired output sequence), i.e.

$$r(t) := \{r(m), r(m+1), r(m+2), ..., r(M+m-1)\} \tag{2.5}$$

Then, subtracting (2.5) from (2.3), gives the system error sequence

$$e_i(t) = r(t) - y_i(t) := \{e_i(m), e_i(m+1), e_i(m+2), ..., e_i(M+m-1)\} \tag{2.6}$$

Let us now introduce the simplest version of ILC (see Figure 2.1), i.e.

$$u_{i+1}(t) = Q(q)[u_i(t) + L(q)e_i(t)] \tag{2.7}$$

where

- i denotes the data from previous iteration,
- $i+1$ denotes the data for the current iteration,
- $L(q)$ is the ILC learning function,
- $Q(q)$ is the *Q-filter* (e.g. a low-pass filter).

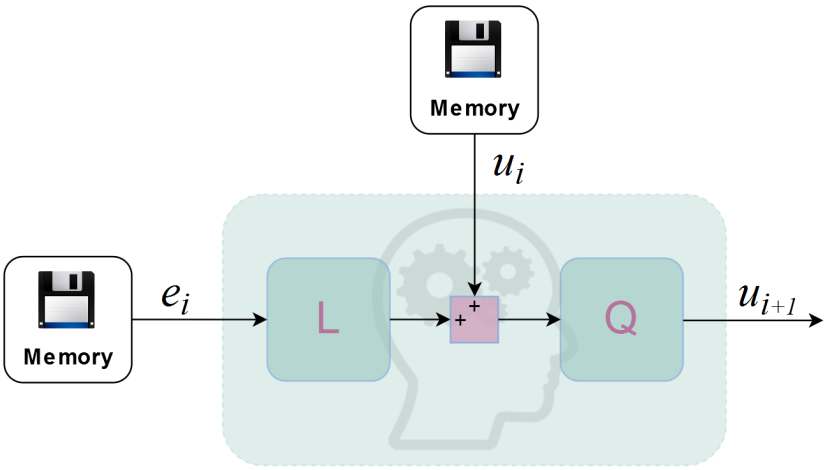

FIGURE 2.1: Basic ILC controller structure

Equation (2.7) is called the *update equation* which determines the learning dynamics. Together with (2.1) it forms a two-dimensional system. Thus, an ILC system is composed of two dimensions : time (or frequency) domain and iteration domain. The time or frequency domain shows the behaviour of the plant dynamics (2.1) while the iteration domain demonstrates the learning behaviour of ILC (2.7). Moreover, the initial conditions have crucial importance for ILC since, in its simplest form, it requires the repetitive task to always begin from the same initial condition. Hence, for a better analysis of the initial conditions, ILC can be designed alternatively considering the state-space equivalent of (2.1). For a SISO linear system with n states, the state-space equations are given as follows :

$$x_i(t+1) = Ax_i(t) + Bu_i(t) \tag{2.8}$$
$$y_i(t) = Cx_i(t) + Du_i(t) \tag{2.9}$$

where $x_i(t) \in \mathbb{R}^n$ denotes the state sequences with the initial condition $x_i(0) = x_0$ for all i. The matrices $A \in \mathbb{R}^{n\times n}$,$B \in \mathbb{R}^{n\times 1}$, $C \in \mathbb{R}^{1\times n}$ and $D \in \mathbb{R}^n$ stand for the state (or system) matrix, the input matrix, the output matrix and the feedthrough (of feedforward) matrix, respectively. For simplicity, it can be assumed that there is no feedthrough in the system, i.e. the matrix D is a zero matrix. In this case, the system (2.8)-(2.9) can be written as

$$y_i(t) = C(qI - A)^{-1}Bu_i(t) + CA^t x_0 \tag{2.10}$$

with $I \in \mathbb{R}^{n\times n}$ being the identity matrix. Moreover,

$$P(q) = C(qI - A)^{-1}B \tag{2.11}$$
$$d(t) = CA^t x_0 \tag{2.12}$$

It can be seen that (2.12) includes the initial condition x_0 which, in an ideal ILC system, should remain same at each iteration. This necessity is due to the fact that any variation on x_0 has a direct effect on the efficiency of the learning dynamics. After the ILC signal passes through the plant dynamics $P(q)$, it varies by the amount of $CA^t x_0$ (see the last term of (2.10)). Thus, the variations on x_0 will deteriorate a proper learning process and

they should be properly reset to same values before each iteration starts. Furthermore, the sensitivity to the initial conditions can be detrimental with the increasing number of iterations. This is simply because the ILC update equation is actually a discrete integrator and even a very small initial error can add up to a huge transient error if its integrated enough, namely if the number of iterations are sufficiently large (e.g. $i \to \infty$). This notion is well demonstrated in the 'equation (5)' of the reference [117].

For the sake of generalisation, the ILC architecture until this point has been given in terms of sequences. Let us now introduce the two popular system representations used for ILC design : time-domain ILC representation and frequency domain ILC representation. Both of these representations have their own advantages/disadvantages as well as different interpretations.

2.1.1 Time-domain ILC design

In time-domain representation, the sequences and the proper functions turn into vectors and matrices, respectively. The transition from sequences to vectors is straightforward. On the other hand, the transition from proper functions to matrices requires an infinite power series expansion. To do that, the denominator of $P(q)$ is divided into its numerator such that for an M-sample sequence and a relative degree of m,

$$P(q) = \sum_{k=m}^{M} {h_k}^{-k} = h_m q^{-m} + h_{m+1} q^{-(m+1)} + h_{m+2} q^{-(m+1)} + ... + h_{m+M-1} q^{-(m+M-1)} \quad (2.13)$$

where h_k and the sequence $\{h_m, h_{m+1}, ..., h_{m+M-1}\}$ denote the *Markov parameters* and the impulse response of $P(q)$, respectively [22]. Assuming zero initial condition ($\mathbf{x}(0) = \mathbf{0}$) The impulse response for a state-space system is written as follows :

$$h_k = \begin{cases} D, & k = 0 \\ CA^{k-1}B & k > 0 \end{cases} \quad (2.14)$$

The function $P(q)$ is finally transformed into an $(m+M) \times (m+M)$ matrix by using the *Lifted-System Framework* as follows :

$$\mathbf{P} = \begin{bmatrix} h_m & 0 & \cdots & 0 \\ h_{m+1} & h_m & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h_{m+M-1} & h_{m+M-2} & \cdots & h_m \end{bmatrix} \quad (2.15)$$

It can be noted that the first column of the matrix P is the impulse response sequence of the function $P(q)$ and the rest of the columns are simply obtained by shifting this sequence by one sample at a time until the column number reaches $(m + M - 1)$. Let us now write the lifted-system representation of (2.1) can be written as

$$y_i(t) = \mathbf{P}u_i(t) + d(t) \quad (2.16)$$

$$\begin{bmatrix} y_i(m) \\ y_i(m+1) \\ \vdots \\ y_i(m+M-1) \end{bmatrix} = \begin{bmatrix} h_m & 0 & \cdots & 0 \\ h_{m+1} & h_m & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h_{m+M-1} & h_{m+M-2} & \cdots & h_m \end{bmatrix} \begin{bmatrix} u_i(0) \\ u_i(1) \\ \vdots \\ u_i(M-1) \end{bmatrix} + \begin{bmatrix} d(m) \\ d(m+1) \\ \vdots \\ d(m+M-1) \end{bmatrix} \tag{2.17}$$

Similarly, the system error in lifted form is written as

$$e_i(t) = r(t) - y_i(t) \tag{2.18}$$

$$\begin{bmatrix} e_i(m) \\ e_i(m+1) \\ \vdots \\ e_i(m+M-1) \end{bmatrix} = \begin{bmatrix} r(m) \\ r(m+1) \\ \vdots \\ r(m+M-1) \end{bmatrix} - \begin{bmatrix} y_i(m) \\ y_i(m+1) \\ \vdots \\ y_i(m+M-1) \end{bmatrix} \tag{2.19}$$

The equations (2.16) through (2.19) define the time-domain dynamics of the ILC system. In order to write the lifted version of the iteration-domain dynamics in (2.7), one should determine the type of functions to be used for ILC's filters, $L(q)$ and $Q(q)$. In the most general form, it is possible to select non-causal functions for these filters. This is allowed because ILC calculates the next iteration data by manipulating the previous iteration data saved in the memory. In other words, this feature provides an anticipation behaviour which can be expressed via a non-causal function (note that this feature does not exist in a feedback controller). The impulse response of a non-causal filter can be expressed by :

$$P(q) = \sum_{k=-M}^{M} {h_k}^{-k} = h_{-(M-1)}q^{M-1} + ... + h_{-2}q^2 + h_{-1}q^1 + h_0 + h_1q^{-1} + h_2q^{-2} + ... + h_{M-1}q^{-(M-1)} \tag{2.20}$$

If the Markov parameters for $L(q)$ and $Q(q)$ are denoted by L_k and Q_k, respectively,

$$L(q) = L_{-(M-1)}q^{M-1} + ... + L_{-2}q^2 + L_{-1}q^1 + L_0 + L_1q^{-1} + L_2q^{-2} + ... + L_{M-1}q^{-(M-1)} \tag{2.21}$$

$$Q(q) = Q_{-(M-1)}q^{M-1} + ... + Q_{-2}q^2 + Q_{-1}q^1 + Q_0 + Q_1q^{-1} + Q_2q^{-2} + ... + Q_{M-1}q^{-(M-1)} \tag{2.22}$$

Then, the iteration dynamics (2.7) in the lifted form is given as

$$u_{i+1}(t) = \mathbf{Q}(u_i(t) + \mathbf{L}e_i(t)) \tag{2.23}$$

$$\begin{bmatrix} u_{i+1}(0) \\ u_{i+1}(1) \\ \vdots \\ u_{i+1}(M-1) \end{bmatrix} = \begin{bmatrix} Q_0 & Q_{-1} & \cdots & Q_{-(M-1)} \\ Q_1 & Q_0 & \cdots & Q_{-(M-2)} \\ \vdots & \vdots & \ddots & \vdots \\ Q_{M-1} & Q_{M-2} & \cdots & Q_0 \end{bmatrix} \left(\begin{bmatrix} u_i(0) \\ u_i(1) \\ \vdots \\ u_i(M-1) \end{bmatrix} + \begin{bmatrix} L_0 & L_{-1} & \cdots & L_{-(M-1)} \\ L_1 & L_0 & \cdots & L_{-(M-2)} \\ \vdots & \vdots & \ddots & \vdots \\ L_{M-1} & L_{M-2} & \cdots & L_0 \end{bmatrix} \begin{bmatrix} e_i(m) \\ e_i(m+1) \\ \vdots \\ e_i(m+M-1) \end{bmatrix} \right) \tag{2.24}$$

It can be observed that **P**, **L** and **Q** matrices have identical diagonal entries which means they are in *Toeplitz form*. Furhermore, **L** and **Q** have elements both above and below their diagonals due to their non-causalities. In the causal case however, since their Markov parameters with $k < 0$ cancel out, they transform into lower triangular matrices such that

$$\mathbf{L} = \begin{bmatrix} L_0 & 0 & \cdots & 0 \\ L_1 & L_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ L_{M-1} & L_{M-2} & \cdots & L_0 \end{bmatrix} \tag{2.25}$$

$$\mathbf{Q} = \begin{bmatrix} Q_0 & 0 & \cdots & 0 \\ Q_1 & Q_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ Q_{M-1} & Q_{M-2} & \cdots & Q_0 \end{bmatrix} \tag{2.26}$$

As another important point, one should make sure that the diagonal entries of P, L and Q are nonzero. Otherwise, the asymptotic stability condition is not satisfied and the system diverges (see Theorem 1.1 in Section 2.2.1). If the plant produces a delay of m, the lifted matrix P should be created considering the Markov parameters starting from p_m which is the first nonzero parameter. In addition, the elements of the vectors $\mathbf{y}_i$, $\mathbf{d}$, $\mathbf{r}$ and $\mathbf{e}_i$ should also be shifted by m samples in order to adapt them to the plant's delay.

2.1.2 Frequency-domain ILC design

The frequency-domain representation is based on applying the Z-transform to discrete-time signals (sequences) in order to obtain their frequency responses (unit impulse response). The general form of the Z-transform is the bilateral or two-sided Z-transform which is given by

$$X(z) = Z\{x[n]\} = \sum_{n=-\infty}^{\infty} x[n] z^{-n} \tag{2.27}$$

where $z \in \mathbb{C}$ is a complex number, $n \in \mathbb{N}$ is an integer, $x[n]$ is a discrete-time signal (sequence), Z is the Z-transform operator and $X(z)$ is the Z-transformed sequence in the Z-domain. The bilateral Z-transform represents the noncausal case of a discrete signal

combining both the effects of causality (only-past-dependency) and anti-causality (only-future-dependency). However, the real systems are in practice causal such that in signal processing one mostly uses the unilateral or one-sided Z-transform which is written by considering $n \geq 0$ in (2.27) :

$$X(z) = Z\{x[n]\} = \sum_{n=m}^{\infty} x[n]z^{-n} \tag{2.28}$$

where m represents the delay as before (for no delay $m = 0$). The sequences and proper functions (2.1)-(2.12) can easily be obtained via replacing the shifting operator q with z. Although the purpose of these two notations is to represent the same notion of shifting, the difference is that q is a general notation which does not specify any further detail in terms of frequency while z represent a complex number which can be used to obtain frequency response of a discrete signal. Applying (2.28), the Z-transform of the system (2.1) is written as

$$Y_i(z) = P(z)U_i(z) + D(z) \tag{2.29}$$

and error of the system is

$$E_i(z) = R(z) - Y_i(z) \tag{2.30}$$

while the ILC update equation is

$$U_{i+1}(z) = Q(z)[U_i(z) + L(z)E_i(z)] \tag{2.31}$$

The next step is to use the Z-transforms for calculating the frequency responses. This is done by assigning the parameter z to a complex number as follows :

$$z = Ae^{j\phi} = A(\cos\phi + j\sin\phi) \tag{2.32}$$

where j is the imaginary unit, $A \in \mathbb{R}$ is the magnitude of z and $\phi \in [-\pi, \pi]$ is the complex argument denoting the phase in radians. If one considers $A = 1$, the frequency responses of the equations (2.29)-(2.31) will correspond to their unit impulse responses such that

$$Y_i(e^{j\phi}) = P(e^{j\phi})U_i(e^{j\phi}) + D(e^{j\phi}) \tag{2.33}$$

$$E_i(e^{j\phi}) = R(e^{j\phi}) - Y_i(e^{j\phi}) \tag{2.34}$$

$$U_{i+1}(e^{j\phi}) = Q(e^{j\phi})[U_i(e^{j\phi}) + L(e^{j\phi})E_i(e^{j\phi}) \tag{2.35}$$

The main idea is to find the proper frequency responses for $L(e^{j\phi})$ and $Q(e^{j\phi})$ such that $Y_i(e^{j\phi})$ (or $E_i(e^{j\phi})$) follows the desired reference $R(e^{j\phi})$. The tools of designing these responses are generally Nyquist and Bode diagrams and the design is determined by the frequency-domain ILC stability criterion given in Section 2.2.2.

2.1.3 Comparison of the representations

The first ILC designs presented in the literature were based on the the time-domain representation and the frequency-domain representation started to appear later on as an alternative design approach. Although both representations can be used to get desirable

result, there are some differences in each that are worth to be emphasizes. First of all, from the point of practical applications, ILC systems have to be finite since one cannot have an infinite number of samples in reality. Thus, the time-domain representation reflects better the reality than the frequency domain one which is rather an approximate representation assuming signals with infinite horizons. However, the accuracy of both representations begin to coincide as the number of samples (M) increases. This means that before choosing one of these representations, it would be clever to consider the number of samples needed for the application (note that the sampling time and total process time determine the length of M and thus the dimensions of the matrices and the overall computational heaviness). Second, both representations use different frameworks. The time-domain design uses the lifted-framework while the frequency domain design utilises the Z-transform. Third, the computational cost of both representations is quite distinct when it comes to especially large M. In time-domain the computational cost can be quite problematic since large M means heavier calculations due to larger matrices and vectors. On the other hand, this is never the case for frequency domain representation because it is always an approximation.

Table 2.1: Comparison of ILC representations

Features	Time-domain ILC	Frequency-domain ILC
Framework	Lifted-system	Z-transform
Sequence length	Finite	Infinite
Practical Accuracy	Constant	increases with larger M
Impulse response	Shifted Markov parameters	Shifted complex numbers
Proper functions mapping	To matrices	To transfer functions
Sequence mapping	To vectors	To frequency responses
Computational cost at large M	High	Low

Overall, it is reasonable to say that there is a sort of trade-off when selecting one of these representations. The efficiency of both is directly dependent on the application requirements. Finally, one can refer to Table 2.1 to have a clearer and a bit more detailed view of the comparison given above.

It is worth to note that a selection between these two representations will depend on the application requirements and can be a design constraint.

2.2 Stability and Convergence

Closed-loop stability is one of the most essential specifications for a control system. When a system is said to be input-output stable, it is meant that its outputs are bounded for the bounded inputs. The strength of the stability feature is usually analysed in three categories : absolute stability (convergent), conditional stability (partially convergent) and marginal stability (neither convergent nor divergent). The absolutely stability is the strongest stability condition for a system. It means that whether open-loop or closed-loop,

the system's poles are all on the left half of the complex s-plane. In other words, the system is stable all over the range of its parameter values. The conditional stability is a bit less restricted stability condition which means that the system is stable over a specific range of system parameters. It can be seen as having stable dynamics and unstable dynamics at the same. Finally, the marginal stability is the limit of stability where system does not absorb the inputs and disturbances, instead it just produces a constant amplitude signal (nonzero poles over the imaginary axis in linear systems).

For an iterative system, the same logic of stability can be applied. The only difference is that the stability notion is now assessed considering the evaluation over the iteration axis. In short, if the system error tends to zero or to a bounded value as the iterations continue, the iterative system is considered to be stable. Before analysing the stability of ILC, it is useful to mention some preliminary assumption on the system to be controlled by ILC. In [85] the authors state that these assumptions that were originally provided by Arimoto can be found in [99] and then they give a reformulation of these assumptions which also suits the framework utilised in this thesis. Thus, the following assumptions are taken from [85].

(A1) The signals are finite in time axis such that each iteration finishes at the same duration $t = T_f \in \mathbb{R}^+$.
(A2) The reference $r(t)$ is given a priori over the interval $t \in [0, T_f]$.
(A3) The initial condition of the system is identical at each iteration, i.e. $x_i(0) = x_0 \in \mathbb{R}^n$, $\forall i \in [0, N]$.
(A4) The dynamics of the system is kept invariant throughout all iterations.
(A5) Every system output $y_i(t)$ is measurable such that the information of the tracking error $e_i(t) = r(t) - y_i(t)$ can be achieved each time $u_{i+1}(t)$ is computed.
(A6) For the given trajectory $r(t)$ having a piece-wise continuous derivative, there exists a unique input trajectory $u^*(t)$ that will match the output to the reference, i.e. $y^*(t) = r(t)$ (*Author's remark : for nonlinear systems Local (or Global) Lipschitz continuity is assumed and $u^*(t)$ is searched around some local conditions (see Section 2.4)*).

Practically, obtaining an identical initial condition can be quite difficult and therefore the assumption (A3) is rather restrictive. Thus, the following modification for (A3) would make ILC more feasible to real applications :

(A3a) The initial condition resetting has a small bounded error, i.e. for some constant $\epsilon \in \mathbb{R}$, $|x_i(0) - x_0| < \epsilon$, $\forall i \in [0, N]$.

Assuming that the assumptions (A1)-(A6) hold, let us also introduce the following useful definitions of stability (cf. [85]) :

Definition 2.1 (ϵ-convergence) : *A system with ILC is said to be ϵ-convergent if*

$$\lim_{i \to \infty} \sup ||u^*(t) - u_i(t)|| < \epsilon$$

In other words, the difference between the ILC input solution $u^*(t)$ and the ILC input at all other iterations $u_i(t)$ is always less than a finite value ϵ. If this property holds, the system with ILC is called stable. However, the definition of ϵ-convergence does not mean any convergence to zero error. The ILC system error at $i \to \infty$ can be higher then the

case where system does not use ILC.

Definition 2.2 (Asymptotic Stability) : *A system with ILC is said to be asymptotically stable if*

$$\lim_{i\to\infty} \sup ||u^*(t) - u_i(t)|| = 0$$

The asymptotically stable means that the ILC input will reach the input solution yielding the perfect tracking or rejection as $i \to \infty$. However, the satisfaction of this condition does not imply that that system error will reduce at each iteration.

Definition 2.3 (Exponential Stability) : *A system with ILC is said to be exponentially stable if*

$$\exists \alpha, \beta > 0 \mid \forall i > 0, ||u^*(t) - u_i(t)|| \leq \alpha ||u^*(t) - u_0(t)|| e^{-\beta i}$$

This simply implies that at each iteration the ILC input approaches more to the input solution or alternatively that there is a monotone decrease of system error at each iteration (i.e. $e_{i+1}(t) < e_i(t)$).

Definition 2.4 (Global Stability) : *The definitions given above become global when the system with ILC is stable for all possible initial conditions (x_0) and initial inputs ($u_0(t)$). Although it might be difficult to achieve, this property can be sought after in cases where ILC is applied on a nonlinear system.*

Definition 5 (Bounded-Input-Bounded-Output (BIBO) Stability) : *A linear iterative system is said to be BIBO-stable if a bounded input ($u_i(t) < \infty$) produces a bounded output ($y_i(t) < \infty$) for all i.*

Thus, if the iterative system has this property, it means that the ILC algorithm provides a stable performance.

Definition 6 (Uniform Exponential Stability) : *A linear iterative system has uniform stability if for the free response ($u_i(t) = 0$) of the system with any initial output ($y_0(t)$) and initial iteration (i_0) satisfies*

$$||y_i(t)|| \leq \gamma ||y_0(t)||, \quad i \geq k_0$$

where γ stands for a positive constant. The uniform stability is extended to the uniform exponential stability if for $\lambda \in (0, 1]$ the following is also true

$$||y_i(t)|| \leq \gamma \lambda^{i-i_0} ||y_0(t)||, \quad i \geq k_0$$

At this point the stability analysis can be shown by introducing some well-known theorems from the linear systems theory (see [85] where these theorems are represented based on the works of [93], [59], [92], [91], [3] and [79]. Since it is possible to use both the time-domain and the frequency domain representations for ILC design, the stability analysis hereafter is presented for each of these representations separately.

2.2.1 Stability analysis of time-domain ILC

In order to better demonstrate the stability and convergence of a time-domain ILC system, the iteration dynamics equation (2.23) can be put into the error dynamics form. From (2.18) and (2.16), the system error and the output at the $(i+1)^{th}$ iteration are

$$e_{i+1}(t) = r(t) - y_{i+1}(t) \tag{2.36}$$
$$y_{i+1}(t) = \mathbf{P}u_{i+1}(t) + d(t) \tag{2.37}$$

respectively. Substituting (2.37) into (2.36) and applying (2.23) leads to

$$e_{i+1}(t) = r(t) - \mathbf{P}(\mathbf{Q}u_i(t) + \mathbf{QL}e_i(t)) - d(t)$$

Then using (2.18) as $r(t) = e_i(t) + y_i(t)$ and replacing $\mathbf{y}_i$ with (2.16), the error dynamics is found as follows :

$$\begin{aligned} e_{i+1}(t) &= r(t) - \mathbf{P}(\mathbf{Q}u_i(t) + \mathbf{QL}e_i(t)) - d(t) \\ &= e_i(t) + y_i(t) - \mathbf{PQ}u_i(t) - \mathbf{PQL}e_i(t) - d(t) \\ &= (\mathbf{I} - \mathbf{PQL})e_i(t) - \mathbf{PQ}u_i(t) + \mathbf{P}u_i(t) + d(t) - d(t) \\ &= (\mathbf{I} - \mathbf{PQL})e_i(t) - \mathbf{P}(\mathbf{I} - \mathbf{Q})u_i(t) \end{aligned} \tag{2.38}$$

It is straightforward to observe in (2.38) that the error convergence of the iterative system is determined mainly by the matrix $(\mathbf{I} - \mathbf{PQL})$ and its eigenvalues. In addition, the value of $\mathbf{P}(\mathbf{I} - \mathbf{Q})\mathbf{u}_i$ has an increasing effect on the error at the next iteration. It is simply the difference between the system output when the input has no filter and the system output when the input is Q-filtered. In order to diminish this effect and achieve a faster convergence, the traditional choice is $\mathbf{Q} = I$. As it will be explained later, $\mathbf{Q}$ is in general a low pass filter applied for increasing the ILC robustness which causes a trade-off for the convergence speed.

The stability of (2.38) can be analysed utilising two measures for matrices which are the spectral radius (ρ) and the maximum singular value ($\overline{\sigma}$). These measures are well-known results from linear systems theory and their definitions are as given below.

Definition 2.7 *(Spectral Radius) For a matrix $\boldsymbol{F} \in \mathbb{R}^{n\times n}$, the spectral radius is the maximum of its eigenvalues, i.e.*

$$\rho(\mathbf{F}) = \max_{k=1,\ldots,n} |\lambda_k(\mathbf{F})|$$

where $\lambda_k(\boldsymbol{F})$ is k^{th} eigenvalue of $\boldsymbol{F}$.

Definition 2.8 (Maximum Singular Value) *For a matrix $\boldsymbol{F}$, the maximum singular value is the maximum gain it can apply on a vector while mapping it, i.e.*

$$(a) \quad \overline{\sigma}(\mathbf{F}) = \sqrt{\rho(\mathbf{F}^\top \mathbf{F})}$$
$$(b) \quad ||\mathbf{F}v|| \leq \overline{\sigma}(\mathbf{F})||v||$$

where v is any column vector.

For the simplicity, let us now assume that $\mathbf{Q} = I$. Then, the error dynamics (2.38) becomes

$$e_{i+1}(t) = (\mathbf{I} - \mathbf{PQL})e_i(t) \tag{2.39}$$

The ILC system with (2.39) is said to be asymptotically stable if Definition 2.7 holds for the matrix $(\mathbf{I} - \mathbf{PQL})$. If the state-space system is minimal (all states are observable and controllable), it can also be said that the ILC system is BIBO-stable according to Definition 2.7. Thus, the following theorem gives a necessary and sufficient condition for both of these stability types (see [85] and [22]) :

Theorem 2.1 (Asymptotic Stability or BIBO-stability) : *The ILC system is asymptotically stable or under the condition of minimality in case of a state-space system, is BIBO-stable if*

$$\rho(\mathbf{I} - \mathbf{PQL}) < 1 \tag{2.40}$$

Alternatively, the stability of ILC can be determined by applying the same logic to the the update equation (2.23). Substituting (2.18) and (2.16) into (2.23),

$$u_{i+1}(t) = \mathbf{Q}(\mathbf{I} - \mathbf{LP})u_i(t) + \mathbf{QL}(r(t) - d(t)) \tag{2.41}$$

Thus, under the same conditions mentioned above the same asymptotic and BIBO-stability are achieved if

$$\rho(\mathbf{Q}(\mathbf{I} - \mathbf{LP})) < 1 \tag{2.42}$$

Although necessary, the satisfaction of the conditions (2.40) or (2.42) do not imply any monotonic convergence for the ILC system (see Definition 2.3). This can be quite problematic for some applications since ILC can still be asymptotically stable while having huge transients, so called *learning transients*. According to Definition 2.3, the monotonic convergence means that under a norm $||.||$ (2.39) satisfies

$$||e_\infty(t) - e_{i+1}(t)|| = \gamma||e_\infty(t) - e_i(t)|| \tag{2.43}$$

where $\gamma \in [0, 1)$ is the rate of convergence and $e_\infty(t)$ is the asymptotic error that is found by putting $i = \infty$ in (2.39) and considering that at infinity $e_{i+1}(t) = e_i(t) = e_\infty(t)$ and $u_i(t) = u_\infty(t)$ (i.e. the final input),

$$\begin{aligned} e_\infty(t) &= (\mathbf{I} - \mathbf{PQL})e\infty(t) - \mathbf{P}(\mathbf{I} - \mathbf{Q})u_\infty(t) \\ &= (\mathbf{PQL})^{-1}\mathbf{P}(\mathbf{I} - \mathbf{Q})u_\infty(t) \\ &= (\mathbf{QL})^{-1}(\mathbf{I} - \mathbf{Q})u_\infty(t) \end{aligned} \tag{2.44}$$

This relation can also be verified by manipulating (2.23) as

$$\begin{aligned} u_\infty(t) &= \mathbf{Q}u_\infty(t) + \mathbf{QL}e_\infty(t) \\ (\mathbf{I} - \mathbf{Q})u_\infty(t) &= \mathbf{QL}e_\infty(t) \\ (\mathbf{QL})^{-1}(\mathbf{I} - \mathbf{Q})u_\infty(t) &= e_\infty(t) \end{aligned} \tag{2.45}$$

Note that the asymptotic error becomes zero for $\mathbf{Q} = \mathbf{I}$ which also transforms (2.43) into

$$||e_{i+1}(t)|| = \gamma||e_i(t)|| \tag{2.46}$$

In order to make the ILC monotonically convergent, one should obtain the description for γ in (2.43) and then apply the Definition 2.8 on it. Using (2.38) and (2.44), one can calculate the error differences in (2.43) as

$$\begin{aligned} e_\infty(t) - e_{i+1}(t) &= (\mathbf{QL})^{-1}(\mathbf{I}-\mathbf{Q})u_\infty(t) - (\mathbf{I}-\mathbf{PQL})e_i(t) - \mathbf{P}(\mathbf{I}-\mathbf{Q})u_i(t) \\ &= (\mathbf{I}-\mathbf{PQL})(e_\infty(t) - e_i(t)) - (\mathbf{I}-\mathbf{PQL})e_\infty(t) \\ &\quad + (\mathbf{QL})^{-1}(\mathbf{I}-\mathbf{Q})u_\infty(t) - \mathbf{P}(\mathbf{I}-\mathbf{Q})u_i(t) \\ &= (\mathbf{I}-\mathbf{PQL})(e_\infty(t) - e_i(t)) + (\mathbf{I}-\mathbf{I}+\mathbf{PQL})(\mathbf{QL})^{-1}(\mathbf{I}-\mathbf{Q})u_\infty(t) \\ &\quad - \mathbf{P}(\mathbf{I}-\mathbf{Q})u_i(t) \\ &= (\mathbf{I}-\mathbf{PQL})(e_\infty(t) - e_i(t)) + \mathbf{P}(\mathbf{I}-\mathbf{Q})u_\infty(t) - \mathbf{P}(\mathbf{I}-\mathbf{Q})u_i(t) \\ &= (\mathbf{I}-\mathbf{PQL})(e_\infty(t) - e_i(t)) + \mathbf{P}(\mathbf{I}-\mathbf{Q})(u_\infty(t) - u_i(t)) \end{aligned} \tag{2.47}$$

For the sake of simplicity, it can be assumed that $\mathbf{Q} = \mathbf{I}$ and letting $||.||$ be the Euclidian norm denoted as $||.||_2$,

$$\begin{aligned} e_\infty(t) - e_{i+1}(t) &= (\mathbf{I}-\mathbf{PQL})(e_\infty(t) - e_i(t)) \\ ||e_\infty(t) - e_{i+1}(t)||_2 &\leq ||\mathbf{I}-\mathbf{PQL}||_2||e_\infty(t) - e_i(t)||_2 \end{aligned} \tag{2.48}$$

From this expression, it is clear to see the definition of γ as

$$\gamma := ||\mathbf{I}-\mathbf{PQL}||_2 \tag{2.49}$$

Thus, according to Definition 2.8, the monotonic convergence is reached if the following theorem holds :

Theorem 2.2 (Monotonic Convergence) The ILC system is monotonically convergent if the error at each new iteration is less than the previous iteration's error, i.e. to ensure

$$\begin{aligned} \gamma &= ||\mathbf{I}-\mathbf{PQL}||_2 < 1 \\ &= \overline{\sigma}(\mathbf{I}-\mathbf{PQL}) < 1 \end{aligned} \tag{2.50}$$

2.2.2 Stability analysis of frequency-domain ILC

In order to show the stability conditions of frequency domain ILC, let us write (2.39) and (2.41) in the Z-domain, respectively.

$$E_{i+1}(z) = [1 - P(z)Q(z)L(z)]E_i(z) - P(z)[1-Q(z)]U_i(z) \tag{2.51}$$

$$U_{i+1}(z) = Q(z)[1 - L(z)P(z)]U_i(z) + Q(z)L(z)[R(z) - D(z)] \tag{2.52}$$

The asymptotic stability is achieved by the frequency domain ILC system if the following theorem holds :

Theorem 2.3 (Frequency-domain Asymptotic Stability) An ILC system in the frequency-domain ($M = \infty$) is asymptotically stable if $1 - P(z)Q(z)L(z)$ or $Q(z)[1 - L(z)P(z)]$ is a contraction mapping [21], i.e.

$$||1 - P(z)Q(z)L(z)||_\infty = \sup_{\theta\in[-\pi,\pi]} ||1 - P(e^{j\theta})Q(e^{j\theta})L(e^{j\theta})|| < 1 \tag{2.53}$$

$$||Q(z)[1 - L(z)P(z)||_\infty = \sup_{\theta\in[-\pi,\pi]} ||Q(e^{j\theta})[1 - L(e^{j\theta})P(e^{j\theta})|| < 1 \tag{2.54}$$

Since the frequency domain representation requires an assumption of infinite samples ($M = \infty$), the conditions (2.53) and (2.54) are results for an approximated system and thus they are only sufficient and not necessary for the asymptotic stability. They are more conservative than (2.40) and (2.42) [21]. Furthermore, the monotonic convergence in frequency domain is written as

$$||E_\infty(z) - E_{i+1}(z)||_\infty = \gamma||E_\infty(z) - E_i(z)||_\infty \tag{2.55}$$

where γ is the convergence rate and the asymptotic error is given by

$$E_\infty(z) = \frac{1 - Q(z)}{Q(z)L(z)}U_\infty(z) \tag{2.56}$$

Note that the asymptotic error becomes zero for $Q(z) = 1$. Following the same procedure given in previous section and choosing again $Q(z) = 1$ for simplification, (2.55) becomes,

$$||E_\infty(z) - E_{i+1}(z)||_\infty = ||1 - P(z)Q(z)L(z)||_\infty||E_\infty(z) - E_i(z)||_\infty \tag{2.57}$$

which indicates

$$\gamma := ||1 - P(z)Q(z)L(z)||_\infty \tag{2.58}$$

Hence, the monotonic convergence is achieved according to the following theorem :

Theorem 2.4 (Frequency-domain Monotonic Convergence) An ILC system in frequency-domain is monotonic convergent if

$$\gamma = ||1 - P(z)Q(z)L(z)||_\infty = \sup_{\theta\in[-\pi,\pi]} ||1 - P(e^{j\theta})Q(e^{j\theta})L(e^{j\theta})|| < 1 \tag{2.59}$$

This result simply proves that the asymptotic stability and monotonic convergence conditions are identical for frequency-domain ILC. Having seen the stability properties of ILC designs, some other useful properties of ILC such as order of ILC and forgetting factor can be introduced.

2.2.3 Order of ILC and forgetting factor

The ILC update law given in (2.7) is a first-order ILC since it utilises only the data from one previous iteration (i.e. $\mathbf{e}_i(t)$ and $\mathbf{u}_i(t)$). However, this is not the only option for ILC system design. There also exists various extensions to so called *higher-order ILC*

(HO-ILC) algorithms in the literature such as [14, 30, 67, 108, 109]. It is proposed in [109] that using the information from the previous $N > 1$ iterations in the calculation of the current iteration control input may provide faster error convergence speed compared to the first-order ILC. The non-causality of the HO-ILC is feasible due to the fact that all previous iteration data is kept in the memory. The HO-ILC update equation can slightly vary depending on designer preferences. For example, a common form of HO-ILC update law [108] in the time-domain representation is :

$$\mathbf{u}_{i+1}(t) = \mathbf{u}_i(t) + \sum_{k=1}^{N} \mathbf{L}_k(t)\mathbf{e}_{l+1}(t) \tag{2.60}$$

where N is the order of HO-ILC, $\mathbf{L}_k$ is the learning gain matrix for each iteration $l = i-k+1$ and $\mathbf{e}_{l+1} = \mathbf{r} - \mathbf{y}_{l+1}$. Some other versions of HO-ILC (e.g. [14]) include several inputs form previous iterations in addition as well :

$$\mathbf{u}_{i+1}(t) = \sum_{k=1}^{N} \mathbf{P}_k(t)\mathbf{u}_{l+1}(t) + \sum_{k=1}^{N} \mathbf{L}_k(t)\mathbf{e}_{l+1}(t) \tag{2.61}$$

where P_k is the gain matrix applied on input k. Furthermore, [108] emphasizes the importance of using a forgetting factor in HO-ILC that can improve the robustness of ILC under the factors such as disturbance, uncertainty, fluctuations of system dynamics and initialisation error. This type of HO-ILC can be given as

$$\mathbf{u}_{i+1}(t) = \beta(i)\mathbf{u}_0(t) + (1 - \beta(i))\mathbf{u}_i(t) + \sum_{k=1}^{N} \mathbf{L}_k(t)\mathbf{e}_{l+1}(t) \tag{2.62}$$

where $\beta(i) = H^{-i}$ is the time-varying forgetting factor and $\mathbf{u}_0(t)$ is the initial input. It is suggested that $0 \leq \beta(i) < 1$ and H is constant such that as $i \to \infty$, $\beta(i) \to 0$ [108]. This simply indicates that as the iterations continue HO-ILC begins to remember less the initial input at the very beginning $\mathbf{u}_0(t)$ and focuses more on the previous iteration input $\mathbf{u}_i(t)$.

Let us now show the possible ways of integrating an ILC to a given system.

2.2.4 ILC integration to an existing system

When it comes to connecting ILC to the system, one may find two options [21] :

1. Adding ILC output to the system's reference signal (serial connection),
2. Adding ILC output to the closed loop controller's output inside the system (parallel connection).

These two different options for connection can be reviewed in Figure 2.2 and Figure 2.3, respectively. It should be noted that a system with serial ILC connection calculates the output slightly different than its version with parallel ILC connection. The reason to that is obviously the change in block calculations due to the different locations ILC signal enters the system. Here, $\mathbf{C}$ and $\mathbf{P}$ denote the closed loop controller block and the plant,

respectively (see Figure 2.2 and Figure 2.3).

$$\mathbf{y}_i = [(1 + \mathbf{PC})^{-1}\mathbf{PC}]\ \mathbf{u}_i + [(1 + \mathbf{PC})^{-1}\mathbf{PC}]\ \mathbf{r} \tag{2.63}$$

$$\mathbf{y}_i = [(1 + \mathbf{PC})^{-1}\mathbf{P}]\ \mathbf{u}_i + [(1 + \mathbf{PC})^{-1}\mathbf{PC}]\ \mathbf{r} \tag{2.64}$$

Equation (2.63) gives the system output with serial ILC connection and (2.64) is the output with parallel ILC connection. The difference comes from an extra usage of term $\mathbf{C}$ for serial ILC connection. Note that the serial ILC does not need to know $\mathbf{P}$ and $\mathbf{C}$ blocks explicitly as $(1 + \mathbf{PC})^{-1}\mathbf{PC}$ can be used as a "black box" which can be obtained through identification using its input-ouput data directly. On the other hand, the parallel ILC requires a knowledge of $\mathbf{P}$ and $\mathbf{C}$. Thus, one can conclude that adding an ILC to the system is much easier with the serial connection since in this case one does not have to know the internal controller parameters.

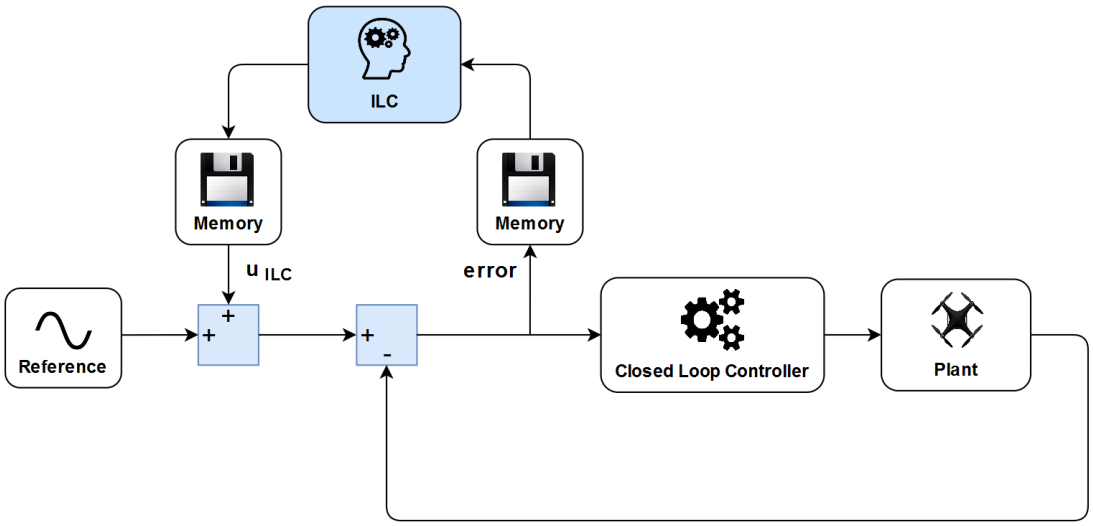

FIGURE 2.2: Serial ILC connection

Before jumping into the existing ILC methods, it can be useful to review some common ILC classification that has appeared in the literature so far. The following classifications are based on the author's observations during the thesis work.

2.3 Common classifications of ILC methods

Apart from the ILC classification given so far, it is possible to notice some other common ILC classifications in the literature as well. These classifications usually appear with following properties :

— The time-domain in which ILC is defined,
— The ILC update method that is used,
— The order of the ILC update law,
— The time-variance of ILC or system parameters.

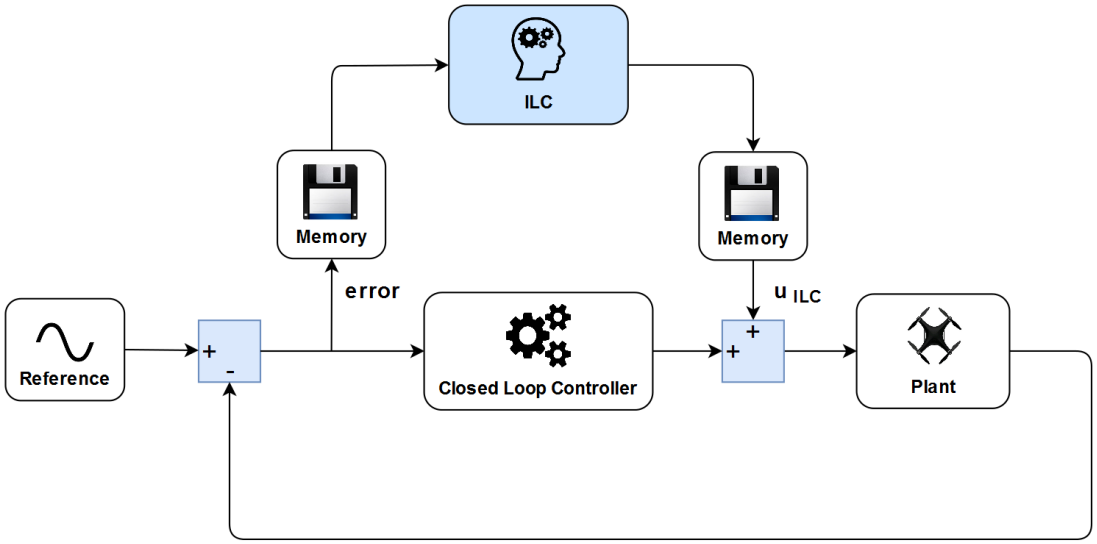

FIGURE 2.3: Parallel ILC connection

The first classification is about the ILC's time-domain, namely continuous-time or discrete-time depending on the application. Although most of the ILC work in literature has been done in discrete-time domain, it is still possible to come across some analyses in continuous-time domain as well (see e.g. [31]). This is a sound preference since all real-life applications are performed using computers that work in discrete-time.

The second classification is made considering which method is used for updating the output of ILC. One can write the ILC update equation using plenty of different update methods such as P-type, D-type, PD-type, fractional-order, inversion-based and optimization-based (e.g. norm optimal ILC, predictive norm-optimal ILC and parameter optimal ILC). It is also possible to see other names for optimization-based ILCs which are given according to the optimization method used (e.g. ILC with steepest-descend, ILC with gradient method, ILC with Newton-Raphson etc.). One can refer to [48] for the above mentioned methods.

The third classification is related to how much of information is used from previous trials in order to calculate the ILC output. One may use only the previous trial's information when updating ILC which in this case makes the update law of first order. On the other hand, it is also possible to use the information from more than one previous trials and this type of ILCs are called high-order ILCs.

The fourth classification comes from the system's behaviour with time. If the system is time-variant then the plant model should be identified before each ILC iteration. In this case the filters used in ILC will be different during each iteration and they need to be recomputed. Otherwise, in a time-invariant system the ILC filters stay the same during each iteration.

Next, a quick overview of the existing ILC methods is given.

2.4 Overview of Existing ILC approaches

In [114], the ILC design approaches are grouped in four categories in terms of system nonlinearities :

1. Linear ILC for linear systems,
2. Nonlinear ILC for linear systems,
3. Linear ILC for global Lipshitz continuous (GLC) nonlinear systems,
4. Nonlinear ILC for local Lipshitz continuous (LLC) nonlinear systems.

However, this categorisation is too detailed for this thesis work and it can be shortened into two categories : the *linear ILC methods* and the *nonlinear ILC methods*. This simplification can be done since both the linear ILC and nonlinear ILC methods are applicable to linear and nonlinear systems. It can be observed that the focus of research in both category has been changing over the years. To begin, in linear ILC, the theoretical problems have mostly been understood such that the current focus is mainly on improving the performance of ILC in practice. The main interest is towards creating ILC algorithms that are nontraditional which deal with issues such as no-resetting possibility, robustness to non-zero initial error [66, 88], iteration-variant trial lengths [98], robustness to non-repetitiveness [76] (i.e. non-learning of the non-repetitive data such as iteration-varying uncertainties or disturbances), time-varying Q-filter [102] etc.

The evolution of the nonlinear ILC on linear systems had started with an aim of achieving faster convergence speeds and better transient performances than the linear ILC. Recently, the research focus of nonlinear ILC branched into *LLC* systems where Lyapunov approach and composite energy function are the fundamental design tools (ILC design cannot be done using contraction mapping since nonexistence of finite escape time phenomenon is not globally applicable to LLC systems) [114]. One of the main drawbacks of nonlinear ILC has been the need of more prior knowledge (e.g. see [117]) about the system dynamics (in case of LLC systems, state dynamics has to be known) which can be restrictive for many practical applications and thus remains open for further research.

The nonlinear ILC is beyond the scope of this thesis work such that the following sections comprise only the linear ILC methods.

2.4.1 Linear ILC for linear systems

This section briefly introduces some popular ILC methods on the literature. The most traditional ILC algorithms, i.e. proportional-type (P-type) ILC, derivative-type (D-type) ILC and and their combination (PD-type ILC), are introduced first. Next, the explanations are extended to the *fractional-order ILC (FO-ILC)*, the *model inversion-based ILCs* and finally the *optimisation-based ILCs*. In the end of the section, the core ILC method of the thesis, that is the *norm-optimal ILC (NO-ILC)* is given a distinct place for more detailed look.

2.4.1.1 P-type, D-type and PD-type ILCs

The simplest ILC methods that can be found are the P-type ILC and the D-type ILC (see section 2 of [107] for various references these type of ILCs and their variations). In P-type ILC, the update is done by utilising the system error at the i^{th} iteration while in D-Type ILC the update is achieved by using the derivative of the system error at the i^{th} iteration. Thus, the update equations of P-Type and D-type ILCs can be written as

$$u_{i+1}(t) = Q(q)[u_i(t) + L_p(q)e_i(t)] \tag{2.65}$$
$$u_{i+1}(t) = Q(q)[u_i(t) + L_d(q)\dot{e}_i(t)] \tag{2.66}$$

respectively. Here, $L(q)$ and $Q(q)$ are user defined functions, namely a learning filter and a low pass filter for robustness, respectively. If ILC is designed in time domain, $L(q)$ and $Q(q)$ become lifted matrices $\mathbf{L}$ and $\mathbf{Q}$ whereas if ILC is designed in frequency domain they transform into transfer functions $L_p(z)$ and $Q(z)$ in Z-domain (see Section 2.1). For the simplest case, the learning filters in (2.65) and (2.66) can be selected as a proportional gain (i.e. $L = \Gamma_p$ with $\Gamma_p \in \mathbb{R}^+$) or a single shift operator (i.e. $L_p = q$ or in case of frequency domain design $L = z$).

Although simple looking and shown to be effective theoretically, P-type and D-type ILCs have significant drawbacks in practice, especially in case of uncertainties, disturbances and noises. A D-type ILC utilises the highest order derivative signals of a dynamics system which are either rather noisy or not measurable (most robots only have position sensors on their joints such that the velocity and acceleration information has to be acquired by numerical differentiation which can cause drastic noises) [107]. On the other hand, P-type ILC does not need any derivative information. However, it does not contain the trend of the error from previous iterations (i.e. if $e_i(t) = 0$ the learning stops even though $\dot{e}_i(t) \neq 0$) and initial state ($x(0) \neq 0 \ \forall i \geq 0$) or initial output errors ($e_i(0) \neq 0 \ \forall i \geq 0$) can cause divergence of the ILC signal (i.e. $u_i(0) \to \infty$ as $i \to \infty$) [107].

The PD-type method is simply combining the both P and D-types [68]. The ILC update equation is in this case is

$$u_{i+1}(t) = Q(q)[u_i(t) + L_p(q)e_i(t) + L_d(q)e_i(t)] \tag{2.67}$$

where $L_p(q)$ and $L_d(q)$ are learning functions for proportional and derivative parts, respectively. For the simplest case, one can select them as $L_p(q) = \Gamma_p \in \mathbb{R}^+$ and $L_d(q) = \Gamma_d \in \mathbb{R}^+$ or similarly as shift operators as explained above. The use of shift operators adds a non-causality to ILC. For example, if $L_d(q) = q$ in (2.67), ILC actually uses the future error from the saved data (iteration i), i.e. $e_i(t+1)$, in order to compute the current control signal $u_{i+1}(t)$. Also, a shifting operator can be used to avoid the system delay that appears when the system's relative degree $m > 0$ (for example, it means that if $m = 1$, the first system error can be obtained at $e_i(t+1)$ such that one can use $L(q) = q$ for getting correct results from ILC update).

Even though it is possible to find some explanations for the PID-type ILC (e.g. in [1, 29, 78]), the addition of an integrator does not have a considerable effect on update since ILC itself acts sort of as an integrator [68]. Yet, when the integral term is added, the

update equation becomes more general [78] and it can be given as,

$$u_{i+1}(t) = Q(q)[u_i(t) + L_p(q)e_i(t) + L_d(q)\dot{e}_i(t) + L_i(q)\int e_i(t)]. \tag{2.68}$$

where $L_p(q)$, $L_d(q)$ and $L_i(q)$ are learning filters for proportional, derivative and integral parts, respectively.

The above-mentioned update techniques have been quite popular due to their low dependency on system knowledge. However, they have some common disadvantages such as troublesome tuning for the learning filter and reset requirement for the the learning process each time the parameters are changed [101]. Also, as mentioned before, their performance deteriorates quickly with uncertainties, disturbances and noise.

Note : *Equations* (2.65)-(2.68) *asume that one does not do numerical differentiation and that the derivative of the error is available. In case where only the output error available the derivative parts of D-type, PD-type and PID-type ILCs are obtained by using a differentiator in their learning filters. For example, in frequency domain ILC design the time difference (derivative) is obtained by using* $L_d(z) = (1 - z^{-1})$ *and the time accumulation (integral) is found by applying* $L_i(z) = (1 - z^{-1})^{-1}$.

2.4.1.2 Fractional-order ILC (FO-ILC)

In addition to P, D, PD and PID-type methods there is another similar method which considers a fractional order derivative. In other words, the degree of the time-shifting operator is between 0 and 1 (here it is assumed that numerical derivation is applied). It can be understood as being a special case between P-type ($L_p(z) = 1 - z^0$) and D-type ($L_p(z) = 1 - z^1$) updates. The learning filter and update equation are written as,

$$\begin{aligned} L(z) &= \Gamma z^{\alpha} \\ u_{i+1}(t) &= Q(z)[u_i(t) + \Gamma(1 - z^{\alpha})e_i(t)]. \end{aligned} \tag{2.69}$$

where $\Gamma > 0$ and $0 < \alpha < 1$. The approach using a fractional order derivator is not as straightforward as the D-type approach which was only a time difference applied on the error data. In [69], they show that FO-ILC gives better results than D-type ILC in terms of convergence rate and it provides a more efficient tuning as well. However, it is still quite limiting fact that (2.69) does a numerical derivation on data which can cause divergence of the $u_{i+1}(t)$ easily.

2.4.1.3 Model Inversion-based ILC

The basic idea behind the model inversion technique is to use the inverse of the system as a filter in order to produce the input that will give the desired output. Therefore, it generally requires a rather accurate prior knowledge of the system. Once the inverse filter is obtained, the desired output is given as input to the filter which then creates the *a priori* needed system input. Consider the following system, $P(q)$, which has an input $u(t)$

that yields an output $y(t)$:

$$y(t) = P(q)u(t). \tag{2.70}$$

Then, assume a desired output $r(t)$ such that,

$$y(t) = r(t). \tag{2.71}$$

Theoretically, filtering $r(t)$ through the inverse of the system should provide the input $u(t)$ that will produce the desired output,

$$u(t) = P(q)^{-1}r(t). \tag{2.72}$$

Thus, sending the $u(t)$ signal to the actual system yields $r(t)$ as,

$$\begin{aligned} y(t) &= P(q)P(q)^{-1}r(t), \\ y(t) &= r(t). \end{aligned} \tag{2.73}$$

The ILC performance from the perspective of convergence rate and converged error depends on how well the learning filter is designed and the ideal learning filter is the one that corresponds to the inverse of the system [101]. Yet, the inversion of the system transfer function may cause some problems when discrete systems are considered. It is a common result that discretisation of continuous systems may produce unstable or lightly damped zeros in the system transfer function [42]. Accordingly, it is not possible to directly use the inverse of such system for controlling purposes since the inverses will be unstable. The systems with unstable zeros are referred to as non-minimum phase (NMP) systems. Obtaining an NMP system can simply be due to using fast sampling rates on systems with high relative degrees or noncollocated sensors and actuators [23]. The method utilized for discretising has also an important effect on the number of unstable zeros obtained when discretising. For some systems, it can be observed that the 'Zero Order Hold method' can produce more unstable zeros than 'Tustin method' for the same sampling time. There are several methods used to solve the problem of finding the inverse of NMP systems : nonmimimum phase zeros ignore (NPZ-ignore), zero-phase-error tracking controller (ZPETC), zero-magnitude-error tracking controller (ZMETC), non-causal series expansion, exact unstable inverse and zeroth order series. All of these methods aims at approximating the stable inverse of the NMP system. Butterworth et al. [23] states that NPZ-ignore, ZPETC, ZMETC and zeroth order series methods are rather effective and easy to design and implement whereas high order non-causal series expansion and exact unstable inverse methods add more complexity on control implementation. It can be seen that ILCs using one of the above-mentioned methods are named by some authors as *inversion-based ILC (IIC)* or *enhanced inversion-based ILC (EIIC)*. It is pointed out in [64] that convergence rate of IIC and EIIC methods are dependent on the accuracy of dynamics model of the system which is inclined to errors as well as it is time-consuming to obtain. To address these issues, [64] proposed a *model-less inversion-based ILC* method to eliminate the need for dynamics model when further improving the output tracking performance. [101] developed a data-based dynamic inversion algorithm to obtain a learning filter which reduces the unmodeled dynamics while successfully approximating NMP systems. Apart from these, in [42], they proposed an approach for using the finite impulse

response of the inverse transfer function to build learning filters that will avoid inversion of unstable or lightly damped zeros.

2.4.1.4 Optimisation-based ILC

Due to its effective and promisive nature, it is possible to find a significant number of publications related to optimisation-based ILC. The method is grounded on solving a convex optimisation problem which is defined by an objective function to be minimised under some constraint equations. A sufficient understanding on the application of optimisation methods on the ILC algorithm can be gained through [4, 28, 46, 47, 103]. Moreover, the PhD theses [84] and [48] are quite useful for having a deeper insight on the mathematical theory behind optimization-based ILC as well as reviewing detailed derivations for convergence conditions in general.

The objective of the optimisation algorithm can be seen as making the tracking error of the system as small as possible. This corresponds to minimising the norm of the tracking error and the simplest objective function for the optimisation problem (2.74, cf. [48]) can be written in time-domain representation as,

$$\min_{u_{i+1}(t)} J(u_{i+1}(t)) = \min_{u_{i+1}(t)} ||e_{i+1}(t)||^2 = \min_{u_{i+1}(t)} ||r(t) - \mathbf{P}u_{i+1}(t)||^2 \tag{2.74}$$

where the tracking error is the difference between desired trajectory and actual output, i.e. $e_{i+1}(t) = r(t) - \mathbf{P}u_{i+1}(t)$. One can interpret (2.74) as finding the optimal ILC signal $u_{i+1}(t)$ that will minimise the $e_{i+1}(t)$. It should be noted that the best possible solution that yields zero tracking error is in case of using the inverse of the system in the optimal solution of ILC input :

$$\begin{cases} u^*(t) = \mathbf{P}^{-1}r(t) \\ e_{min}(t) = r(t) - \mathbf{P}u(t)^* \end{cases} \tag{2.75}$$

with $e_{min}(t) = r(t) - \mathbf{P}\mathbf{P}^{-1}r(t) = r(t) - r(t) = 0$.

Equation (2.75) is the ideal solution and it is in general difficult to be obtained practically due to the unmodeled nonlinearities of the system dynamics. Thus, the goal of the optimisation algorithm is to approach the inverse solution as much as possible. Furthermore, a better solution can be obtained by including additional measures in the objective function (2.76) (see [48]) such as the input difference between each iteration, i.e.

$$\min_{u_{i+1}(t)} J(u_{i+1}(t)) = \min_{u_{i+1}(t)} ||e_{i+1}(t)||^2 + ||u_{i+1}(t) - u_i(t)||^2 \tag{2.76}$$

In the literature, ILC update laws that work on minimising some norm criteria such as the ones given in (2.74) and (2.76) (weight matrices can also be added for a better control of optimisation) are called norm-optimal ILC (NO-ILC). It is the basic optimisation-based approach used on ILC. The information given in [48] about NO-ILC is based on a more detailed material provided by [2] and it explains with proofs the step-by-step procedure in obtaining the optimal solution and the learning filter for NO-ILC. Moreover, some other examples of optimisation-based ILC methods can be counted as the predictive norm-

optimal ILC (PNO-ILC) and the parameter optimal ILC (PO-ILC). A PNO-ILC differs from NO-ILC by the fact that it includes the future predictions of the tracking error and the input difference. [48] gives the objective function of PNO-ILC as,

$$J(u_{i+1}(t)) = \sum_{k=1}^{n} \lambda^{k-1}(||e_{i+k}(t)||^2 + ||u_{i+k}(t) - u_{i+k-1}(t)||^2) \tag{2.77}$$

where n determines the total number of future iterations to be considered and $\lambda > 0$ is a weighting factor that defines the importance of future iterations that is similar to the forgetting factor in identification methods. In order to make this representation easier to understand, one may look at the case where $i = 1$, $n = 2$ and $\lambda = 0.5$:

$$\begin{aligned} J(u_2(t)) &= \sum_{k=1}^{2} \lambda^{k-1}(||e_{1+k}(t)||^2 + ||u_{1+k}(t) - u_{1+k-1}(t)||^2) \\ &= (0.5)^0(||e_2(t)||^2 + ||u_2(t) - u_1(t)||^2) \\ &\quad + (0.5)^1(||e_3(t)||^2 + ||u_3(t) - u_2(t)||^2) \end{aligned} \tag{2.78}$$

The first term in (2.78) involves the current values of tracking error and input difference whereas the second term considers the predictions of the same measures for the incoming iteration. Furthermore, it was suggested that the effect of adding these extra future values in the cost function should improve the convergence speed of PNO-ILC [48]. For the sake of brevity, the information regarding the latter method, i.e. PO-ILC, is not explained here and the reader may again refer to [48] for detailed explanations.

The NO-ILC is the core algorithm of the thesis work and it is used in all the remaining chapters. Therefore, it is given below a separate section for extensive explanations.

2.4.2 Norm optimal ILC (NO-ILC)

The algorithm is based on optimising a cost function and the aim is to find the best system input that will produce the closest output signal to the desired trajectory. The procedure proposed by Norrlof [84] can be seen as the main source of the following explanations. When deriving the NO-ILC algorithm, it is possible to use both time-domain and frequency-domain representations. Below only the time-domain one (see [84]) is provided since this is the one used in the following chapters.

Design in time-domain representation :

In order to have more control on the optimisation procedure, the cost function can be written using weighting matrices. The cost function is a quadratic criterion which is

subject to a constraint. The optimisation problem can be written as,

$$\begin{aligned} \min_{u_{i+1}(t)} J(u_{i+1}(t)) = \quad & \min_{u_{i+1}(t)} e_{i+1}^T(t)\mathbf{W}_e r_{i+1}(t) + u_{i+1}^T(t)\mathbf{W}_u u_{i+1}(t) \\ \textit{subject to} \quad & [u_{i+1}(t) - u_i(t)]^T[u_{i+1}(t) - u_i(t)] \leq \delta \end{aligned} \tag{2.79}$$

To use Lagrange method for minimisation, one should put the constrained optimisation problem in (2.79) into the form of an unconstrained problem that includes a Lagrange multiplier (λ) :

$$\min_{u_{i+1}(t)} J(u_{i+1}(t)) = \quad \min_{u_{i+1}(t)} e_{i+1}^T(t)\mathbf{W}_e e_{i+1}(t) + u_{i+1}^T(t)\mathbf{W}_u u_{i+1}(t) + \lambda[(u_{i+1}(t) - u_i(t))^T(u_{i+1}(t) - u_i(t)) - \delta] \tag{2.80}$$

The parameters effecting the solution of the optimisation problem in (2.79) and (2.80) can be listed as

— The weighting matrices, $\mathbf{W}_e$ and $\mathbf{W}_u$,
— The constraint on the input difference, δ and
— The Lagrange multiplier, λ.

The purpose of $\mathbf{W}_e$ and $\mathbf{W}_u$ is to determine which error and input elements will be of more importance along the minimisation process, respectively. The parameter δ, on the other hand, limits the increment of input between consecutive iterations while λ determines how the query goes towards optimal value. Moreover, for simplification it is assumed that the system has no measurement disturbances and the initial conditions for states are zero. Hence, the input-output relation can be expressed as,

$$y_i(t) = \mathbf{P}(r(t) + u_i(t)) \tag{2.81}$$

In (2.81), $u_i(t)$ is the input from ILC algorithm and the system has a desired reference $r(t)$. This means that the ILC signal modifies the reference in order to better the output tracking (i.e. a feedforward action). Using (2.81), the system error can be written as,

$$\begin{aligned} e_i(t) &= r(t) - y_i(t) \\ &= (\mathbf{I} - \mathbf{P})r(t) - \mathbf{P}y_i(t). \end{aligned} \tag{2.82}$$

It is straightforward that the error for the next trial is

$$\mathbf{e}_{i+1} = (\mathbf{I} - \mathbf{P})r(t) - \mathbf{P}u_{i+1}(t). \tag{2.83}$$

Now, (2.83) is placed into the objective function in (2.80) as,

$$\begin{aligned} J_{i+1}(u_{i+1}(t)) = & [(\mathbf{I} - \mathbf{P})r(t) - \mathbf{P}u_{i+1}(t)]^T\mathbf{W}_e e_{i+1}(t) \\ & + u_{i+1}^T(t)\mathbf{W}_u u_{i+1}(t) \\ & + \lambda[(u_{i+1}(t) - u_i(t))^T(u_{i+1}(t) - u_i(t)) - \delta] \end{aligned} \tag{2.84}$$

Applying the transposes and performing the multiplications leads to

$$\begin{aligned} J_{i+1}(u_{i+1}(t)) = &[(\mathbf{I}-\mathbf{P})r(t)]^T\mathbf{W}_e e_{i+1}(t) - (\mathbf{P}u_{i+1}(t))^T\mathbf{W}_e e_{i+1}(t) \\ &+ u_{i+1}^T(t)\mathbf{W}_u u_{i+1}(t) \\ &+ \lambda[(u_{i+1}(t)-u_i(t))^T(u_{i+1}(t)-u_i(t)) - \delta] \end{aligned} \tag{2.85}$$

Also, consider the following relation for derivation where v is a vector of variables and $\mathbf{M}$ is a matrix of constant values.

$$\frac{\partial(v^T\mathbf{M}v)}{\partial v} = 2\mathbf{M}v \tag{2.86}$$

Using (2.86), one may take the derivative of (2.85) with respect to $u_{i+1}(t)$.

$$\frac{\partial J_{i+1}}{\partial u_{i+1}(t)} = -\mathbf{P}^T\mathbf{W}_e e_{i+1}(t) + 2\mathbf{W}_u u_{i+1}(t) + 2\lambda(u_{i+1}(t)-u_i(t)) \tag{2.87}$$

Here, $\mathbf{e}_{i+1}$ may be replaced by using (2.83) and since the optimal value exists where the derivative is equal to zero,

$$-\mathbf{P}^T\mathbf{W}_e(\mathbf{I}-\mathbf{P})r(t) + \mathbf{P}^T\mathbf{W}_e\mathbf{P}u_{i+1}(t) + 2\mathbf{W}_u u_{i+1}(t) + 2\lambda(u_{i+1}(t)-u_i(t)) = 0 \tag{2.88}$$

This can be simplified by defining $2\mathbf{W}_u \equiv \mathbf{W}_u$ and $2\lambda \equiv \lambda$:

$$\begin{aligned} -\mathbf{P}^T\mathbf{W}_e(\mathbf{I}-\mathbf{P})r(t) + \mathbf{P}^T\mathbf{W}_e\mathbf{P}u_{i+1}(t) + \mathbf{W}_u u_{i+1}(t) + \lambda(u_{i+1}(t)-u_i(t)) &= \mathbf{0} \\ (\lambda\mathbf{I} + \mathbf{P}^T\mathbf{W}_e\mathbf{P} + \mathbf{W}_u)u_{i+1}(t) = \lambda u_i(t) + \mathbf{P}^T\mathbf{W}_e(\mathbf{I}-\mathbf{P})r(t) & \end{aligned} \tag{2.89}$$

Hence, the solution for $\mathbf{u}_{i+1}$ is obtained as,

$$\mathbf{u}_{i+1}(t) = (\lambda\mathbf{I} + \mathbf{P}^T\mathbf{W}_e\mathbf{P} + \mathbf{W}_u)^{-1}(\lambda u_i(t) + \mathbf{P}^T\mathbf{W}_e(\mathbf{I}-\mathbf{P})r(t)) \tag{2.90}$$

The next step is to write the expressions for the $\mathbf{Q}$ and $\mathbf{L}$ filters. In order to do that, let us consider writing (2.90) as in (2.41) by making the following substitution (consider $\mathbf{d} = \mathbf{0}$) :

$$(\mathbf{I}-\mathbf{P})r(t) = e_i(t) + \mathbf{P}u_i(t) \tag{2.91}$$

Then,

$$\begin{aligned} \mathbf{u}_{i+1}(t) &= (\lambda\mathbf{I} + \mathbf{P}^T\mathbf{W}_e\mathbf{P} + \mathbf{W}_u)^{-1}(\lambda u_i(t) + \mathbf{P}^T\mathbf{W}_e(e_i(t) + \mathbf{P}u_i(t))) \\ &= (\lambda\mathbf{I} + \mathbf{P}^T\mathbf{W}_e\mathbf{P} + \mathbf{W}_u)^{-1}(\lambda u_i(t) + \mathbf{P}^T\mathbf{W}_e e_i(t) + \mathbf{P}^T\mathbf{W}_e\mathbf{P}u_i(t)) \\ &= (\lambda\mathbf{I} + \mathbf{P}^T\mathbf{W}_e\mathbf{P} + \mathbf{W}_u)^{-1}[(\lambda\mathbf{I} + \mathbf{P}^T\mathbf{W}_e\mathbf{P})u_i(t) + \mathbf{P}^T\mathbf{W}_e e_i(t)] \\ &= [(\lambda\mathbf{I} + \mathbf{P}^T\mathbf{W}_e\mathbf{P} + \mathbf{W}_u)^{-1}(\lambda\mathbf{I} + \mathbf{P}^T\mathbf{W}_e\mathbf{P})][u_i(t) + (\lambda\mathbf{I} + \mathbf{P}^T\mathbf{W}_e\mathbf{P})^{-1}\mathbf{P}^T\mathbf{W}_e e_i(t)] \end{aligned} \tag{2.92}$$

From this result one can see that the ILC filters are

$$\begin{cases} \mathbf{Q} = (\lambda\mathbf{I} + \mathbf{P}^T\mathbf{W}_e\mathbf{P} + \mathbf{W}_u)^{-1}(\lambda\mathbf{I} + \mathbf{P}^T\mathbf{W}_e\mathbf{P}) \\ \mathbf{L} = (\lambda\mathbf{I} + \mathbf{P}^T\mathbf{W}_e\mathbf{P})^{-1}\mathbf{P}^T\mathbf{W}_e \end{cases} \tag{2.93}$$

$\underline{\lambda = 0 :}$

The challenge here for the designer is to determine λ, $\mathbf{W}_e$ and $\mathbf{W}_u$. Choosing $\lambda = 0$ converts ILC algorithm to a feedforward control logic since the control signal is found in one step. Setting $\lambda = 0$ in (2.90) leads to,

$$u_{i+1}(t) = (\mathbf{P}^T\mathbf{W}_e\mathbf{P} + \mathbf{W}_u)^{-1}(\mathbf{P}^T\mathbf{W}_e(\mathbf{I} - \mathbf{P})r(t)). \tag{2.94}$$

$\underline{\mathbf{W}_u = 0 :}$
If the weight on the input is removed, i.e. $\mathbf{W}_u = 0$, the system can reach its maximum convergence rate since there is no limit in the input increment. This result gives the following ILC filters :

$$\begin{cases} \mathbf{Q} = \mathbf{I} \\ \mathbf{L} = (\lambda\ \mathbf{I} + \mathbf{P}^T\mathbf{W}_e\mathbf{P})^{-1}\mathbf{P}^T\mathbf{W}_e \end{cases} \tag{2.95}$$

$\underline{\lambda = 0, \mathbf{W}_u = 0 :}$
If one chooses $\lambda = 0$ and $\mathbf{W}_u = \mathbf{0}$, then one can obtain the ultimate best solution for the ILC system where $\mathbf{L}$ becomes the inverse of the system. Thus, the filters are

$$\begin{cases} \mathbf{Q} = \mathbf{I} \\ \mathbf{L} = \mathbf{P}^{-1} \end{cases} \tag{2.96}$$

Comments :
However, this solution is usually not quite satisfactory considering the issues such as robustness, non-minimum phase systems and unwanted transients in the output. In order to cope with these issues, one can make the following choices :

$$\begin{cases} \lambda > 0 \\ \mathbf{W}_e = \mathbf{I} \\ \mathbf{W}_u = \rho\mathbf{I} \quad with\ \rho > 0 \end{cases} \tag{2.97}$$

Remark : A similar approach is found in Linear LQ control.
Thus, the ILC filters become

$$\begin{cases} \mathbf{Q} = ((\lambda + \rho)\mathbf{I} + \mathbf{P}^T\mathbf{P})^{-1}(\lambda\mathbf{I} + \mathbf{P}^T\mathbf{P}) \\ \mathbf{L} = (\lambda\mathbf{I} + \mathbf{P}^T\mathbf{P})^{-1}\mathbf{P}^T \end{cases} \tag{2.98}$$

Let us now write the update equation in the following form :

$$u_{i+1}(t) = \mathbf{Q}(u_i(t) + \mathbf{L}e_i(t)) = \mathbf{Q}u_i(t) + \mathbf{QL}e_i(t) \tag{2.99}$$

The convergence conditions are obtained by investigating largest singular values of $\mathbf{Q}$ and $\mathbf{QL}$ in (2.99) :

$$\begin{cases} ||\mathbf{Q}||_2 < 1 \\ ||\mathbf{QL}||_2 \leq \dfrac{1}{2\sqrt{\lambda + \rho}} \end{cases} \tag{2.100}$$

These conditions can be used when choosing the accurate filters for the ILC system. For more detailed mathematical explanations including the steps for acquiring these conditions, one should refer to [84].

2.5 Numerical Examples

In this section, two numerical examples are provided for giving the reader a better intuition of ILC. The first example demonstrates a frequency domain PD-type ILC application while the second example shows a time-domain norm optimal ILC (NO-ILC) application. The system to be controlled for these two applications is the following second order linear time-invariant single-input-single-output (SISO) system that is given in both its continuous-time and discrete-time forms (the discretisation is done using 'bilinear Tustin method' with a sampling time of $T_s = 0.01s$).

Continuous-time system :

$$P(s) = \frac{1}{0.3333s^2 + 1.167s + 1} = \frac{1}{(1 + 0.5s)(1 + 0.666s)} \tag{2.101}$$

Discrete-time system :

$$P(z) = \frac{0.0000737z^2 + 0.0001474z + 0.0000737}{z^2 - 1.965z + 0.9656} = \frac{0.000073705(z + 1)^2}{(z - 0.9851)(z - 0.9802)} \tag{2.102}$$

Note that the given system is already stable and thus it is suitable for being tested with an open-loop ILC in iterative feedforward fashion (for the opposite case where the initial system is unstable, an inner feedback controller must be employed to ensure stability). Furthermore, in both applications below, ILC is integrated to the system via serial connection as shown in Fig. 2.2.

2.5.1 PD-type ILC application

The first numerical analysis demonstrates the design procedure of a PD-type ILC for the discrete system (2.102). The design procedure given in this section is adopted from [68] and it should be referred to for further details. Before beginning this section, it can be useful to first read the basic design procedure of a traditional frequency-domain in Appendix A for an easier understanding. Let us consider the PD-type ILC filter proposed by [68] :

$$L(z) = z[K_p + K_d(1 - z^{-1})] \tag{2.103}$$

which is slightly different than a typical PD-type ILC filter (i.e. $L(z) = K_p + K_d(z - 1)$) since it applies the proportional and derivative operations on the error differently. Putting (2.103) into the general ILC update equation $u_{i+1}(t) = Q(z)[u_i(t) + L(z)e_i(t)$:

$$u_{i+1}(t) = Q(z)[u_i(t) + K_p e_i(t + 1) + K_d[e_i(t + 1) - e_i(t)] \tag{2.104}$$

It can be seen that the filter (2.103) applies a proportional gain K_p on the error from previous iteration that is shifted by one term in time (since ILC is applied offline, 'future error' of the previous iteration is already available) and it also multiplies the derivative of the error by a derivative gain K_d. Moreover, Q-filter is as mentioned before a low-pass

filter which can be of many different types such as Butterworh, Chebychev and so on. Thus, the main issue is to determine the proper values for the gains K_p and K_d. If (2.104) is arranged as

$$u_{i+1}(t) = Q(z)[u_i(t) + ((K_p + K_d)z - K_d)e_i(t)], \tag{2.105}$$

the learning filter can be rewritten as

$$\begin{aligned} L(z) &= (K_p + K_d)z - K_d \\ &= K_m z - K_d \end{aligned} \tag{2.106}$$

For a chosen convergence margin γ , ILC converges to the desired input $u_i(t)$ if

$$0 \leq |1 - L(e^{jw})P(e^{jw})| \leq \gamma, \quad \forall w \in [0, w_c] \tag{2.107}$$

where $P(e^{jw})$ is the plant. For $P(e^{jw}) = A(w)e^{j\Phi(w)}$ (2.107) is written as

$$0 \leq |1 - (K_m e^{jw} - K_d)A(w)e^{j\Phi(w)}| \leq \gamma, \quad \forall w \in [0, w_c] \tag{2.108}$$

Making the substitution $e^{jw} = \cos(w) + j\sin(w)$ and $e^{j\Phi(w)} = \cos(\Phi(w)) + j\sin(\Phi(w))$ and then taking the absolute of the resulting complex expression, one gets the following quadratic equation :

$$\begin{aligned} N_1 = A(w)^2 K_d^2 + [2A(w)\cos(\Phi(w)) - 2K_m A(w)^2 \cos(w)]K_d \\ [1 - \gamma^2 - 2A(w)K_m \cos(\Phi(w) + w) + K_m^2 A(w)^2] \leq 0 \end{aligned} \tag{2.109}$$

This can be seen as a quadradic equation with variable K_m and unknown parameter K_d. Then, the real solution of K_d is obtained if

$$\begin{aligned} N_2 = A(w)^2(\cos^2 w - 1)K_m^2 - 2A(w)\sin(\Phi(w))K_m \sin w \\ \cos^2(\Phi(w)) + \gamma^2 - 1 \geq 0 \end{aligned} \tag{2.110}$$

Since $A(w)^2(\cos^2 w - 1) < 0$, for $N_2 = 0$ the real solutions of K_m (i.e. λ_1 and λ_2) are found if the discriminant of (2.110) can be made greater than zero with the proper cut-off frequency w_c and γ, i.e. $\Delta = 1 - \cos^2 w_c > 0$. Thus, $\lambda_1 \leq K_m \leq \lambda_2$ which means that the value of K_m can vary between these solutions with the central value

$$K_m = -\frac{\sin \Phi w_c}{A(w_c)\sin w_c} \tag{2.111}$$

satisfying (2.110). Once the value for K_m is determined, considering $N_1 = 0$, one can obtain two real solutions (η_1 and η_2) similarly for K_d in (2.109) with the central value

$$K_d = K_m \cos w_c - \frac{\cos \Phi w_c}{A(w_c)} \tag{2.112}$$

satisfying the condition of (2.109). At this point, the stability condition (2.107) is guaranteed at $w = w_c$. However, this still does not mean that (2.107) will be fulfilled at low frequency, e.g. $w = 0$. Therefore, in order to be sure that the error will be zero at low frequencies, one must guarantee $|1 - L(z)P(z)| = 0$ at $w = 0$ which means that the gain

K_d in (2.112) has to be modified into

$$K_d = K_m - \frac{1}{A(0)} \tag{2.113}$$

where one can assign the proportional gain as $K_p = 1/A(0)$. This finally simplifies the overall design procedure since the sine and cosine terms cancel out. For a "common unit-loop closed system" as mentioned in [68], $A(0) = 1$ (other values are also possible depending on the plant) and then $K_m = K_d + 1$ from (2.113). This allows to rewrite (2.109) as

$$aK_d^2 + bK_d + (c - \gamma^2) = 0, \tag{2.114}$$

where

$$\begin{aligned} a &= 2A(w)^2[1 - cos(w)], \\ b &= 2A(w)[cos(\Phi(w)) - cos(\Phi(w) + w)] + 2A(w)^2[1 - cos(w)], \\ c &= 1 + A(w)^2 - 2A(w)cos(\Phi(w) + w). \end{aligned} \tag{2.115}$$

The solution for K_d exists when

$$\begin{aligned} a &> 0 \\ \frac{4ac - b^2}{4a} &< \gamma^2 \end{aligned} \tag{2.116}$$

Thus, the solutions of (2.114) are obtained as

$$\begin{aligned} K_{d,1} &= \frac{-b + \sqrt{b^2 - 4ac}}{2a} \\ K_{d,2} &= \frac{-b - \sqrt{b^2 - 4ac}}{2a} \end{aligned} \tag{2.117}$$

Next, to satisfy the stability condition (2.107) at $w = 0$, the following relation should hold between gains K_m and K_d :

$$K_m = K_d + \frac{1}{A(0)} \tag{2.118}$$

One can see that the calculation of K_m and K_d depends on the chosen cut-off frequency w_c and the convergence margin γ. The designer should find the highest w_c at which the ILC system can operate without diverging. Then, a low-pass filter should be used to eliminate the frequencies greater than this cut-off frequency.

Simulation results :

The system desired reference is chosen to be a sinusoidal signal, i.e. $y_d(t) = \sin(t)$. The rest of the parameters used in the initialization of the ILC process is given on Table 2.2. Note that γ is selected as 1 meaning that there is no extra restriction on the convergence rate and the maximum cut-off frequency is selected around 19Hz via trial-and-error considering the discriminant condition for (2.110) (see the design explanation therein). Also,

the Q-filter is assumed to be $Q(z) = 1$ for this example and other selections (i.e. low-pass filters mentioned in the design procedure given above) are possible for eliminating high-frequencies within the control signal $u_{i+1}(t)$.

TABLE 2.2: PD-type ILC parameters

Parameters	
Sample time, T_s	0.01 sec.
Simulation time, T_{sim}	20 sec.
Discretization method	$Tustin$
Initial system input, $u_{sys}(t)$	$y_d(t)$
Initial ILC input, $u_{ILC}(t)$	0
Number of ILC iterations	15
Cut-off frequency, w_c	19 Hz
Convergence margin, γ	1

Under these initial conditions, (2.107) is satisfied since

$$\begin{aligned} a &= 1.5152 \times 10^{-6} > 0 \\ \frac{4ac - b^2}{4a} &= 0.9395 < 1 \end{aligned} \tag{2.119}$$

The solutions for the gains for $w_c = 19$Hz and $\gamma = 1$ are

$$\begin{aligned} K_d &= 27.76 \\ K_m &= 28.76 \end{aligned} \tag{2.120}$$

Therefore, the learning filter is

$$L(z) = 28.76z - 27.76 \tag{2.121}$$

which satisfies the stability condition (2.107) by making

$$|1 - L(e^{jw_c})P(e^{jw_c})| = 0.9968 < 1 \tag{2.122}$$

The system response without ILC which is shown by Figure 2.4 has a phase delay and a lower amplitude. When the ILC is activated the system approaches the desired trajectory in 15 iterations. The first and every fifth iteration of this process are provided in Figure 2.5. Similarly, the error and the control signal produced by ILC can be observed in Figure 2.6 and Figure 2.7, respectively. It can be seen that the error gets lower and lower with each iteration while the control signal converges to the proper value. Figure 2.8 gives the variation of the squared norm of the system error $||e_i(t)||^2$ which illustrates that the error becomes nearly zero in 10 iterations. Also, the squared norm of the difference of the ILC control signals between consecutive iterations $||u_{i+1}(t) - u_i(t)||^2$ approaches zero which is

seen in Figure 2.9. This is an expected result since the control signal converges as iterations continue.

Finally, Figure 2.10, Figure 2.11 and Figure 2.12 can be reviewed for the results regarding the last iteration only. One can figure out that the ILC output corresponds to the exact desired output after 6 seconds and there is a slight deviation between the seconds 1 and 2. Increasing the iterations to 150, almost completely solves this problem ; however this still creates small deviations at some other parts of the signal. If the iterations continue even further until 1000, the output becomes flawless.

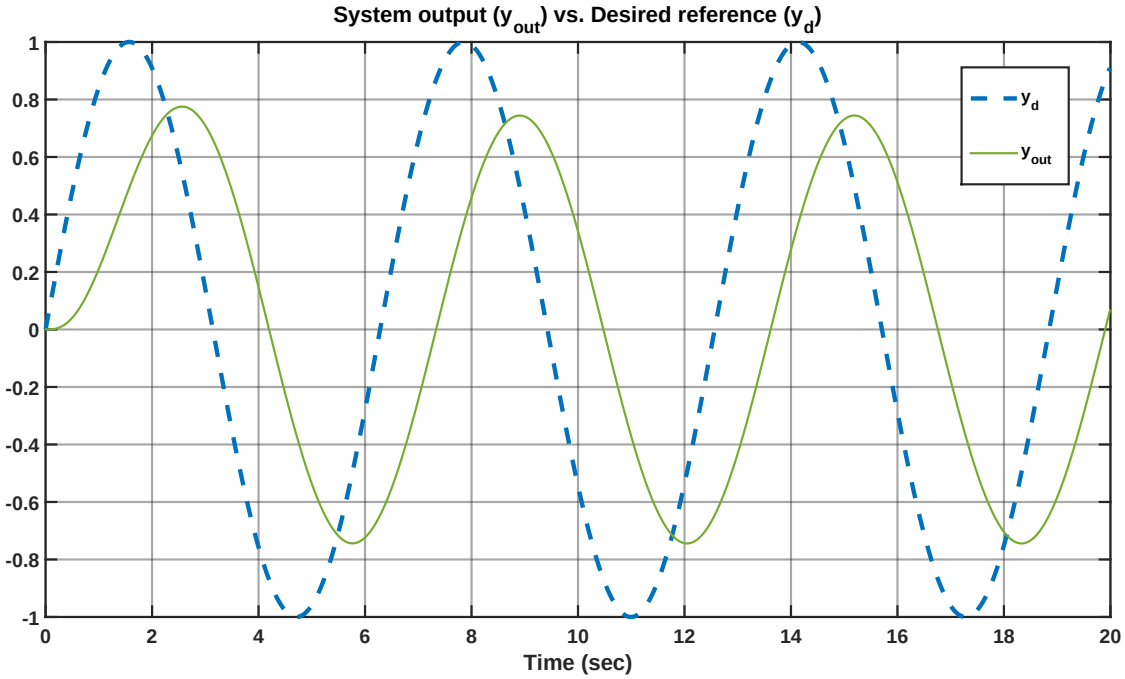

FIGURE 2.4: System output without ILC intervention.

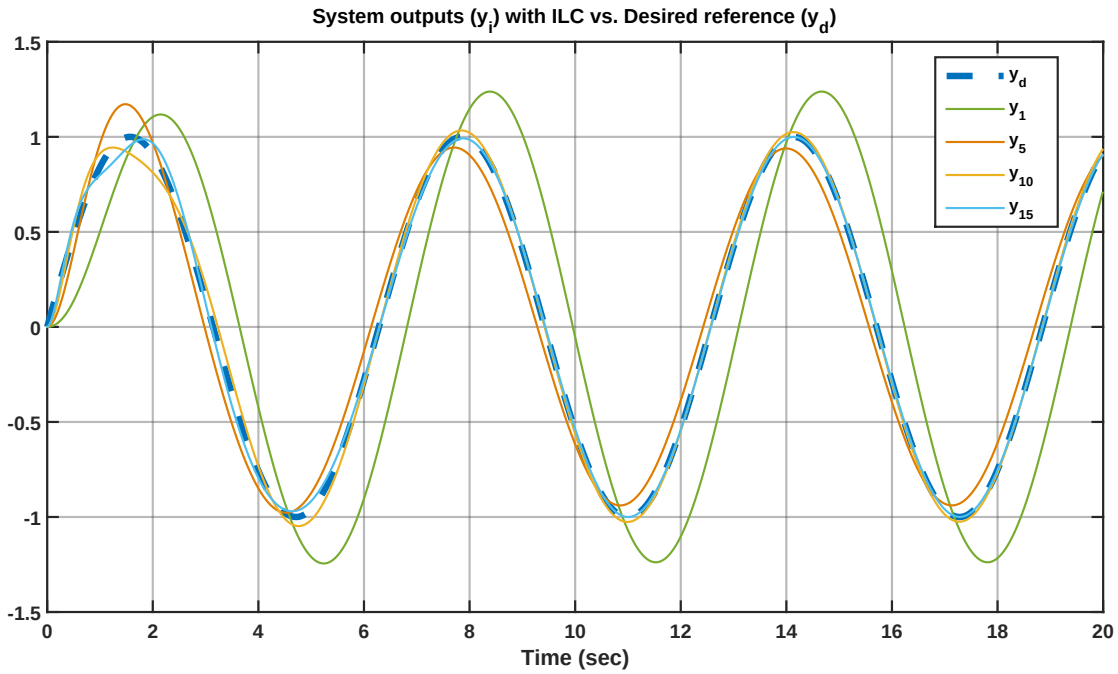

FIGURE 2.5: Effect of ILC on the system outputs.

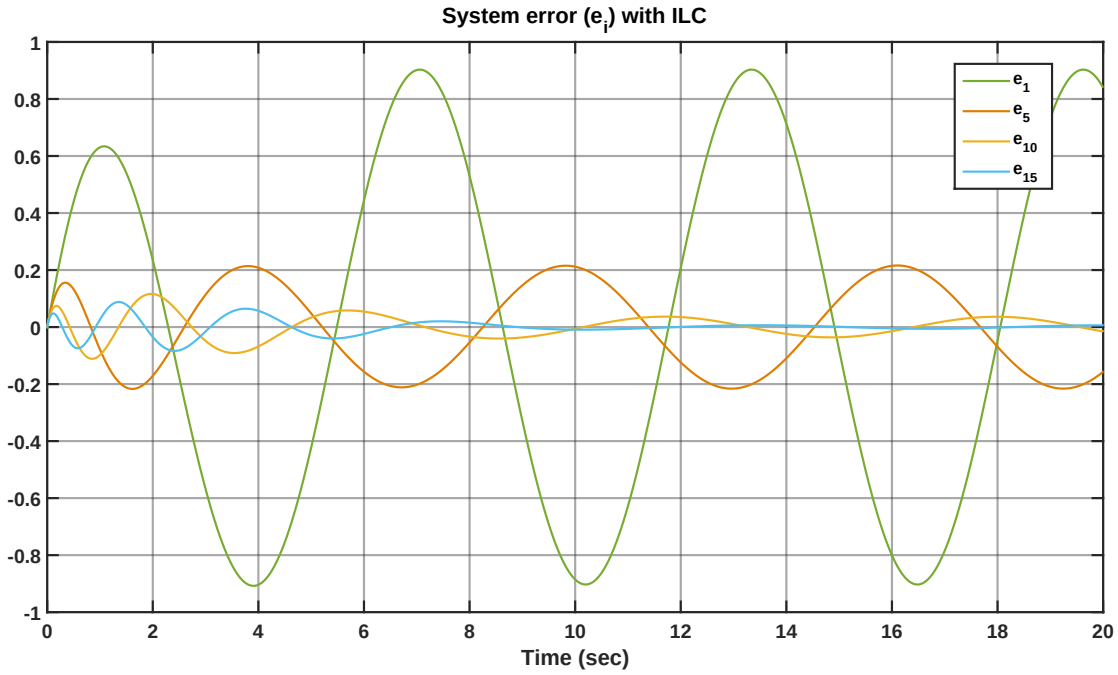

FIGURE 2.6: System error for each iteration.

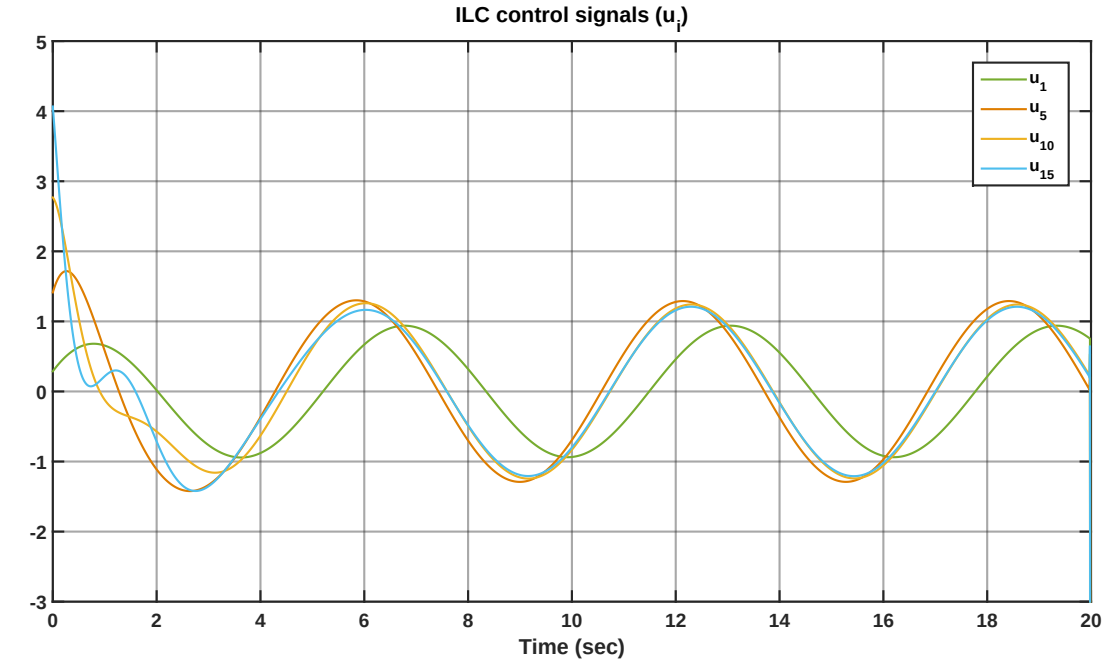

FIGURE 2.7: ILC control signals for each iteration.

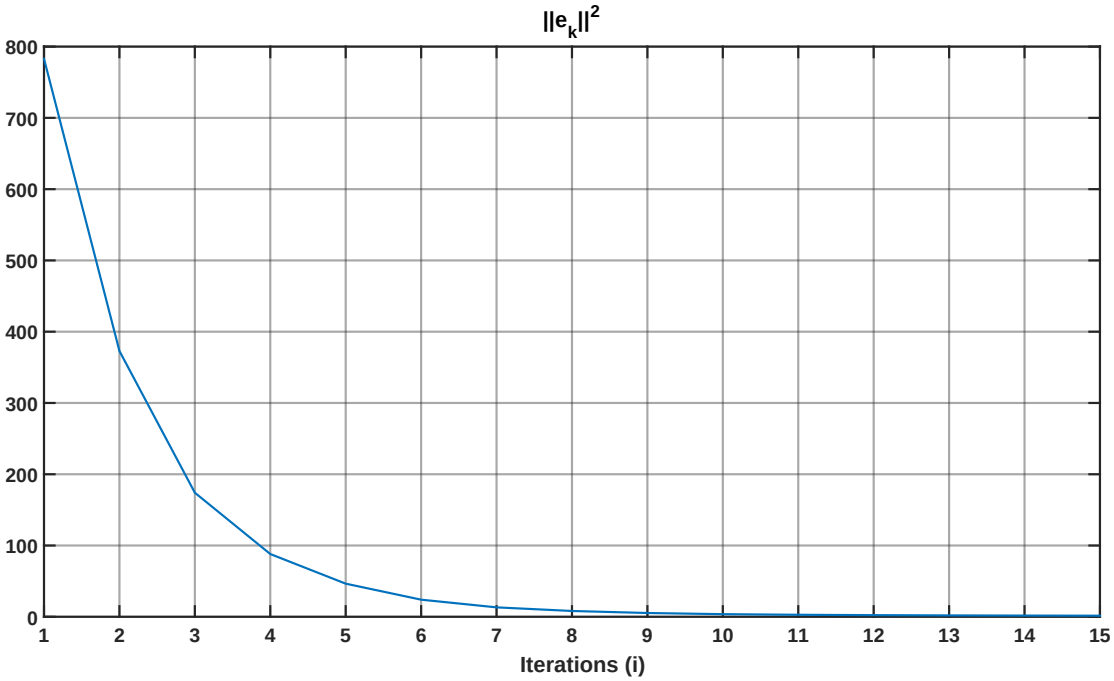

FIGURE 2.8: Squared norm of the system error with respect to ILC iterations.

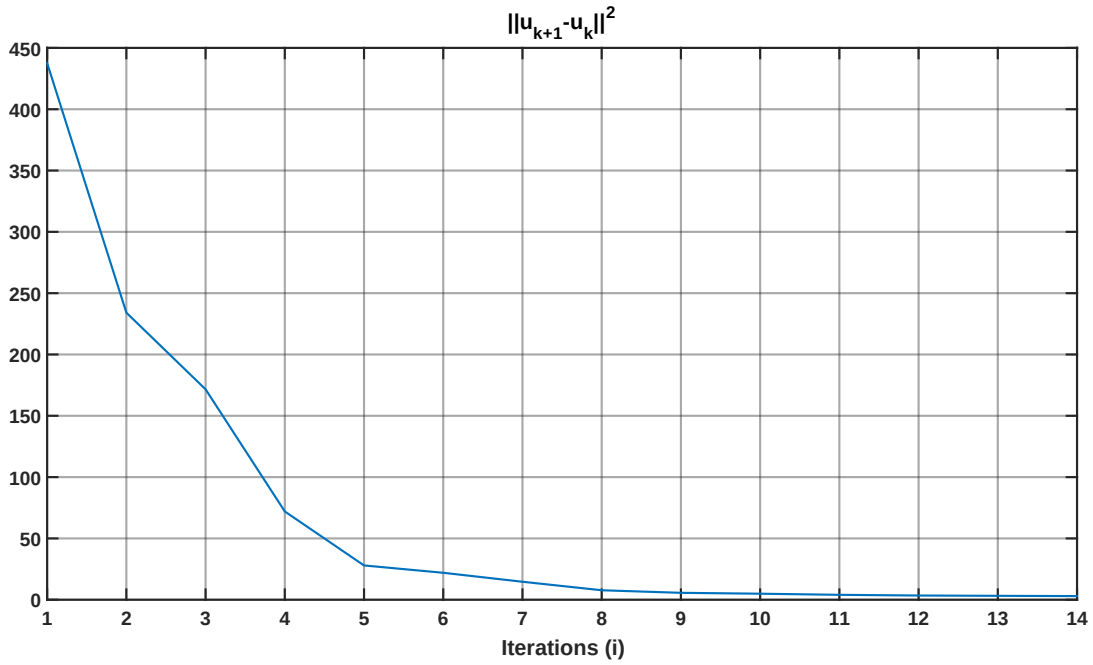

FIGURE 2.9: Squared norm of the ILC output difference with respect to ILC iterations

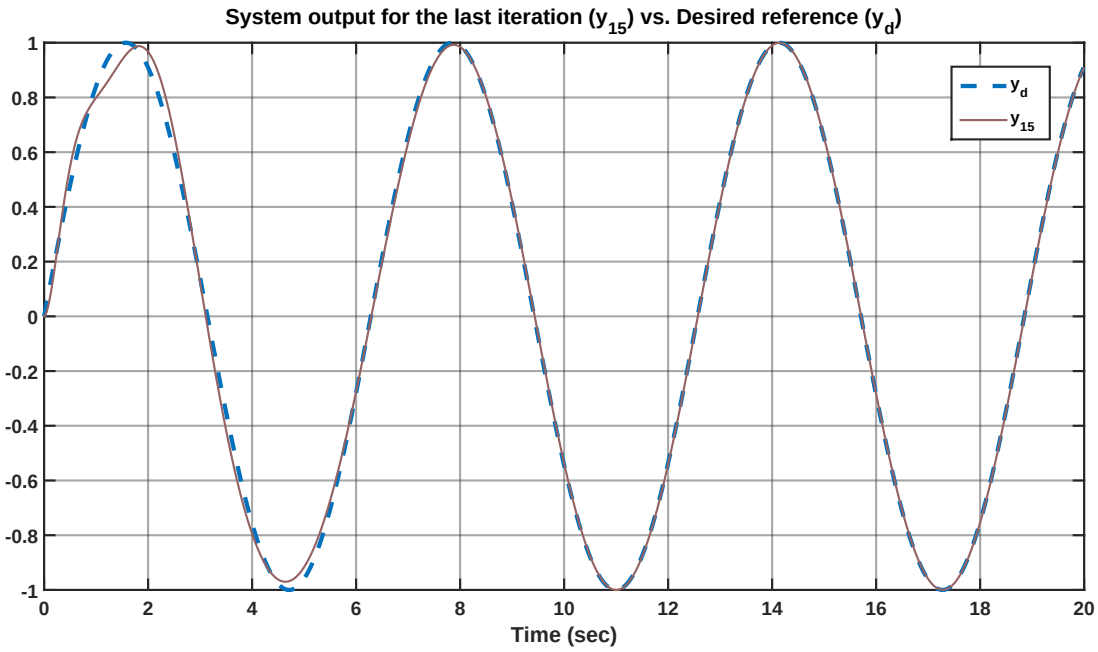

FIGURE 2.10: System output for the last ILC iteration

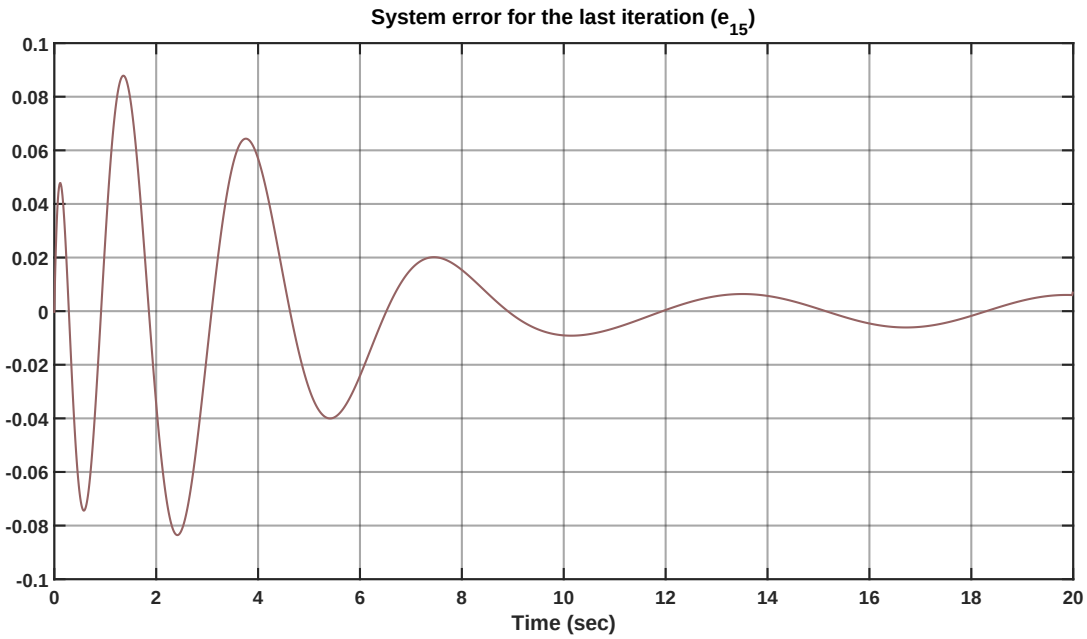

FIGURE 2.11: System error for the last ILC iteration

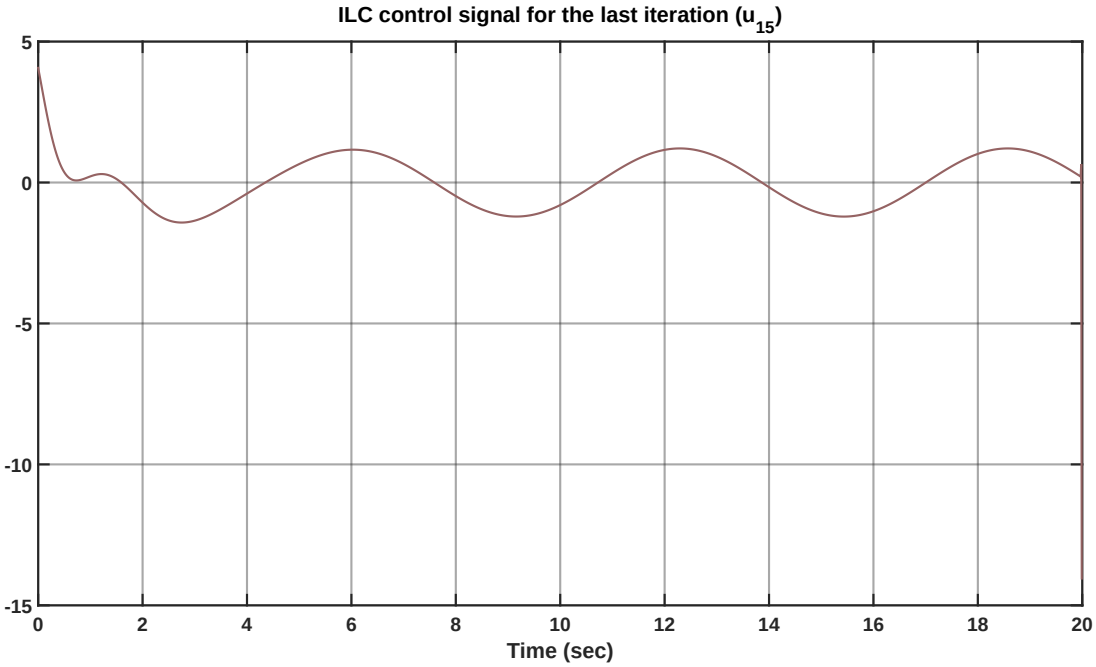

FIGURE 2.12: ILC control signal for the last iteration

2.5.2 NO-ILC application

The second numerical analysis shows the design procedure of a norm-optimal ILC in time-domain representation. The procedure described in Section 2.4.2 is applied on the state-space version of the system (2.102), i.e.

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{2.123}$$
$$y(t) = Cx(t) + Du(t). \tag{2.124}$$

with $x(t), y(t) \in \mathbb{R}^2$, $u(t) \in \mathbb{R}^1$ and

$$A = \begin{bmatrix} -3.5014 & -3.0003 \\ 1 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

The system (4.1)-(4.2) is discretised on *MATLAB* via 'Tustin method' and this time with larger sampling time to reduce heaviness of calculations, i.e. $T_s = 0.01s$). The system is then put in the lifted-framework by obtaining its impulse response and following the procedure given in Section 2.1.1. For keeping the simplicity, the desired reference trajectory is again chosen as $y_d(t) = \sin(t)$. The remaining simulation parameters used for the NO-ILC simulation are given on Table 2.3.

TABLE 2.3: NO-ILC parameters

Parameters	
Sample time, T_s	0.1 sec.
Simulation time, T_{sim}	50 sec.
Number of samples, T_{sim}/T_s	M
Discretization method	$Tustin$
Initial states	$[0\ 0\ 0\ 0\ 0]^T$
Initial system input, $u_{sys}(t)$	$y_d(t)$
Initial ILC input, $u_{ILC}(t)$	0
Number of ILC iterations	40
W_e	ρI
W_u	I
ρ	0.01
λ	0.1

It can be seen that the selections of ρ and λ on Table 2.3 satisfy the convergence requirements in (2.100) by yielding,

$$\begin{aligned} ||Q||_2 &= 0.9999 < 1 \\ ||QL||_2 &= 1.5075 < 1.5076 \end{aligned} \tag{2.125}$$

In the beginning of the process, the system is run without any ILC intervention, i.e.

$u_{ILC} = 0$. This is required for gathering the data that is needed to calculate the first ILC output which will modify the reference signal on the next run. The output of the system without ILC can be reviewed in Figure 2.13. On the other hand, one can observe in Figure 2.14 that as the iterations continue the output of the system with ILC gets closer and closer to the desired trajectory. After 40 iterations the system achieves almost the exact desired trajectory. Furthermore, the convergence of the error and the control signals of the ILC system are depicted in Figure 2.15 and 2.16, respectively. The converged signals, i.e. the last iteration, are shown in blue while the green signals are for the first iteration. One can observe how ILC manipulates the control signal and thus the error by looking at the signals plotted in red. Furthermore, Figure 2.17 and Figure 2.18 shows the squared norm values of the error and the ILC output difference, respectively. The error of the ILC system is very close to zero after 25 iterations and as the process continues towards 40^{th} iteration the error is nearly diminished. Also, the change in control signal between two consecutive iterations approaches nearly to zero in 25 iterations and a full convergence approximately occurs in 40 iterations. Besides, Figure 2.19, Figure 2.20 and Figure 2.21 are the illustrations of the output, the error and the control signals corresponding to the last iteration, respectively. One may refer to these figures to analyse the converged signals in more detail.

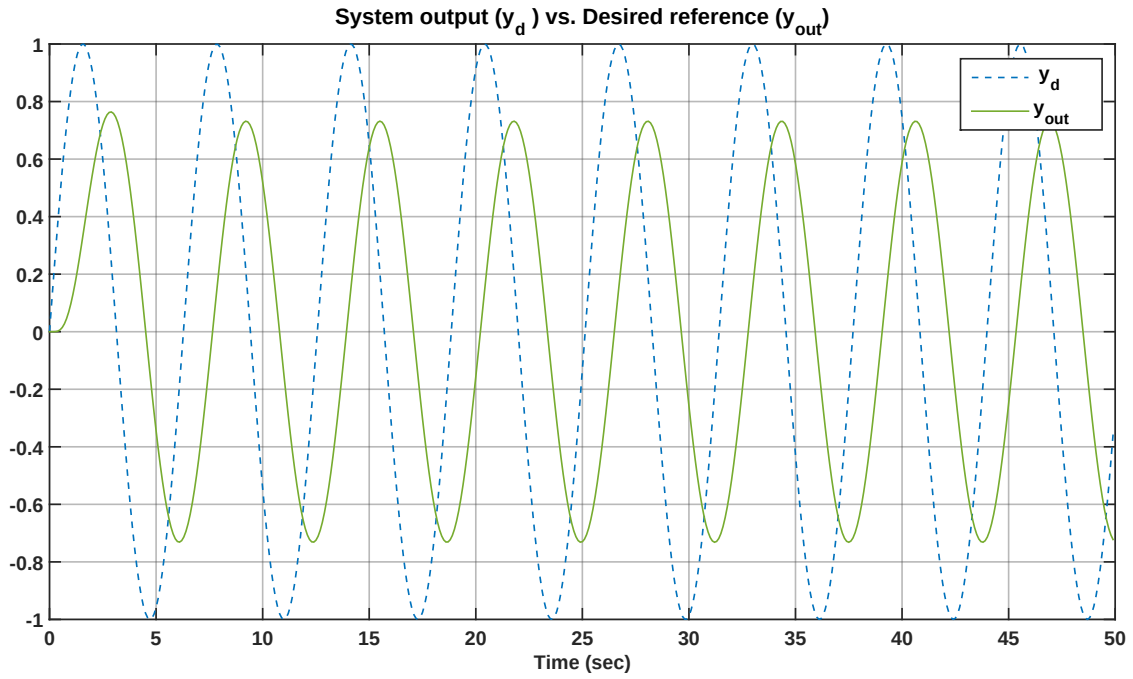

FIGURE 2.13: System output without ILC intervention.

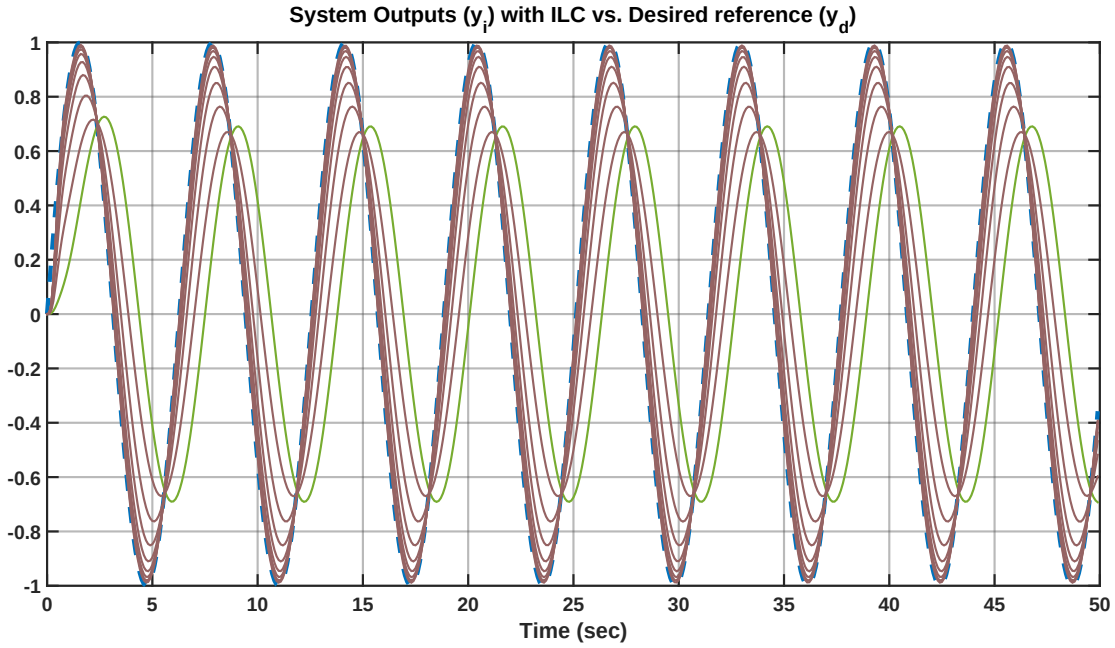

FIGURE 2.14: Effect of ILC on the system outputs : the 1^{st} iteration (*green*), every 5^{th} iteration (*red*) and the desired reference (*blue*).

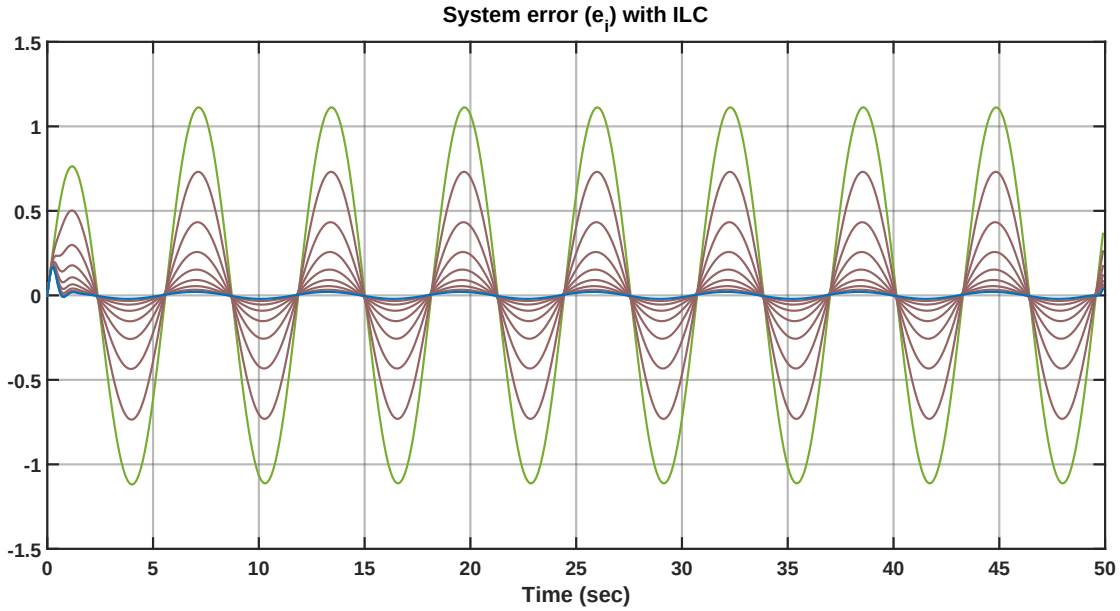

FIGURE 2.15: System error for each iteration : the 1^{st} iteration (*green*), every 5^{th} iteration (*red*) and the last iteration (*blue*).

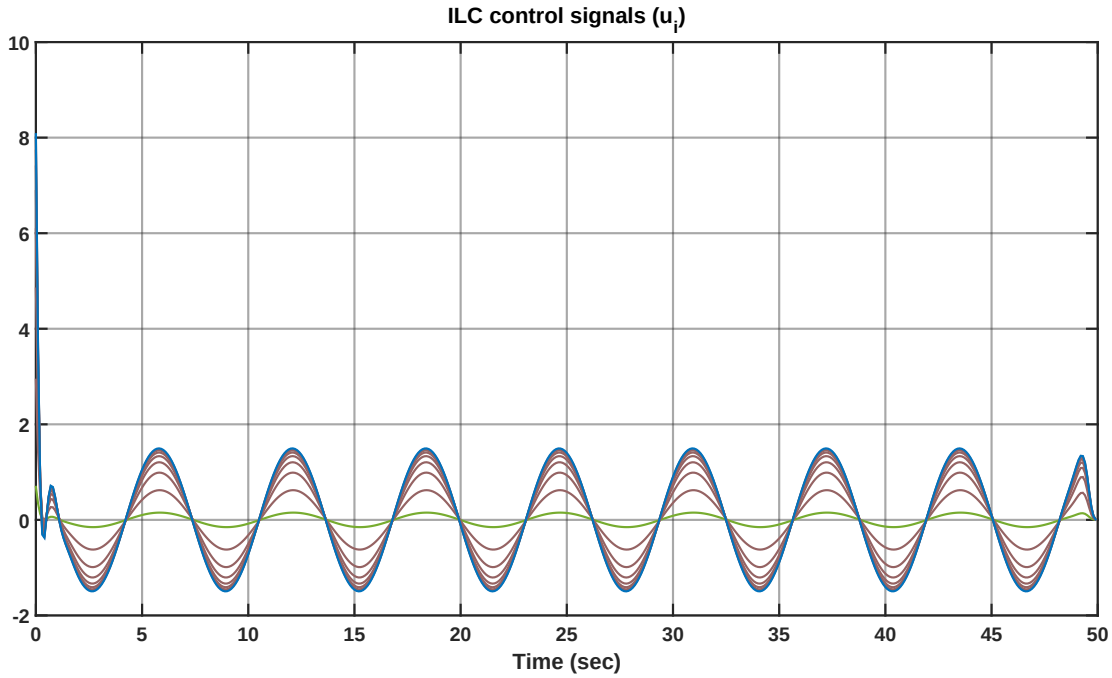

FIGURE 2.16: ILC control signals for each iteration : the 1^{st} iteration (*green*), every 5^{th} iteration (*red*) and the last iteration (*blue*).

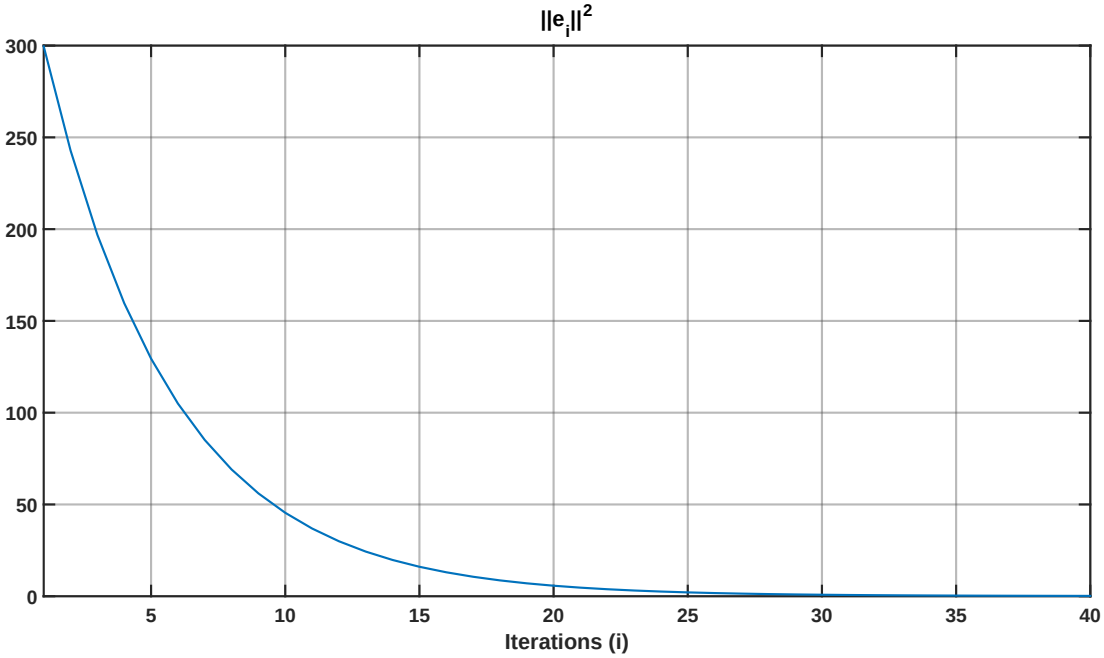

FIGURE 2.17: Squared norm of the system error with respect to ILC iterations.

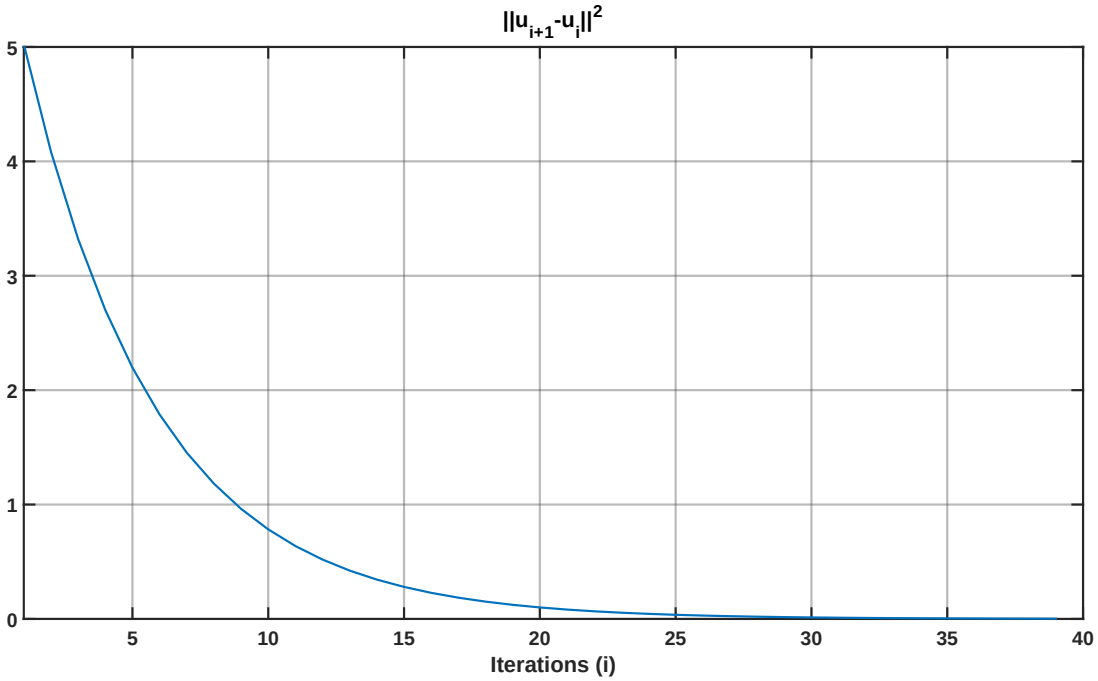

FIGURE 2.18: Squared norm of the ILC output difference with respect to ILC iterations

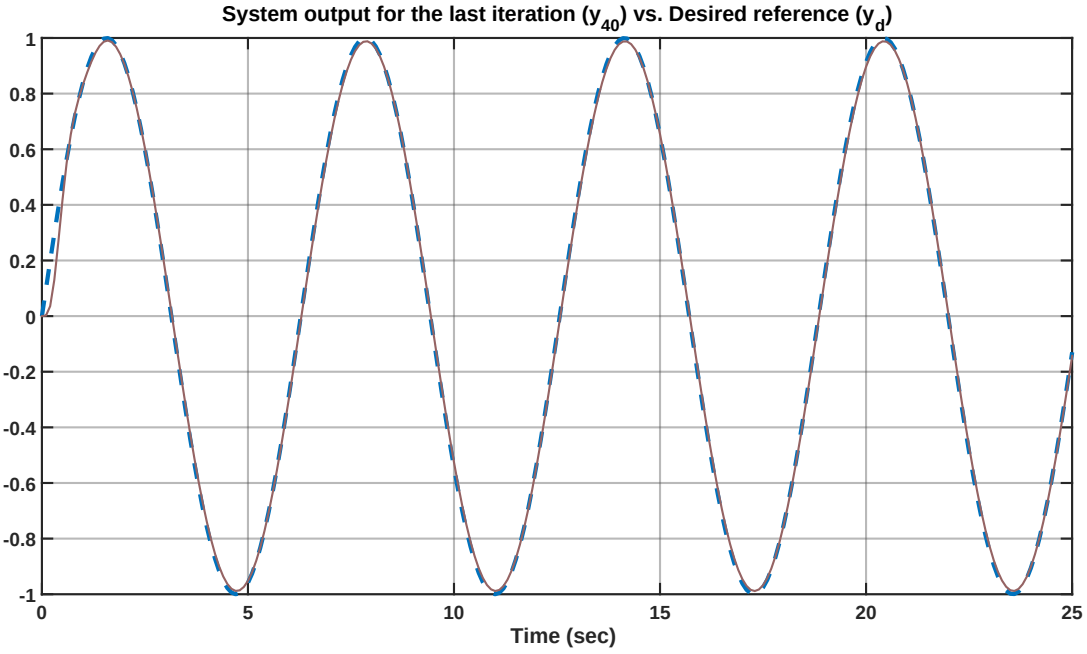

FIGURE 2.19: System output for the last ILC iteration

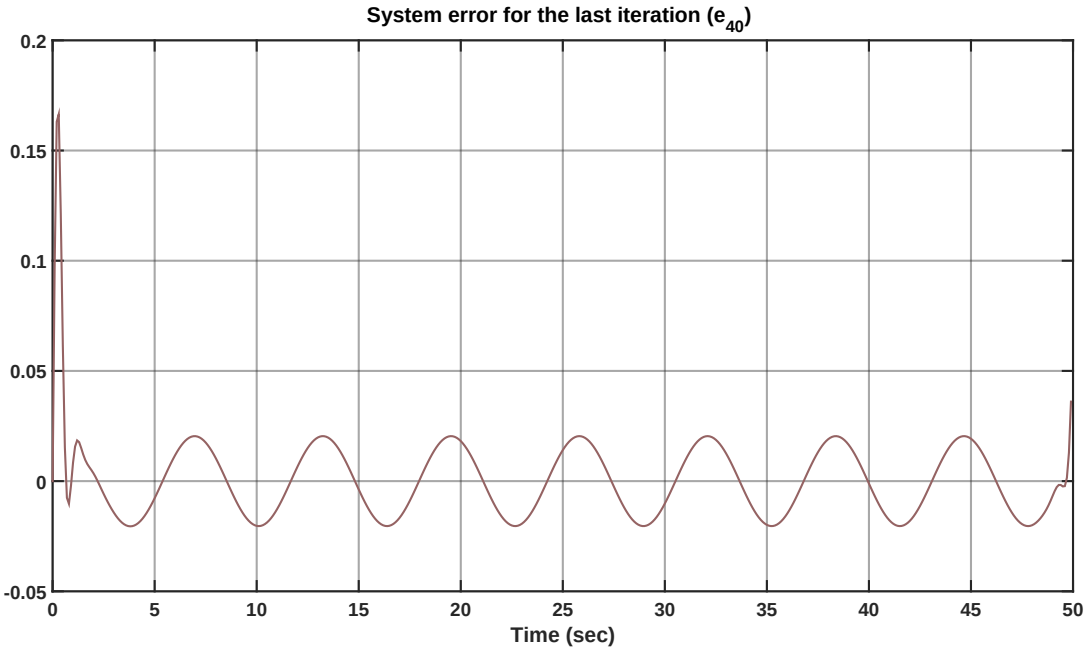

FIGURE 2.20: System error for the last ILC iteration

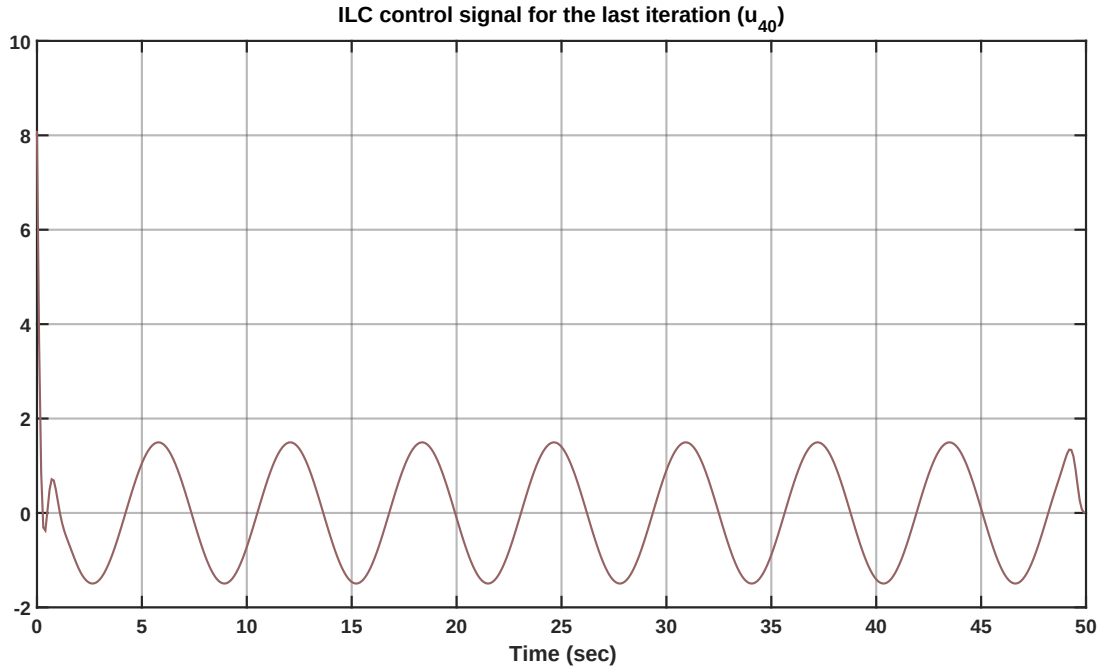

FIGURE 2.21: ILC control signal for the last iteration

Practical note : *In this numerical application, the main aim is to demonstrate how ILC reduces the system error while it converges to a specific input value. However, similar applications on different systems may need a more precise consideration from the practical point of view. Instead of heuristically finding the best number of iterations for the needed convergence results, one can define a termination criteria considering the normalised versions of the squared norms* $||e_i(t)||^2$ *and* $||u_{i+1}(t) - u_i(t)||^2$*. This kind of approach allows to have an independency with respect to the studied system. For example, the termination*

criteria of ILC can be defined considering the normalised input difference as bellow :

C.1 Termination (normal) : $\frac{||u_{i+1}(t)-u_i(t)||^2}{||u_1(t)-u_0(t)||^2} = \frac{||\Delta u_i(t)||^2}{||\Delta u_0(t)||^2} < \epsilon_u\%$

C.2 Termination (more robust) : $\frac{||\Delta u_i(t)||^2-||\Delta u_{i-1}(t)||^2}{||\Delta u_{i-1}(t)||^2} < \epsilon_u\%$

where ϵ_u is an ultimate goal for the ILC input difference variation in percentage chosen by the user. After some point, the input may start to vary so low that further iterations may not affect precision performance. This point can be imagined as ϵ_u.

Similarly, the termination criteria can be constructed considering the amount of error :

C.3 Termination (normal) : $\frac{||e_i(t)||^2}{||e_0(t)||^2} < \epsilon_e\%$

C.4 Termination (more robust) : $\frac{||e_i(t)||^2-||e_{i-1}(t)||^2}{||e_{i-1}(t)||^2} < \epsilon_e\%$

where ϵ_e is an ultimate goal for the ILC error variation in percentage that is left to user's preference. In order to include the impact of measurement noise, ϵ_e can be obtained chosen with respect to σ_y^2 which is the variance of the additive noise in the output $y(t)$. In conclusion, the user can simply stop the ILC iterations utilising these four above given criteria.

2.6 Conclusion

The concept of iterative learning control has been explained in detail [1] including the fundamental idea behin ILC, the design steps, the different representations, the stability analysis, the integration of ILC, the existing ILC methods, the order of ILC, some practical suggestions and two numerical applications. It can be said that ILC has a simple structure ; yet it can be quite effective in high precision applications. Although it is open-loop, it allows a very easy integration to an existing system that is already controlled via feedback without effecting the inner controller or system parameters. This feature of ILC is rather motivating for carrying out a practical application with the aim of improving the system performance in reference tracking beyond feedback capabilities. One of the main features that can provide this kind of result is ILC's anticipation property that comes from its data-based nature. ILC can simply use the repetitive data from previous system iterations to remove transients and improve precision to a very high level.

As a result of such motivation, the initial study of the thesis has began by applying an existing ILC method to a real indoor UAV. Among the presented ILC methods in this chapter, the norm optimal ILC (NO-ILC) has been chosen as a good candidate since it is more suitable for uncertainties, disturbances or noise compared to other ILC methods such as P-type ILC, D-type ILC, PD-type ILC and so on. Thus, the following chapter demonstrates an ILC experimentation on a UAV and, in addition, it indroduces a new experimental procedure for speeding up the UAV flight experiments using ILC.

1. At a fairly high level (further information is available in the references).

Chapter 3

Applying ILC on a real UAV

Contents

This chapter can be seen as the first step in the development of the thesis which is based on an ILC application on a real indoor UAV. The main goals of the presented work here are to test the feasibility of an optimisation-based ILC on real UAV and to propose a new experimentational procedure for ILC experiments with UAVs that speeds up the conventional approach.

3.1 UAV Oriented Problem Statement

In almost all ILC experiments with UAVs, the main practice is to perform one ILC update after each flight of the same reference trajectory. This corresponds to the classical way of applying an ILC scheme. The tracking precision is improved throughout the flights by means of updating the system inputs utilising experimental data from previous flight(s). Even though this update scheme gives desired results in terms of converged error, its efficiency is arguable when one considers the number of flights needed for achieving the desired convergence. The more this number increases, the more becomes the duration for reaching the required tracking performance. Some other factors that increase the time of UAV experimentation can be counted as the time spent for the experimental setup, sampling time, flight duration, battery changes and various unexpected problems that may

occur along the flight causing a rerun. Under such factors, even conducting 20 experiments may take sufficiently long, not to mention the cases in which the number of experiments reaches more than 100 as in [74]. On the other hand, it is usually not very difficult to obtain the linear dynamic model of a UAV for a given trajectory (linear models can be considered as accurate enough on local trajectories and even though a linear model may not be well-fitting for very aggressive trajectories, it can still be sufficient to be used in ILC).

> If the task is to reach a certain tracking performance, using a linear model with ILC can make it much faster to approach the desired performance when ILC is applied in a non-traditional way by using predicted iteration data

. The linear model can be utilised to predict the future flight data and to carry out large number of ILC iterations which would then allow one to largely improve the ILC signal without needing to actually perform real flight experiments. What makes this practical approach interesting is that one can highly improve the tracking accuracy by using simple linear models such as UAV models identified with a few number of poles and zeros.

In the following section, the control design is presented. First, the initial requirements for the reference trajectory is explained. Then, the UAV's internal feedback controller and the ILC algorithm are shown.

3.2 Philosophy and Control Design

In order to obtain proper results from ILC experimentation on a UAV, it is necessary to take into account some factors for making the control objectives feasible as per the dynamics of the UAV. This section demonstrates these factors under a cascade control design architecture that consists of three parts : the reference trajectory, the internal feedback controller and the ILC. The usage of this control architecture is aimed to be motivated by putting an emphasis on its practicality.

3.2.1 Reference trajectory & Pre-filter

For an ILC application, it is assumed that the operator can give an arbitrary repetitive sequence of position (or velocity) setpoints. As this sequence can be discontinuous and not achievable by the UAV, this raw sequence should first be filtered systematically in order to smoothen it before it enters the UAV system as a reference input [1]. For this purpose, in the context of a UAV, a 3rd order pre-filter is a suitable choice since it will render the raw repetitive sequence continuous along the attitude references as well as twice differentiable (i.e. suitable for ILC control objectives). This process can be interpreted as allowing some time for the UAV to reach the desired position at the beginning of the reference trajectory which is initialised according to the UAV's initial state (e.g. asking a UAV to be at $1m$ immediately at $t = 0s$ is not realistic in terms of physical limitations). Moreover, pre-filters

1. This is directly related to systems achievable performances

of same structure are applied to each axis of the UAV such that it is possible to separately adjust the DC-gains and to produce only a small phase delay without having any change in amplitude with respect to the initial trajectory given by the operator. The pre-filter's transfer function can be described as :

$$\frac{(.)_{ref}(s)}{(.)_{raw\_ref}(s)} = \frac{G_{(.)}}{(\tau_{1(.)}s+1)(\tau_{2(.)}s+1)(\tau_{3(.)}s+1)} \tag{3.1}$$

where (.) can be X, Y or Z, the '*ref*' and '*raw_ref*' stand for the filtered and unfiltered references, respectively and $\tau_{1(.)}$,$\tau_{2(.)}$ and $\tau_{3(.)}$ represent the time responses chosen to be less than 2 seconds. Here, $G(.)$ is the inverse of the gain of the filter's transfer fuction evaluated at the reference frequency w_{ref}, i.e.

$$G_{(.)} = \frac{1}{|H(jw)|}\bigg|_{w=w_{ref}}$$

where

$$H(jw_{ref}) = \frac{1}{(\tau_{1(.)}jw_{ref}+1)(\tau_{2(.)}jw_{ref}+1)(\tau_{3(.)}jw_{ref}+1)}$$

This filtering procedure is applied on three different UAV trajectories which are used during the experimentation and simulations in Section 3.3. These reference trajectories can be seen below (note that reference signal construction is a complete research field where contributions can be found in robotics).

1 - Eight-shaped trajectory

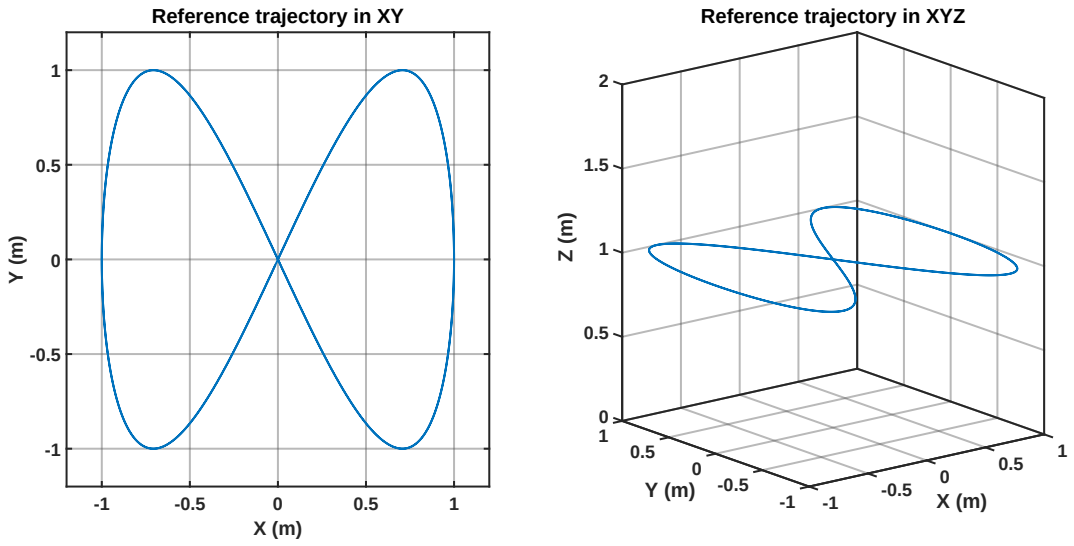

FIGURE 3.1: Eight-shaped reference trajectory (no filter)

The UAV is asked to follow an eight-shaped trajectory with a period of $T = 7.5$ seconds on its XY-plane. The altitude of the UAV is set to 1 meter in Z-axis. The eight-shape is obtained by combining two sinusoidal signals on X and Y-axes with the same amplitude ($A_{X,Y} = 1m$) but different frequencies; i.e. $f_X = 0.5f_Y$. The motion starts from starts

from hovering position in the center of the trajectory at the specified altitude. The raw and filtered versions of this trajectory can be viewed in Fig. 3.1 and 3.2, respectively.

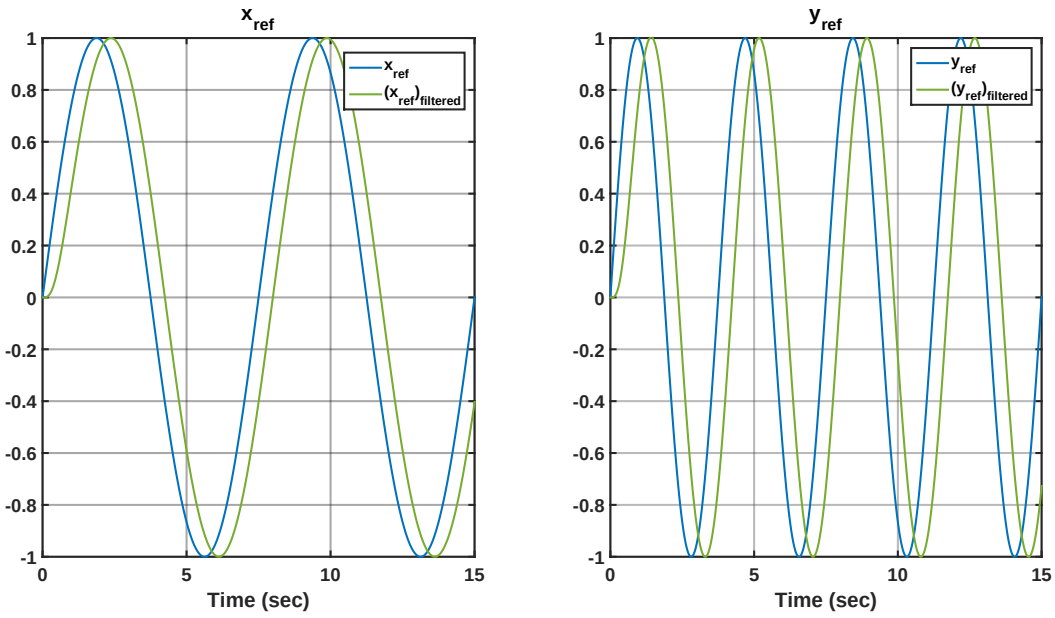

FIGURE 3.2: Filtered eight-shaped reference trajectory vs. no filter

2 - Square-shaped trajectory

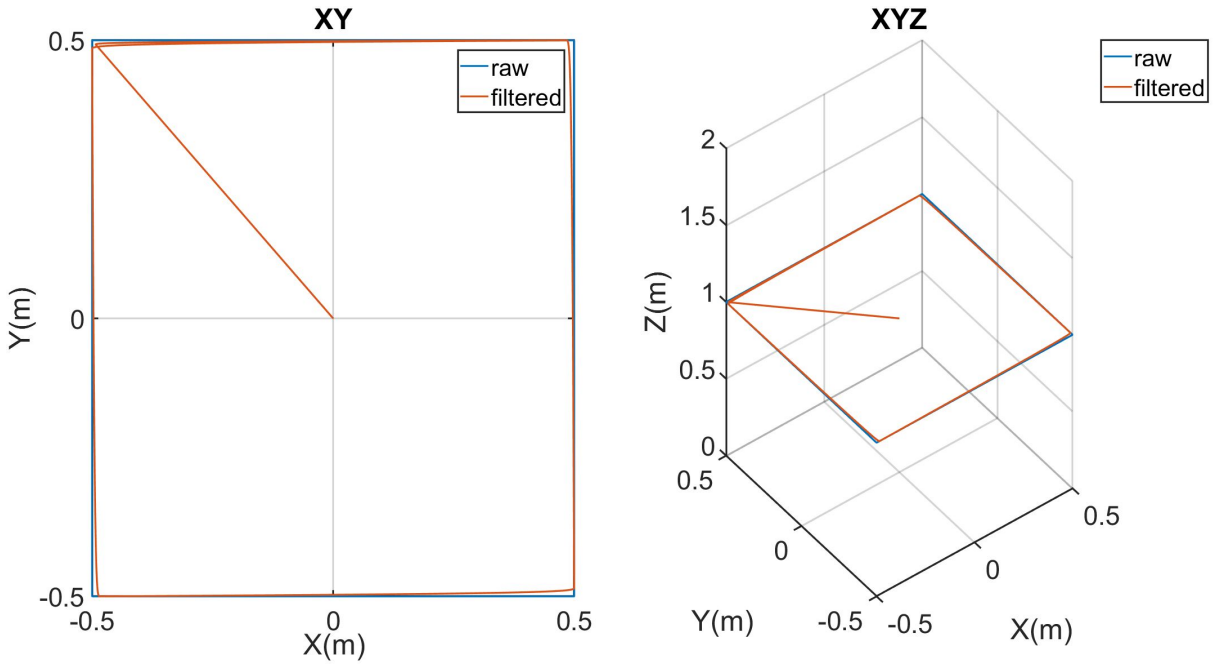

FIGURE 3.3: Square reference trajectory (left : top view, right : tilted side view)

The UAV is asked to follow a square-shaped trajectory with a period of 8 seconds on the XY-plane. The length of the trajectory is 1 meter in each axis. The motion starts from the center of the square where the UAV is at hovering position at $1meter$ of altitude. The original and filtered trajectories can be observed in Fig. 3.3 and Fig. 3.4.

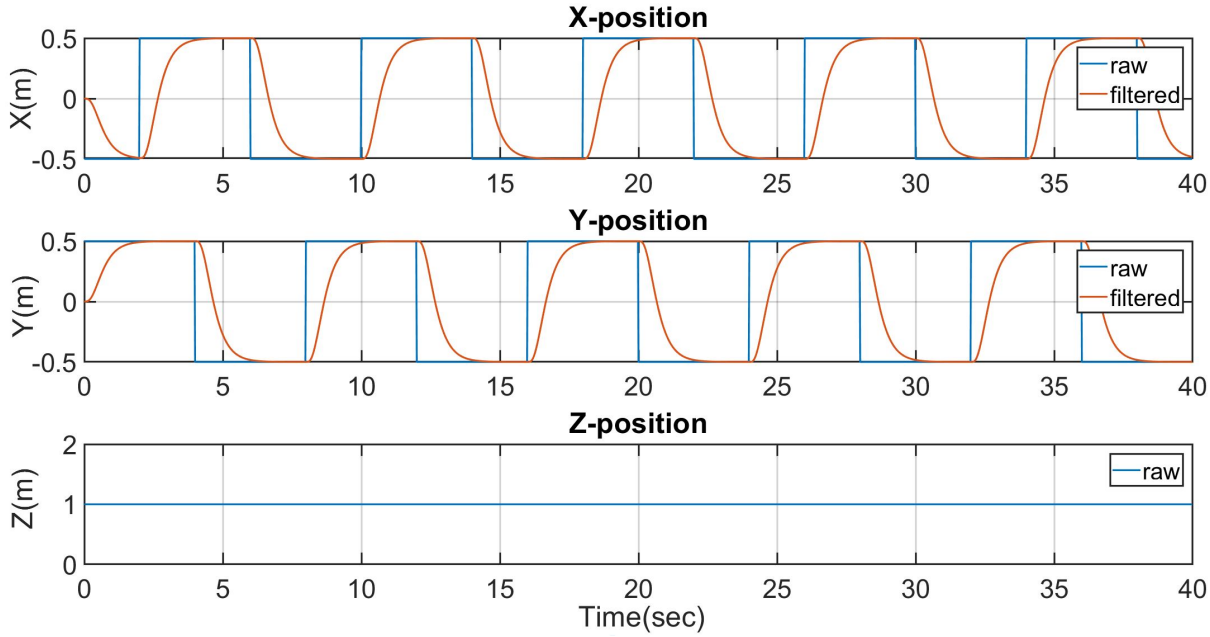

FIGURE 3.4: Square reference trajectory components in each axis

3 - Elliptical trajectory

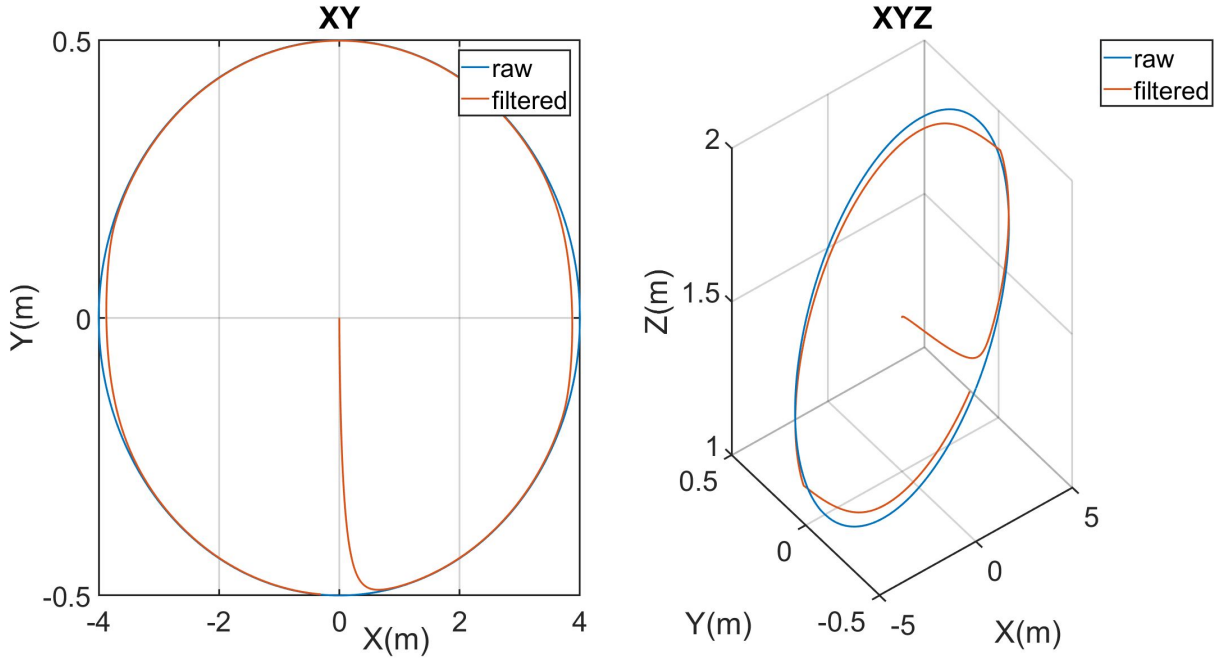

FIGURE 3.5: Square reference trajectory (left : top view, right : tilted side view)

The UAV is asked to follow an inclined elliptical trajectory with 0.8 meter semi-major axis and 0.4 semi-minor axis. The trajectory is inclined by 15 degrees. The altitude of the UAV is chosen to be 1.5 meters and the motion starts from the center of the UAV. The original and filtered trajectories can be observed in Fig.3.5 and Fig. 3.6.

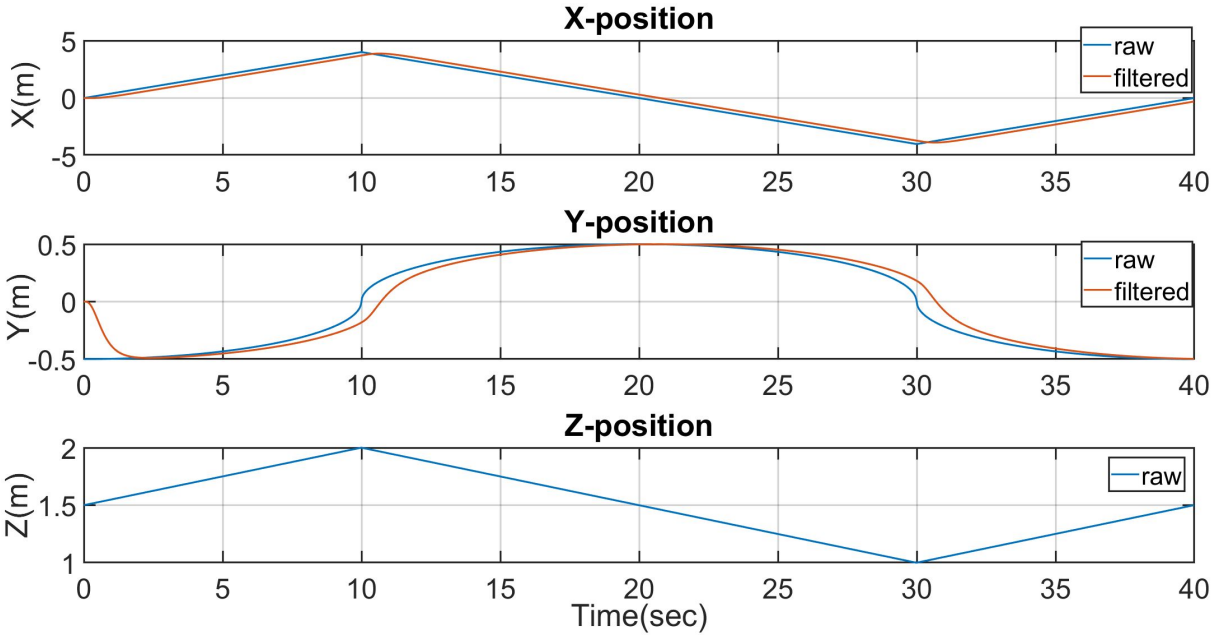

FIGURE 3.6: Square reference trajectory components in each axis

3.2.2 The inner feedback control

The UAV's embedded feedback controller is a combination of 3 controllers : a position, a speed and an attitude controller (see Fig. 3.7 [2]). The horizontal and vertical positions of the UAV are each controlled by the position and speed feedback controllers which in each axis compute the required attitude and total thrust (for horizontal and vertical movements, respectively) in order to follow the set-points of a given repetitive sequence. This information is then used by the attitude controller to calculate the necessary torque values from which a mixer matrix finally deduces the thrust contribution of each rotor.

2. The parameters with 'bar' notation are feedback measurments.

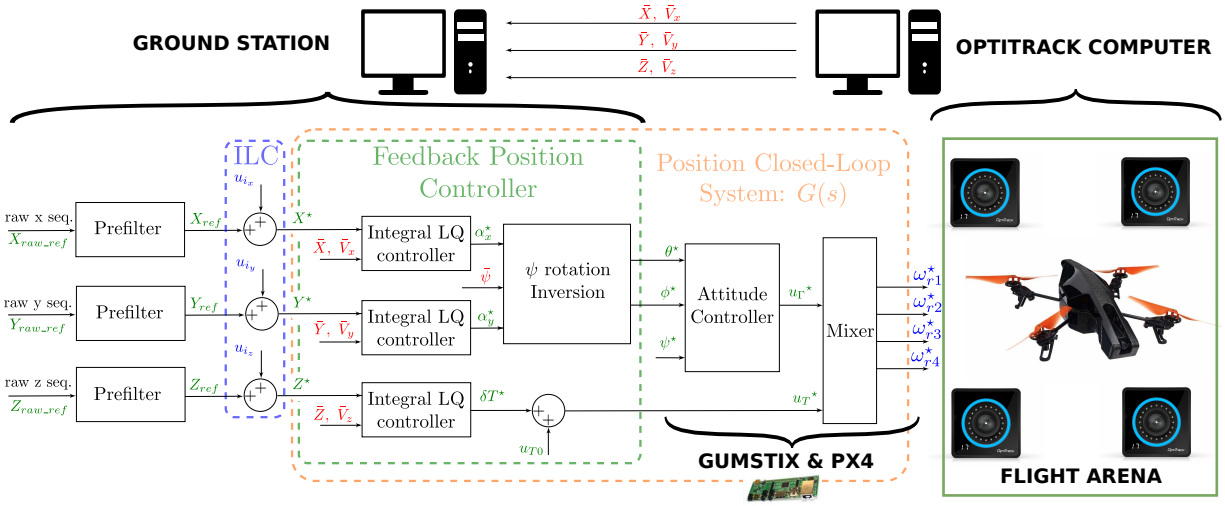

FIGURE 3.7: NO-ILC experimental setup and control architecture.

The designs of the feedback controllers are done utilizing an *LQ method* with the linearised model of the UAV considering the following control objective :

- An overshoot of $< 5\%$ for the position tracking (X, Y, Z),
- A limited command noise,
- Good robustness properties.

In Fig. 3.7, one can observe the architecture of the position closed-loop system. The *LQ* horizontal controllers deliver the attitude references $\alpha_x^\star$ and $\alpha_y^\star$ in the *OptiTrack* frame which are then converted in Euler angles references $\phi^\star$ and $\theta^\star$. The vertical *LQ* controller generates thrust reference $\delta T^\star$ to which a nominal thrust u_{T0} is added to compensate for the weight of the UAV. Furthermore, Euler angle references and thrust references are tracked by the *PX4* controllers which generate setpoints for the motors (ω_{ri}). The position and speed of the UAV are then measured by the optitrack system and used by the *LQ* controllers. Hence, the closed-loop system considered by ILC consists of the transfer functions between the position references $X^\star$, $Y^\star$ and $Z^\star$ that are assigned to position controllers and the measured position outputs $\bar{X}$, $\bar{Y}$ and $\bar{Z}$:

$$G_X(s) = \frac{\bar{X}(s)}{X^\star(s)},\ G_Y(s) = \frac{\bar{Y}(s)}{Y^\star(s)},\ G_Z(s) = \frac{\bar{Z}(s)}{Z^\star(s)}. \tag{3.2}$$

The main drawback of this feedback controller is, as illustrated later, its "slow" time response (see Fig. 3.10 and 3.11). It takes approximately 5.5 seconds to reach a reference position which introduces a delay when tracking fast varying trajectories. Therefore, the aim of the feedback controller is to maintain the robustness while the precision performance is left to ILC.

The initialisation of the ILC is done as demonstrated on Table 3.1. The values of ρ and λ are selected considering the two convergence conditions given in (2.100). The low value set for ρ indicates that a high convergence rate is aimed. Besides, identification algorithm is asked for continuous transfer functions of low complexity (3 poles and no zeros). It should also be taken in to account that the initial UAV state is 'hovering' at 1 meter altitude.

3.2.3 The ILC algorithm

The goal of using an ILC algorithm is to compensate for the weak performance of the embedded feedback controller such that the UAV can achieve high tracking performance for operator's repetitive sequences. The ILC method chosen for the control design is the norm optimal ILC (NO-ILC) given in Section 2.4.2.

TABLE 3.1: NO-ILC initialisation parameters

Sample time, T_s	0.02 sec.
Simulation time, T_{sim}	37.5 sec.
Discretisation method	$Tustin$
Identified models in X, Y and Z	$n_{poles} = 3\,;\, n_{zeros} = 0$
Initial states	$[0\ 0\ -1]^T \in \mathbb{R}^3$
Initial system input	$[x_{ref}\ y_{ref}\ z_{ref}] \in \mathbb{R}^{N\times 3}$
Initial ILC input	$[0\ 0\ 0] \in \mathbb{R}^{N\times 3}$
Number of ILC iterations	300
Weight on the error, $\mathbf{W}_e$	$\rho\mathbf{I} \in \mathbb{R}^{N\times 3}$
Weight on the system input, $\mathbf{W}_u$	$\mathbf{I} \in \mathbb{R}^{N\times 3}$
ρ	0.001
λ	0.1

3.3 Experimental setup

The flight experiments are carried out by using the presented cascade control architecture on UAV called *Parrot AR. Drone 2.0*. The place of the experimentation is an indoors flight arena equipped with an *OptiTrack* motion capture system. Figure 3.7 shows the experimental environment in a schematic way. The fast and robust attitude regulation loop is implemented in the *Pixhawk PX4* autopilot which computes motors' references to track attitude set-points. The guidance and navigation loops are implemented in a more powerful calculator, the *Gumstix*, yielding attitude references to the *Pixhawk* to track desired position trajectories. Moreover, the specific role of each component can be detailed as below :

1. Ground station : a Simulink model that can both send attitude reference and high level position reference to the *Gumstix* or execute its own guidance law (position feedback and ILC).
2. Wi-Fi Link : the data transfer between the quadrotor, the ILC computer and *OptiTrack* system is done via Wi-Fi.
3. Gumstix Board : the bridge between the data received via Wi-Fi and the *PX4* autopilot. The *Gusmtix* can run its own guidance law or can directly receive attitude commands coming from the ground station.

4. Pixhawk PX4 : joined to the *Gumstix* by a serial link, the *PX4* is the actual autopilot of the quadrotor, where attitude control takes place.

So far, the process is fairly standard.

3.3.1 Proposition of a new ILC data-flow

The main contribution of this chapter is presented in this section by introducing a new experimentation method for ILC. It is shown how one can perform experiments following a nontraditional update procedure for ILC. In the common practice, one uses the experimental data from the previous flight in order to obtain the input signal in the next flight which leads to one ILC update per one experiment. The problematic side of this issue comes when one needs to do large number of iterations to reach a desired reference. In order to overcome this problem, a new experimentation scheme can be applied by using a closed-loop system identification process together with an ILC algorithm. This method is provided in form of a data flow chart and can be viewed in Figure 3.8. The given procedure can be described as follows : The first flight includes no ILC signal, i.e. the UAV flies with its own internal closed-loop controllers that are adjusted to have low performance. The data from the first flight is used to identify linear models of the UAV dynamics along X, Y and Z-axes. Then, ILC makes 300 iterations using predicted positions of virtual incoming flights (note that the number of iterations can be chosen different than 300 depending on the needed accuracy and the speed of the simulation time). The signal from the last iteration is then used to modify the reference signal of the system and a new real flight is performed. Next, using the flight data from this new flight the UAV model is re-identified and then the simulated iterations are restarted. After applying the same steps for 3 times, this approach reaches its limit and the identification does not improve the trajectory further (note that the number of repetitions, i.e. 3 times, is determined by observing UAV tracking results). At this point, the update process is switched back to the traditional way where the ILC begins directly using the error from the real flights and this process yields one update per flight.

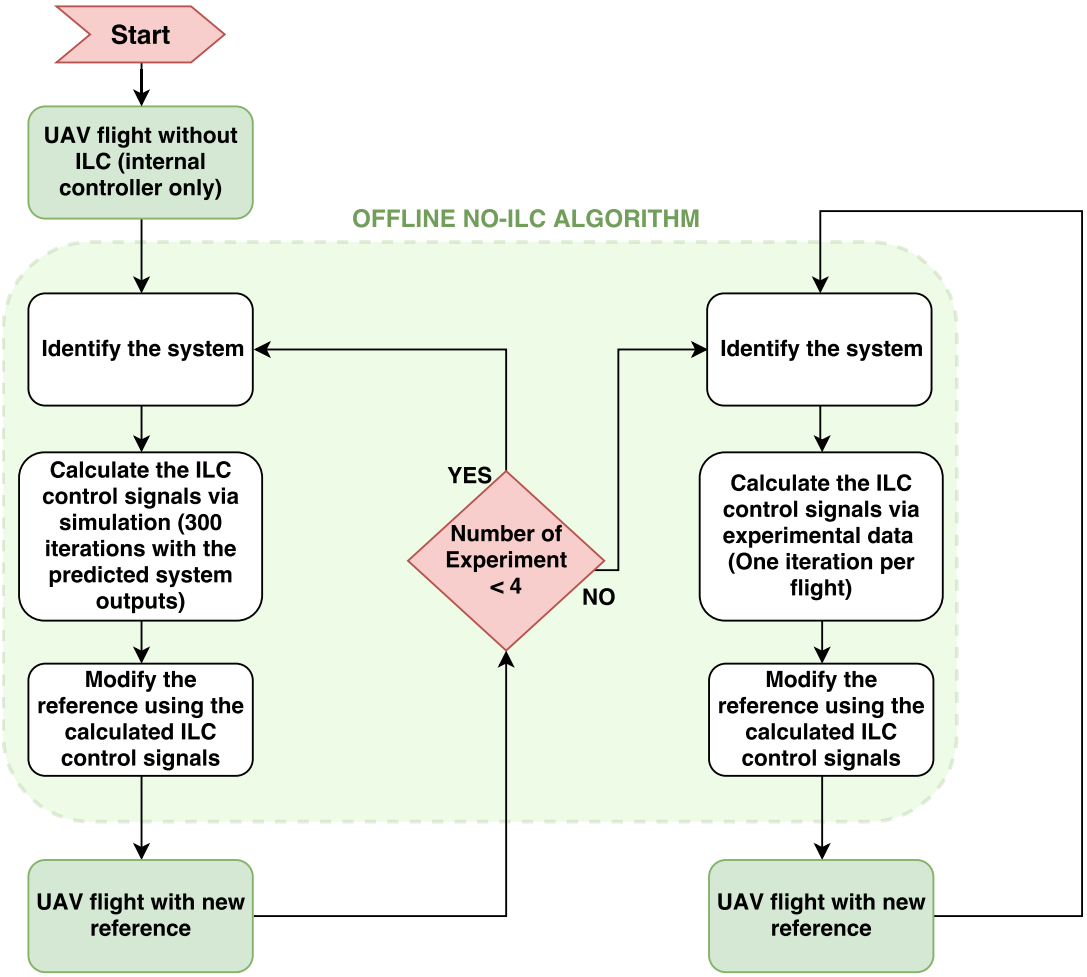

FIGURE 3.8: Data flow procedure for NO-ILC experiment

The proposed procedure can provide a rapid correction of closed-loop lags and errors from model imperfections by means of only simulated ILC iterations in several flights at the beginning. This is because the re-identification process of the linear UAV model after each flight allows us to obtain a close behaviour to the desired tracking. Basically, what ILC does is to refine the UAV linear model as of the first flight. One can think of each newly performed flight with the current ILC signal as a new operating point which is closer to the desired reference trajectory. Thus, the forthcoming ILC signal needs to include the information of the last operating point to better the tracking and this is the point where the re-identification is made use of. In this way, the simulation based ILC approach improves the tracking until the limits of the linear UAV model. Accordingly, it can be said that when ILC is switched back to the traditional update, the remaining errors are due to the non-repeating disturbances or very nonlinear behaviours occurring in real flight which cannot be approximated by the linear model.

3.4 Experimental results and analyses

The control architecture given in Section 3.2 is tested using three reference trajectories shown in Section 3.2.1. The flight experiments are conducted with respect to the data flow in Figure 3.8 and the initialisation on Table 3.1. Note that depending on the reference trajectory some readjustments are also applied in the simulations time and the initial UAV altitude.

3.4.1 Eight-shaped trajectory tracking results

For the eight-shaped trajectory, the experimentation was done until the 9^{th} flight considering the fact that the tracking precision was not improving further. The tracking results obtained with the NO-ILC algorithm are depicted in Figure 3.9. The ILC update for the first 4 flights are based on the predicted data obtained through the identified models of the system. On the other hand, the ILC used directly the real flight data for the remaining 5 flights. It can be observed that the reference tracking almost does not improve as of the 5^{th} flight. This is due to the fact that it was possible to reach the maximum tracking performance via ILC using the identified system models by the end of the fourth flight. Moreover, one can also see in Figure 3.9 how the mean position error on each axis changed along the flight experiments. It is straightforward to conclude that a sufficiently accurate tracking performance can be obtained in only 4 flight experiments.

On the identified transfer functions : *After each flight, we identify the UAV transfer functions on each axis. Since the translational speeds, and the attitude angles become more aggressive as the ILC improves the tracking, the identified linear transfer functions are slightly different after each flight. This simply highlights the utility of identifying a new transfer function after each flight. As shown in Figure 3.10, the identified transfer function of the X-axis evolves only for a small amount along the flights whereas the variations are relatively larger on the Y-axis (see Figure 3.11). This can be found reasonable since it is harder to follow the trajectory on the Y-axis that is of a frequency twice as the one on X-axis. Moreover, for more complex trajectories, it could be more suitable to choose higher-order transfer functions to identify in order to better fit the nonlinear behaviour. For the sake of practicality, one can go further by performing an 'automatic order selection' of these transfer functions based on the model fit percentage or on the prediction error for the real flight.*

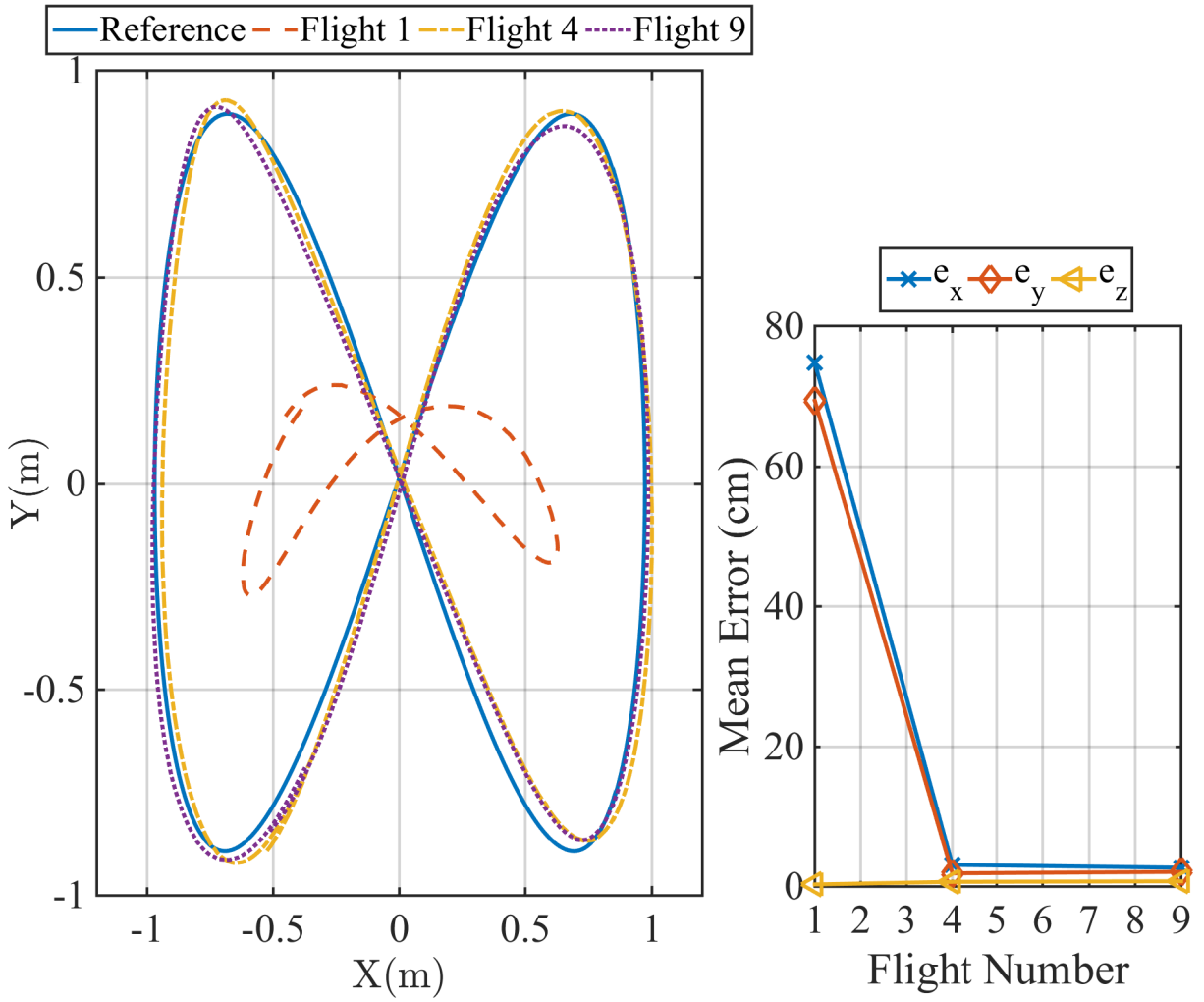

FIGURE 3.9: Tracking results for the eight-shaped reference

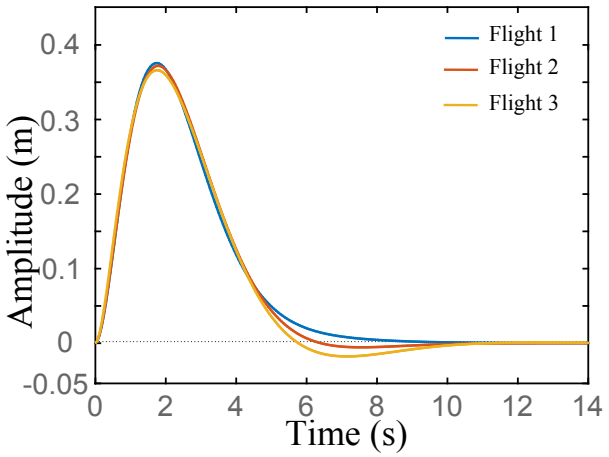

FIGURE 3.10: Evolution of the impulse response of the identified transfer function of the X-axis with the flights.

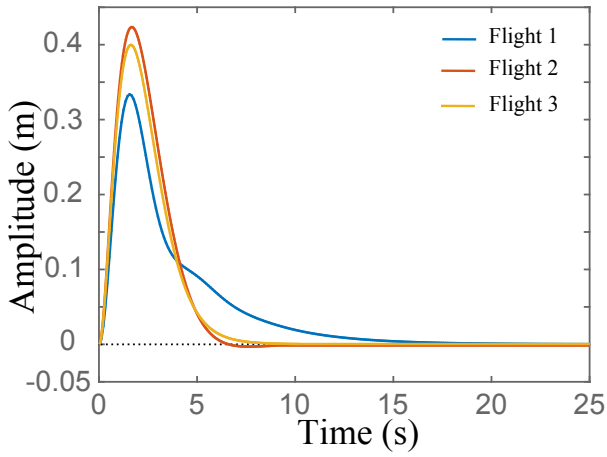

FIGURE 3.11: Evolution of the impulse response of the identified transfer function of the Y-axis with the flights.

3.4.2 Amplitude analysis for eight-shaped trajectory tracking

After obtaining a satisfactory tracking performance for an eight-shaped trajectory, an amplitute analysis was performed in order to find the limitation of the used ILC algorithm. In different words, the same procedure was applied using the reference trajectories with higher amplitudes but the same frequency as before. In this process, a full nonlinear simulator of the UAV (including saturations, aerodynamics effects etc.) was used instead of carrying out real flight experiments. Three simulations were shown for each amplitude considering the fact that the simulator output became invariant after three flight simulations. The results of this amplitude analysis were given on Table 3.2 in terms of the absolute values of the mean and maximum position errors on each axis. Looking at the error values in general, one may observe that increments in reference amplitude cause larger mean and maximum errors. Until $A_4 = 1.75m$, the errors on Z-axis show a different fashion along the runs; i.e. they increase whereas the errors on X and Y-axes decrease as aimed. Then, at $A_5 = 2.00m$, the mean errors on X and Y-axes first decrease and then slightly increase while the errors on Z-axis keep the same trend. The issue observed on Z-axis can be seen as an outcome of higher pitch and roll angles that were taken by the UAV for improving the accuracy of the lateral trajectory which accordingly caused a compromise of altitude accuracy. Hence, one can say that the ILC algorithm for the eight-shaped trajectory starts to become unsatisfactory as of $A_4 = 1.75m$.

TABLE 3.2: Mean position error vs. Reference amplitude (in cm)

Frequency, $f = 0.1333$ **Hz.**								
Reference Amplitude	**Mean error**	**Run 1**	**Run 2**	**Run 3**	**Max. error**	**Run 1**	**Run 2**	**Run 3**
$A_1 = 1.00$	$\bar{e}_x$	81.98	3.366	3.400	e_{xmax}	128.6	6.192	5.683
	$\bar{e}_y$	80.29	3.504	1.292	$e_{y_{max}}$	126.3	7.831	3.363
	$\bar{e}_z$	0.091	1.235	1.357	e_{zmax}	0.181	2.127	2.363
$A_2 = 1.25$	$\bar{e}_x$	102.7	4.082	4.084	e_{xmax}	161.3	7.687	6.985
	$\bar{e}_y$	100.3	5.114	1.773	$e_{y_{max}}$	157.8	11.14	4.770
	$\bar{e}_z$	0.142	2.001	2.260	e_{zmax}	0.284	3.435	3.864
$A_3 = 1.50$	$\bar{e}_x$	123.4	4.757	4.678	e_{xmax}	193.8	9.166	8.150
	$\bar{e}_y$	120.2	6.899	2.363	$e_{y_{max}}$	189.3	14.69	6.361
	$\bar{e}_z$	0.204	3.039	3.582	e_{zmax}	0.409	5.220	6.062
$A_4 = 1.75$	$\bar{e}_x$	144.3	5.399	5.140	e_{xmax}	226.4	10.67	9.455
	$\bar{e}_y$	140.1	8.819	3.122	$e_{y_{max}}$	220.7	18.34	7.897
	$\bar{e}_z$	0.277	4.405	5.244	e_{zmax}	0.557	7.669	9.288
$A_5 = 2.00$	$\bar{e}_x$	165.2	5.907	8.713	e_{xmax}	259.0	12.49	14.91
	$\bar{e}_y$	160.0	10.26	15.26	$e_{y_{max}}$	252.2	21.24	32.30
	$\bar{e}_z$	0.362	6.503	14.27	e_{zmax}	0.729	11.41	27.33

The unwanted results after a certain point are most probably due to nonlinearities

or nonrepetitive elements such as noise and uncertainties in identified models which are used within ILC. A solution to these issues can come from an improvement in the method by improving the proposed scheme in a way that it can be robust to uncertainties and nonlinearities while keeping the tracking precision still as high as possible (see Chapter 5 for such approach).

3.4.3 Square-shaped trajectory

A square-shaped trajectory is much harder to follow with precision compared to the eight-shaped trajectory if the UAV is required to fly continuously without stopping at the corners of the square. In different words, at the corners of the square the UAV is asked to perform a sudden change in its direction of movement which is a strenuous control task. Moreover, a prefiltering process similar to the one in Section III-B was also implemented on the square trajectory before beginning the ILC experimentation. Thus, the reference trajectory was made more real-like from the point of UAV dynamics and the thust level could be kept below the limit of saturation.

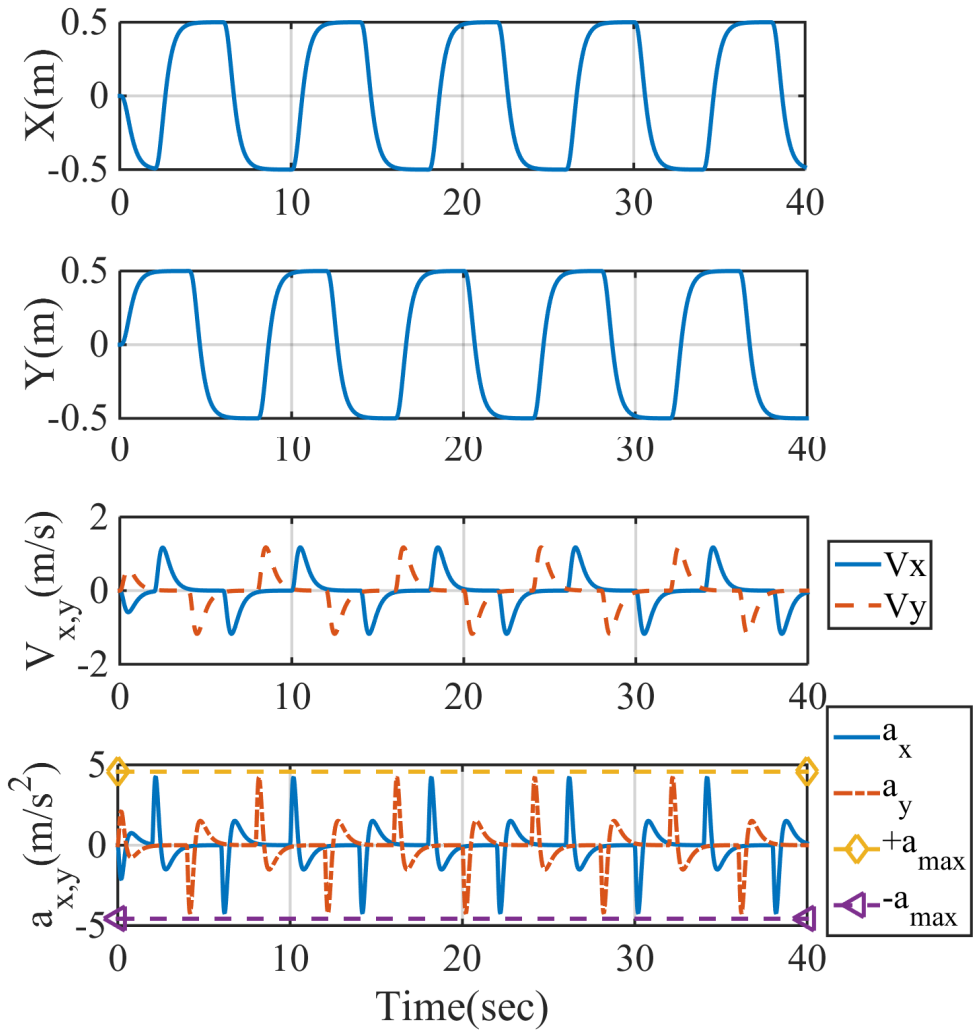

FIGURE 3.12: Filtered square reference trajectory (V : speed, a : acceleration)

Figure 3.12 depicts the filtered square reference positions for X and Y-axes as well as the corresponding velocity and acceleration curves. One can observe that the demanded acceleration from the UAV was kept within a range defined by a limit of ±25 degrees in pitch and roll angles. In order to avoid saturations at sharp corner turns, this limit was made more strict compared to the one set for the eight-shaped reference, i.e ±35 degrees.

The square trajectory has a period of 8 seconds and it was repeated for 40 seconds which corresponds to 5 loops. Figure 3.13 demonstrates the evolution of the reference tracking according to the predefined data flow procedure and the plotted data represents only the fifth loop of the UAV's trajectory. It can be figured out that the most of the improvement in the tracking was obtained during the first 4 flights where the ILC algorithm used the predicted flight data. The rest of the flights were carried out with the real flight data using the traditional ILC update process and the precision of the tracking could be slightly improved.

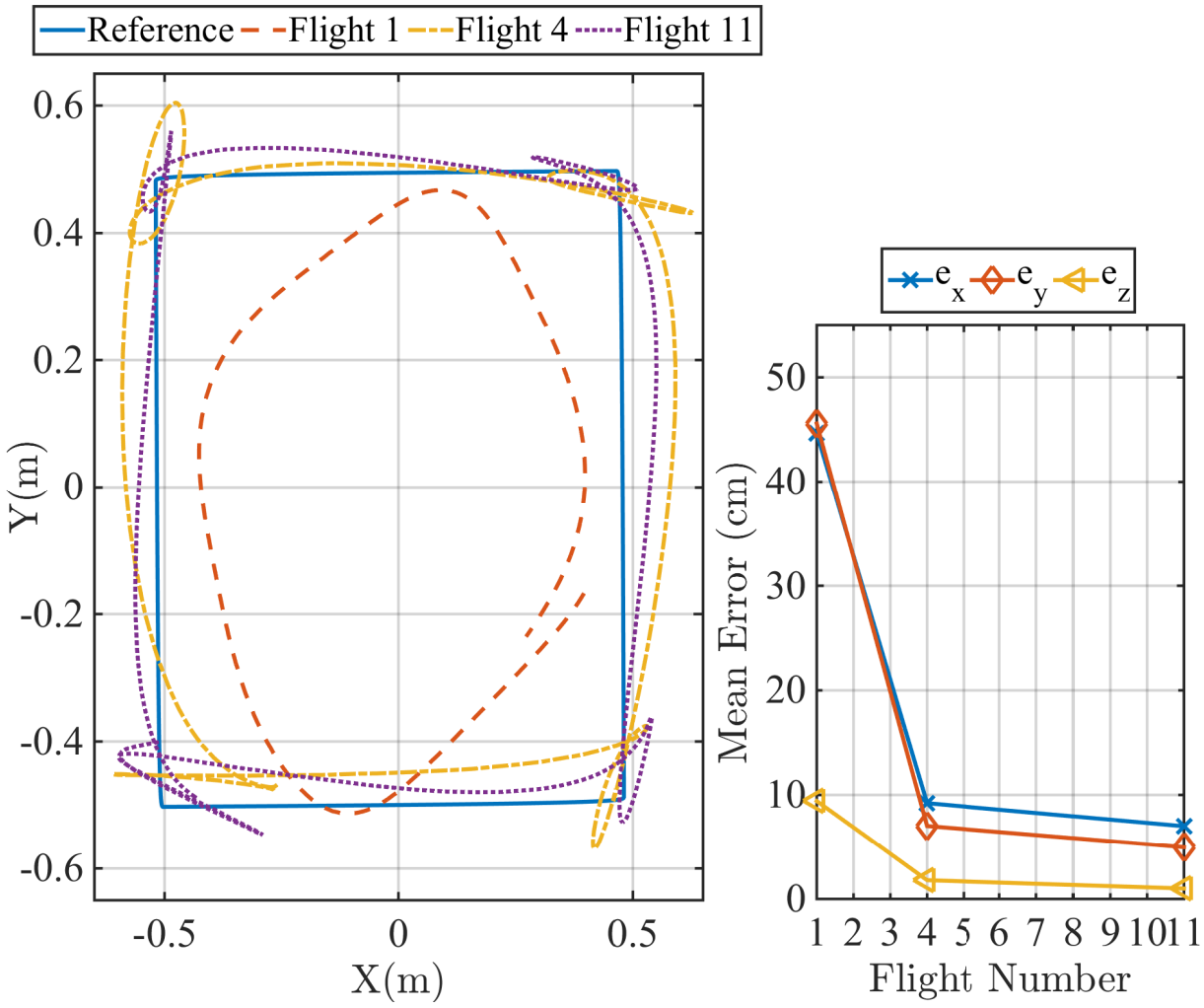

FIGURE 3.13: Tracking results for the square-shaped reference

3.4.4 Elliptical inclined trajectory

The final trajectory that was analysed is an ellipse with 0.8 meter semi-major axis, 0.4 meter semi-minor axis and 15 degrees inclination. The altitude of the UAV was initially set to 1.5 meters and the UAV starts its flight from the center of the ellipse. Furthermore, the period of the ellipse and the flight duration were chosen to be 4 seconds and 20 second, respectively.

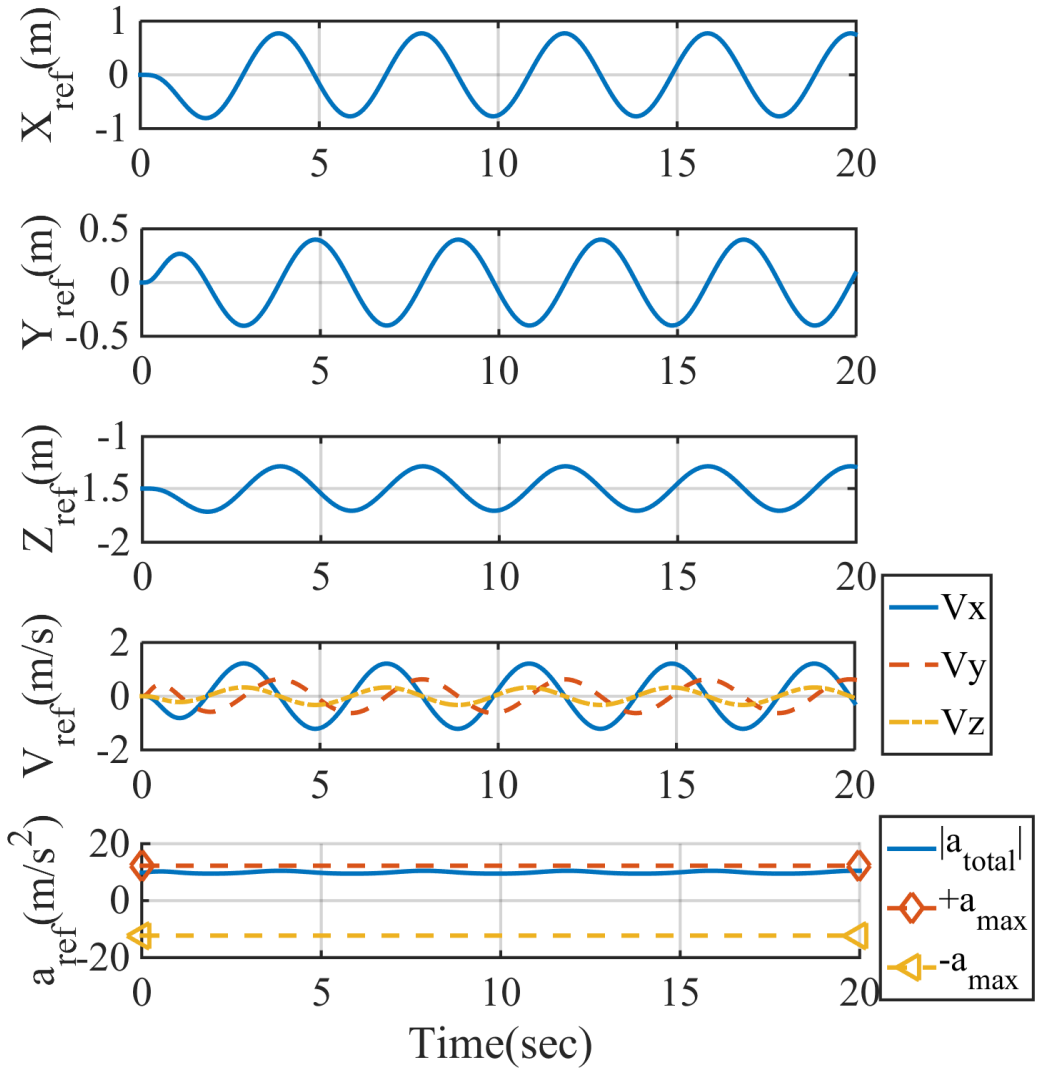

FIGURE 3.14: Filtered elliptical reference trajectory

Due to the inclination of the reference, the UAV needs to allocate some of its total allowable thrust for altitude increment which limits the reachable accelerations for the maneuvers on the lateral plane. Accordingly, the reference was filtered in order to be sure that the demanded maneuvres stay below the saturation limits. Figure 3.14 demonstrates the positions, velocities and accelerations for the filtered reference trajectory where one can also observe that the norm of the vector compound of the accelerations on each axis remains between a certain acceleration range. The absolute value of the maximum allowable acceleration of the UAV is 1.25 times the gravitational acceleration, i.e. $12.26 m/s$. The evolution of the reference tracking is demonstrated in Figure 3.15. The experiments were

carried out according to the same data flow procedure used before and the represented data corresponds to the 5th loop of the trajectory. Similar to the results of other reference trajectories, the largest improvement in tracking was seen to be during the first four flights where the ILC used predicted data. Moreover, the seven flights that were performed afterwards using traditional update could only reduce a small amount of the remaining error.

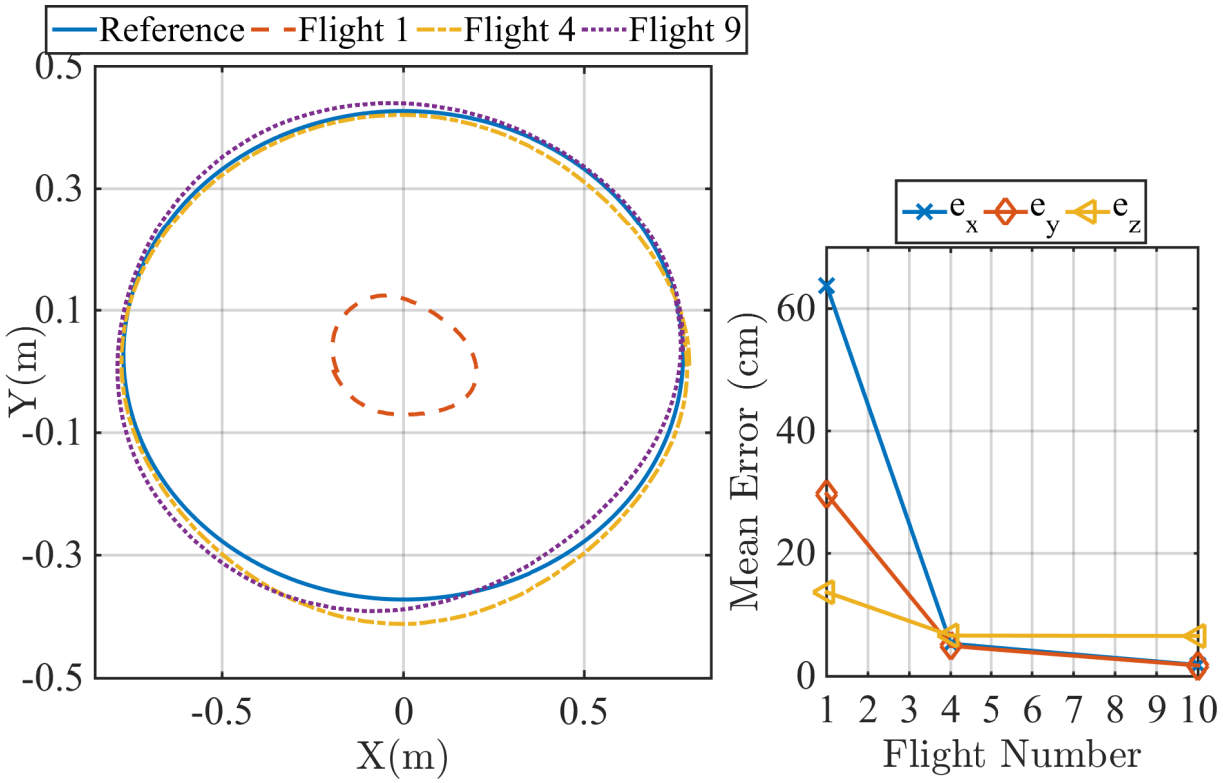

FIGURE 3.15: Tracking results for the elliptical reference (V : speed, a : acceleration)

3.5 Conclusion

This chapter has presented the application of norm optimal ILC (NO-ILC) on a real indoor UAV. The efficiency of NO-ILC has been proven with good tracking results during the experiments. The agility of the UAV has been used to its limit via NO-ILC and a large amount of precision has been achieved in only four flights. During this application, NO-ILC has been integrated in open-loop to an already existing UAV system that is controlled via feedback system. This has demostrated that ILC can be used as tool for increasing the performance of a real nonlinear system as a UAV.

The main contribution of the chapter can be seen as a practical procedure proposed for reducing the time spent during ILC experiments. This procedure is based on the idea of using a hybrid ILC update, i.e. integrating large amount of predicted flight data while performing the ILC updates instead of directly using real data (see Section 3.3.1). The proposed procedure has proved to be efficient despite the low order linear UAV models used in the update process (identified models have been used in generating predicted

flight data and thus improving the system input without requiring real flights). The flight experiments with three different trajectories demonstrated that, for the given specific flight experiments, it was possible to acquire a large amount of tracking performance in only 4 flight experiments.

One of the challenges observed during the experiments can be seen as the increased nonlinearity and parametric uncertainty in the UAV's dynamics. These two adverse effects had an impact on the learning of ILC as the UAV was reaching closer and closer to the given agile reference. An improvement for resolving this issue can be done first by trying to use higher-order identified transfer functions during the simulated ILC updates. Another improvement can come from changing the UAV and apply the same experiment. This has sense since the UAV used during the experiments had limited agility. Finally, the ILC method that was used can be changed or a different tuning can be applied. If the measurement noise and disturbances can be maintained well, some D-type or PD-type ILCs may also be tried as better candidates. However, it would be suggested to rather choose a more sophisticated optimasation approach inside the NO-ILC (the one used in the application is based on minimisation via Lagrange multiplier technique which is analytical).

It is also possible to address the issue of reduced learning after several flights due to nonlinearities by trying the switch to a nonlinear ILC method from linear ILC (i.e. the one used during experiments). It may improve convergence speed ; yet, it is generally required that there is some well-known information about these nonlinearities (or uncertainties, disturbances etc.).

Another point that is worth to mention is related to the offline nature of the ILC. The term offline signifies here that after each real flight trajectory tracking, the UAV has to stop flying (or wait in hovering position) and wait for ILC to compute the next system input. This is quite normal since ILC is a method developped for the need of improving batch processes, i.e. discontinuous processes (e.g. reference tracking during a pick-and-place of robotic arm). However, it can be quite useful for a UAV to be able to carry out an online computation during its flight. In other words, the ILC signal for the next flight can tried to be calculated while the UAV is performing the current flight. Of course, some limitations for this type of application can be the computational speed, the communication speed and the length of the simulation (flight trajectory).

Finally, the results of the presented UAV application has demonstrated how an ILC that is in open-loop can improve a system with an inner feedback controller. This has been a motivation in continuing the research further and developing a way of using ILC to build linear feedback controllers. Thus, the following chapter is focusing on using ILC as a tool for automatically tuning feedback controllers (it can also be seen as transforming open-loop ILC to a close feedback controller).

Chapter 4

Learning Based Controller Tuning

Contents

The *internal model principle (IMP)* based state feedback control is an intuitive and a well established method for controlling dynamical systems. One can find it quite natural to build controllers based on the states of a dynamical system since the state variables can be used to anticipate the future behaviour of the system [13]. Accordingly, it is not a suprise that the state feedback method has been referred to since 70s when analysing the disturbance rejection problems [16]. In order to lessen the effect of disturbance on the system, one can simply augment the state-space of the nominal system with the model of the disturbance (i.e. IMP) and then apply a non-learning type feedback method. Although there are many non-learning-type control methods developed to handle system disturbances such as the ones given in Section 1.1.1, the simplest one among all can be considered as the *pole (or eigenvalue) placement method* that uses state feedback through a linear controller structure. This approach is quite useful when dealing with periodic disturbances for which the structure is known. Some examples to these type of disturbances can be seen as a step signal, a sinusoidal signal at some specific frequency or a combination of several sinusoidal signals with different frequencies. Since the dynamics of such disturbances has periodic behaviours, they have a known structure and can be eliminated if their structure is utilised in the design of the controller.

From the point of view of repetitive disturbance rejection, it would be quite natural to expect some similarity between the state state feedback via IMP (pole-placement in this case) and the ILC. The difference of ILC, however, is that it is a data-driven open-loop method based on iterative feedforward control calculated via error filtering. ILC can successfully vanish the repetitive errors that occur between the system runs (iterations) and it has been traditionally used to "optimise" the tracking performance of repetitive trajectories for over a decade. As suggested by [121], one can obtain high trajectory tracking performance for motion control systems via feedforward control and in the case of a system that repeats the same task, the ILC can be used as the feedforward method to increase

the system performance. It should be noted that the ILC performance is highly dependent on the type of the system disturbance. Since ILC is based on remembering the errors from the previous runs and compensating for the repeating errors, it becomes a powerful approach when dealing with repetitive disturbances. Accordingly, the ILC performance can significantly reduce when the system is subject to non-repetitive disturbances simply because ILC will also remember the non-repeating error from the previous run. Due to this drawback, ILC is usually applied on a pre-designed closed-loop system which compensates for the non-repetitive characteristics of the system. Despite this fact, ILC applications can bee observed in many fields (see Section 1.2.2).

In this chapter, it is proposed that ILC can be used as a tool to provide an easier solution for obtaining the proper parameters of a linear state feedback controller since ILC requires no disturbance modelling and it does not get affected by the complexity of the structure of the repetitive disturbance. The given analysis demonstrate first the comparison of ILC and the pole-placement method in case of unknown complex periodic disturbances. Then the proposition is given in the form of a design workflow. The overall result corresponds to an *automatic tuning* of linear state feedback controllers.

4.1 Comparison of ILC to IMP

One of the simplest examples for a disturbance rejection controller based on IMP can be seen as the state feedback controller designed via augmenting the closed loop system dynamics with the model of the periodic disturbance, namely an *augmented state feedback (ASFB) controller*. As this method is purely based on the exact knowledge of the disturbance signal's frequencies, it is capable of fully rejecting the periodic disturbance asymptotically depending on the accuracy of the disturbance model. However, the efficiency of this approach reduces with the increasing complexity of the disturbance signal (e.g. nonlinearities and uncertainties) as well as with the amount of precision obtained during disturbance measurement and estimation processes.

In order to better assess the similarities between ILC and ASFB, a simulation model is created as demonstrated in Figure 4.1. One can simply see that the simulation model consists of three main blocks : the ILC, the state augmentation and the plant (from left to right). The plant used in the simulation model is the one given in Section 2.5.2, i.e.

$$\dot{x}(t) = \mathbf{A}x(t) + Bu(t), \tag{4.1}$$
$$y(t) = \mathbf{C}x(t) + Du(t). \tag{4.2}$$

where $x(t), y(t) \in \mathbb{R}^2$, $u(t) \in \mathbb{R}^1$ and

$$\mathbf{A} = \begin{bmatrix} -3.5014 & -3.0003 \\ 1 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

The system (4.1)-(4.2) is discretised on *MATLAB* via *Tustin method* with a sampling time of $T_s = 0.01s$). The periodic disturbance $d(t)$ (depicted with a red block) is acting on the

second output of the plant and it is in the sinusoidal form, i.e $d(t) = a \cdot \sin(wt)$ where $a = 0.2$ and $w = 1rad/s$ are the amplitude and the pulsation of the signal, respectively. Morover, a switch mechanism separates the simulation process of ILC and ASFB (see Figure 4.1). Thus, two simulations are performed disjointly. Further details regarding each simulation is provided below.

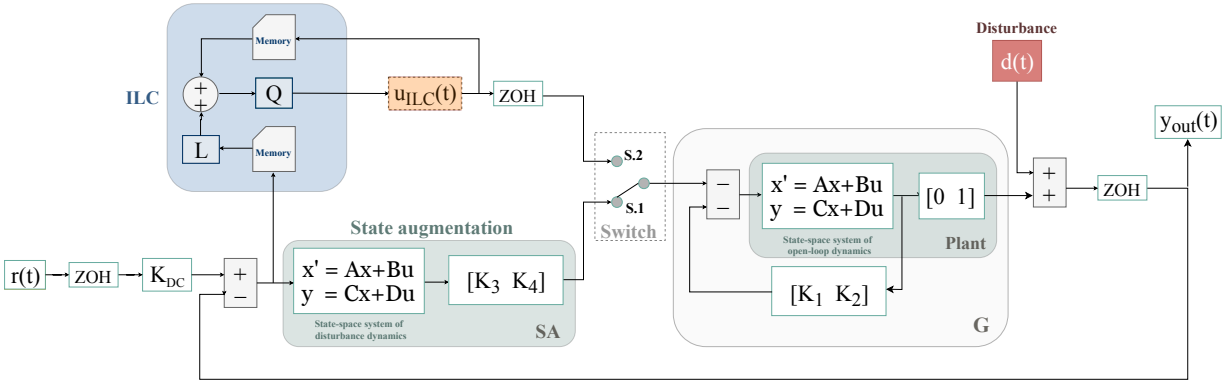

FIGURE 4.1: Simulation model

1) Simulation with ASFB :

According to IMP, the model of the repetitive disturbance $d(t)$ should be used to augment the state-space of the plant dynamics. Therefore, we define an output $\tilde{z}(s)$ by considering the *Laplace transform* of $d(t)$ with the system reference $r(t)$ and the system output $y(t)$:

$$\tilde{z}(s) = \frac{aw}{s^2 + w^2}(\tilde{r}(s) - \tilde{y}(s)), \tag{4.3}$$

for which the state-space is written as :

$$\begin{bmatrix} \ddot{z}(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} 0 & -w^2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{z}(t) \\ z(t) \end{bmatrix} + \begin{bmatrix} aw \\ 0 \end{bmatrix} \epsilon(t) \tag{4.4}$$

where $\epsilon(t) = r(t) - y(t)$ is the error. Next, one can use (4.4) to augment the system $(\mathbf{A}, B, \mathbf{C}, D)$ given by (4.1)-(4.2) which will lead to (4.5). Since the disturbance enters the system at the second output, $C_2 = [0\ 1]$ and the augmented system is thus built as follows :

$$\begin{bmatrix} \dot{x}(t) \\ \ddot{z}(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & 0_{2\times1} & 0_{2\times1} \\ -C_2 aw & 0 & -w^2 \\ 0_{1\times2} & 1 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ \dot{z}(t) \\ z(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ aw \\ 0 \end{bmatrix} r(t). \tag{4.5}$$

At this point, a state feedback method should be chosen in order to calculate the closed-loop gains that will provide good rejection performance. A simple classical choice for this purpose is to apply the *Pole Placement* method with the control law $u = -Kx_{aug}(t)$ to move the disturbed system's poles to the desired locations that will produce the required tracking and disturbance rejection ($x_{aug}(t) = [x(t)\ \dot{z}(t)\ z(t)]^T$ is the augmented state

vector). Mathematically, the control objective of the simulation is $r(t) = 0$ where $r(t)$ is the system reference. A solution to the given rejection problem can obtained by selecting new pole locations utilising the Bessel poles chart. For a 4th order system and a settling time of $1s$, one should choose the poles as : $p_{1,2} = -4.016 \pm 5.072i$ and $p_{3,4} = -5.528 \pm 1.655i$. Then, the final step of pole placement is the calculation of the gain vector K_{aug} that will move the system poles to the new ones. Finally, the result of the simulation with the built ASFB controller can be analysed in Figure 4.2 (denoted by the legends 'ASFB'). It can be easily observed that ASFB is capable of perfectly rejecting the sinusoidal disturbance in about $1.5s$.

2) Simulation with ILC

After observing the performance of the ASFB controller, the switch '$S.2$' in the simulation model of Figure 4.1 is set $'ON'$ for the ILC simulation. The ILC algorithm used during the simulation is the same as in Section 2.5.2. The disturbance parameters remain the same as before ($a = 0.2$, $w = 1rad/s$) and the initialization of the ILC is performed with the parameters shown on Table 4.1 where ρ, λ and M are selected by considering the effect of each in the optimisation. The calculation of the ILC input u_{ILC} is performed 'off-line' and updated after each run of the model. For having a better insight of the tuning process, consider the objective function (2.80) and its derivative (2.87), i.e.

$$\begin{aligned}\min_{u_{i+1}(t)} J(u_{i+1}) = \quad & \min_{u_{i+1}(t)} e_{i+1}^T(t)\mathbf{W}_e e_{i+1}(t) + u_{i+1}^T(t)\mathbf{W}_u u_{i+1}(t) \\ & + \lambda[(u_{i+1}(t) - u_i(t))^T(u_{i+1}(t) - u_i(t)) - \delta] \\ \frac{\partial J_{i+1}}{\partial u_{i+1}(t)} = & -\mathbf{P}^T\mathbf{W}_e e_{i+1}(t) + 2\mathbf{W}_u u_{i+1}(t) + 2\lambda(u_{i+1}(t) - u_i(t))\end{aligned}$$

In these equations, the parameter ρ is the weight on the system input $u_{i+1}(t)$ and the parameter λ is the weight on the input increment $u_{i+1}(t) - u_i(t)$. Since NO-ILC approach is a minimisation process, the goal is to find the input $u_{i+1}(t)$ that will make $\partial J_{i+1}/\partial u_{i+1}(t)$ reach zero iteratively. This derivative can be rewritten considering $\mathbf{W}_u = \rho I$, $\mathbf{W}_e = I$ and $u_{i+1}(t) = \mathbf{Q}u_i(t) + \mathbf{LQ}e_i(t)$ as

$$\frac{\partial J_{i+1}}{\partial u_{i+1}(t)} = -\mathbf{P}^T e_{i+1}(t) + (2\rho + 1)\mathbf{LQ}e_i(t) + 2[\rho\mathbf{Q} + \lambda\mathbf{Q} - 2\lambda]\mathbf{Q}u_i(t) \tag{4.6}$$

Thus, choosing a low value for ρ while keeping λ much higher with a proper $\mathbf{Q}$ filter is a reasonable choise for faster arrival to the minimum. However, for more precise results one should look at the convergence conditions given by (2.100), i.e.

$$\begin{aligned}||\mathbf{Q}||_2 &< 1 \\ ||\mathbf{QL}||_2 &\leq \frac{1}{2\sqrt{\lambda + \rho}}\end{aligned}$$

which are obtained by looking at the largest singular values of $\mathbf{Q}$ and $\mathbf{QL}$ in (2.99). Moreover, the parameter M, which is the number of iterations, depends on how fast ρ and λ make the system converge as well as how much the error needs to be reduced by the

designer.

TABLE 4.1. NO-ILC initialisation

Sample time, T_s	0.01 sec.
Simulation time, T_{sim}	50 sec.
Initial states, x	$[0\ 0]^T \in \mathbb{R}^2$
Initial ILC input, u_{ILC}	0
Number of ILC iterations, M	1000
Weight on the error, $\mathbf{W}_e$	$\rho\mathbf{I} \in \mathbb{R}^{N\times N}$
Weight on the system input, $\mathbf{W}_u$	$I \in \mathbb{R}^{N\times N}$
ρ	0.001
λ	0.1

It is expected from ILC to converge to the control signal of the ASFB as the iterations continue which, theoretically, is a rather normal expectation since for a linear system there exists only one single solution. After 1000 system iterations, the converged ILC signals can be seen as in Figure 4.2. An interesting similarity between the ILC and ASFB can be noticed as their asymptotic rejection performances yield quite close results. The only difference, however, is that ILC is capable of handling the transients that are inevitable for the ASFB controller at the beginning of the simulation. This is simply due to the fact that ILC can detect the repeating error patterns between system runs and anticipate for them whereas ASFB needs to initialise itself by a first convergence to *synchronise* the phase of its internal model with the one of the disturbance. Furthermore, ASFB controller can completely reject the disturbance making the output exactly zero while the results with ILC still have some remaining oscillations that are of magnitude less than 0.00074 (These residual oscillations can be considered insignificant since their magnitude is 0.37% of the original disturbed output). The amount of disturbance attenuation reached by ILC in the steady-state (see Figure 4.2 (zoom)) is generally related to the choices made for its initialisation parameters such as M and ρ and λ.

Having seen the rejection performances of both methods, a structured comparison between them can be presented for a more clarified understanding. In this regard, Table 4.2 combines both the advantages and disadvantages of ILC and ASFB in a comparative manner. The given comparison is rather encouraging for having a further look at the benefits of the data-based ILC with respect to the systems going under unknown complex periodic disturbances. The ILC properties such as model independence, high precision and anticipation can make it easier to achieve the high performance requirements of many industrial applications where an approach based on IMP can hardly provide the required system performance. Nevertheless, one should not either discard the usefulness of already existing information regarding disturbance and system models. Although replacing an existing feedback controller with ILC can provide better results in some cases, it can be much more beneficial in most cases to use ILC as an integration to the closed-loop system having its embedded controller. This is due to the fact that both ILC and embedded feedback controller can compensate for each other's deficiencies which may eventually lead to much higher performances.

TABLE 4.2. A comparison between iterative learning control (ILC) and augmented state feedback (ASFB) control

ILC	ASFB
Based on repetitive data	Based on disturbance knowledge
Approximate results	Exact results
Free from disturbance complexity	Dependent on disturbance complexity
Iterative result	Constant result
Anticipation for transients	Unchanging transients
Small oscillations on steady-state	Full rejection on steady-state
Open-loop strategy	Closed-loop strategy
Higher precision	Precision within feedback bandwidth
Better for unknown complex disturbances	Better for simple known disturbances
Suffers from non-repeating elements	Deals with non-repeating elements

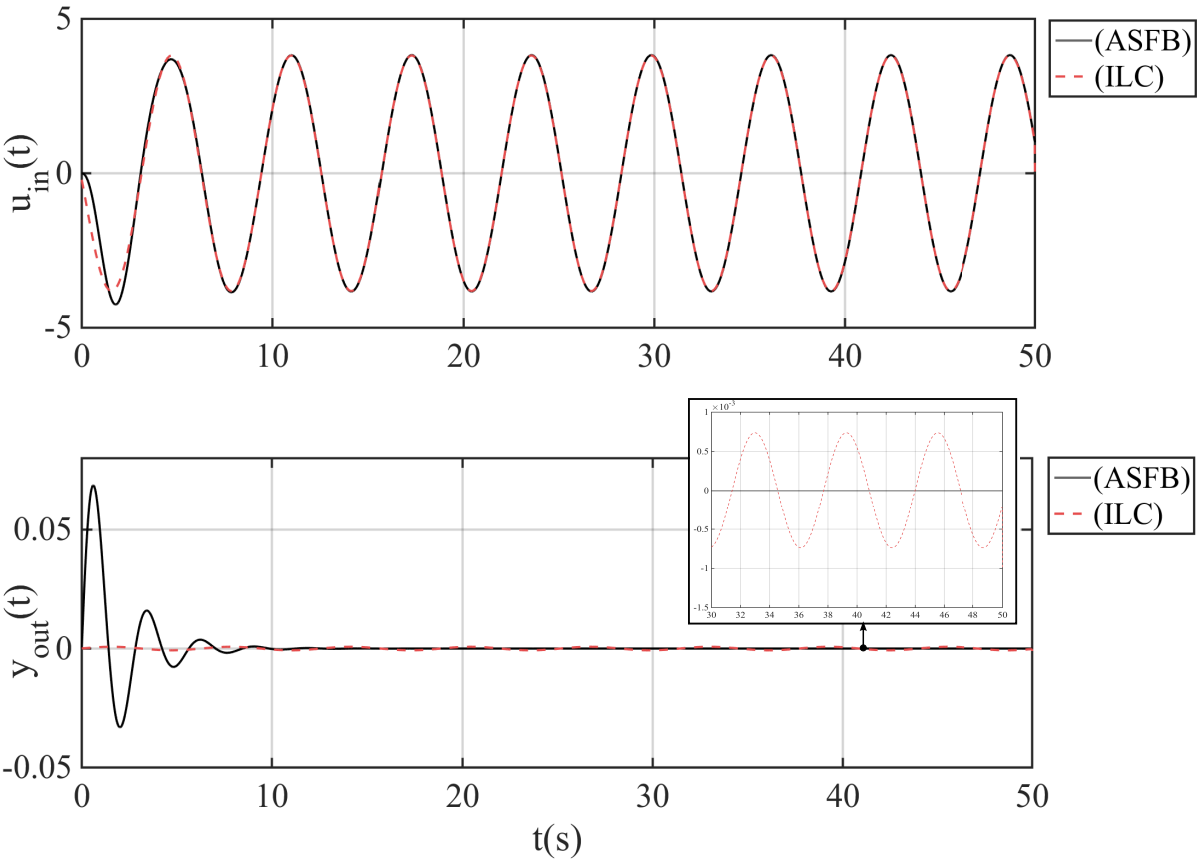

FIGURE 4.2: Augmented state feedback system vs. ILC system

4.2 Learning Based Controller Tuning (LBCT)

As a result of its useful properties, ILC can be utilised as an automatic tuning tool for linear controllers. By its open-loop integration to the system, the frequency spectrum and amplitudes of unknown periodic disturbances acting on a system can quickly be learned. This means that the modeling phase (or measurement/estimation) of designing a periodic

disturbance rejection controller (which is in general a tedious process) can be skipped and replaced by highly accurate approximations that are difficult to acquire without approaches based on repetitive data. Such a tool for automatic tuning is proposed below as a workflow named ***L**earning **B**ased **C**ontroller **T**uning (**LBCT**)* which can be seen as a procedure combining ILC with model identification.

4.2.1 Description of the LBCT workflow

The input of the LBCT workflow (see Workflow 1) is a stable system P under unmodelled/unknown disturbances and the output of LBCT is a transfer function containing the approximation of the disturbance spectrum. Thus, the main aim of the proposed workflow can be seen as to provide a guiding function from which a linear controller can be built. In other words, the whole procedure can be boiled down to automatically finding correct tuning parameters of a linear controller such that the periodic disturbances are attenuated (note that the 'automatic' property here signifies the fact that the disturbance is learned without any modelling efforts of the designer and this is an advantage of this approach).

As it can be seen in Workflow 1, LBCT consists of six steps. First, it uses an ILC algorithm to find an optimal system input $u^*_{i=M}(t)$ that rejects a repeating disturbance $d(t)$ (**Steps 1-3**). Thus, this phase identifies the structure of the periodic disturbance. Next, LBCT applies an FFT on the converged signal of ILC to obtain the corresponding frequency data set $\{\omega_j, \phi_j\}_{j=1}^{N_f}$ where $j \in \mathbb{N}^+$, $\omega_j \in \mathbb{R}^+$, $\phi_j \in \mathbb{C}^{1\times 1}$ and $N_f \in \mathbb{N}^+$ are the frequency data index, the frequency at j, the frequency response at ω_j and the number of frequency samples, respectively (**Step 4**). Then, the workflow includes a *Loewner framework* ([6] and [75]) based method to make an approximate fit on this data with a rational function $H(s) \in \mathcal{H}_\infty$ [1] (**Step 5**). Finally, the properties of $H(s)$ is used for designing a linear controller via state feedback, loop shaping or any other suitable method (**Step 6**).

Workflow 1: Learning Based Controller Tuning (LCBT)

Data: An internal closed-loop system $P \in \mathcal{H}_\infty$ (the open-loop plant plus a feedback controller) that is stable (see Figure 4.1) and subject to repetitive disturbance $d(t)$; a desired reference input $r(t)$; values for $\{\rho,\ \lambda\} \in \mathbb{R}^+$ and $M \in \mathbb{N}^+$ on Table 4.1.

Result: A linear controller rejecting a non-modelled repeating disturbance, $d(t)$.

1. ρ, λ and M can be chosen as suggested in [83];
2. Consider the switch is at '$S.2$' position in Figure 4.1;
3. Run ILC to find the system input that will attenuate the unwanted repeating frequencies;
4. Obtain the frequency data $\{\omega_j, \phi_j\}_{j=1}^{N_f}$ of the converged input signal, $u^*_{i=M}(t)$, from the last iteration;
5. Approximate a stable linear model $H(s) \in \mathcal{H}_\infty$ making a fit to this frequency data utilising [6] and [75];
6. Design a controller based on $H(s)$ properties (internal model control).

1. "$\mathcal{H}_\infty$ is the *Hardy space* of matrix-valued functions that are analytic and bounded in the open right-half of the complex plane defined by $Re(s) > 0$"

4.3 LBCT under complex disturbances

The advantages of LBCT can be appreciated the best when the complexity of the periodic disturbance increases. The more unknown/undetectable elements a disturbance has the more difficult it becomes to reject it. The complexity level of a disturbance may be due to many reasons such as its number of unknown frequencies, nonlinearities between its frequencies, the channel it enters the system etc. In order to asses these issues, let us assess the feasibility of LBCT approach by showing that it is capable of detecting all the repeating frequencies for two different complex periodic disturbances : (1) linearly combined of periodic disturbances and (2) non-linearly combined of periodic disturbances.

4.3.1 Linearly combined disturbance signals

To begin the analysis with a relatively simpler example, one can consider an unknown periodic disturbance signal made of several sinusoids of different frequencies. It is assumed that the combination of these sinusoids are linear and they all have the same amplitude for the sake of simplicity. This complex disturbance can then be written as :

$$d(t) = a(\sin(w_1 t) + \sin(w_2 t) + \sin(w_3 t) + \sin(w_4 t)) \tag{4.7}$$

where $a = 0.2$, $w_1 = 0.25 rad/s \approx 0.04 Hz$, $w_2 = 0.5 rad/s \approx 0.08 Hz$, $w_3 = 0.75 rad/s \approx 0.12 Hz$ and $w_4 = 1 rad/s \approx 0.16 Hz$. One can view the plot of this periodic disturbance in Figure 4.4. This disturbance is again added to the output of the system (4.1)-(4.2). For the simulation, Figure 4.1 is used together with the LBCT procedure provided in Workflow 1. As the first step, the parameters of ILC is set again as given on Table 4.1. It should be noted that the desired reference signal of the system is zero (i.e. $r(t) = 0$) and $r(t)$ is also the initial ILC signal at first iteration (i.e. $u_{i=0}(t) = r(t) = 0$). Next, ILC is run for 1000 iterations in order to learn the required system input that will meet the control objective, (i.e.$y_2(t) = 0$). The ILC signal $u_i(t)$ at $i = 1000$ is the converged signal which provides sufficiently low error in the output. For different design choices, this error can of course be reduced further by increasing the number of iterations (M) or by retuning ILC filter parameters (ρ and λ). The simulation results are given in Figure 4.5 and 4.6 which show the variation of system inputs and the corresponding system outputs in the iteration axis, respectively. The system inputs in Figure 4.5 are the ILC signals for the corresponding iterations since each new ILC signal was defined as the new system input. One can clearly observe from the results in Figure 4.6 that the ILC algorithm is able to gradually attenuate the disturbance effect to a huge degree such that the magnitude of the output at the last iteration reaches below 0.242%. Moreover, as it is expected mathematically, the ILC signal tended towards a signal which is a close approximation of the inverse of the disturbed system (see Figure 4.5).

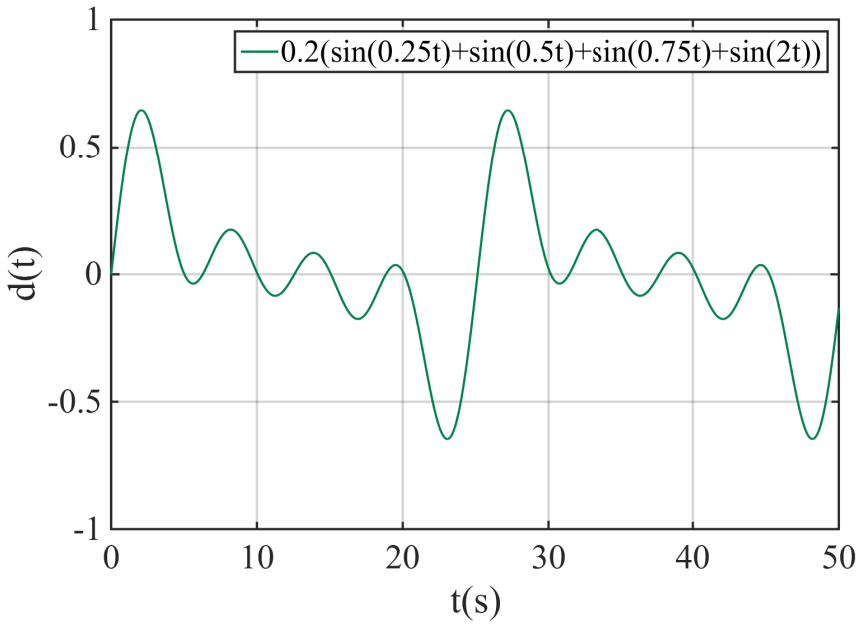

FIGURE 4.4: Linearly combined disturbance signal

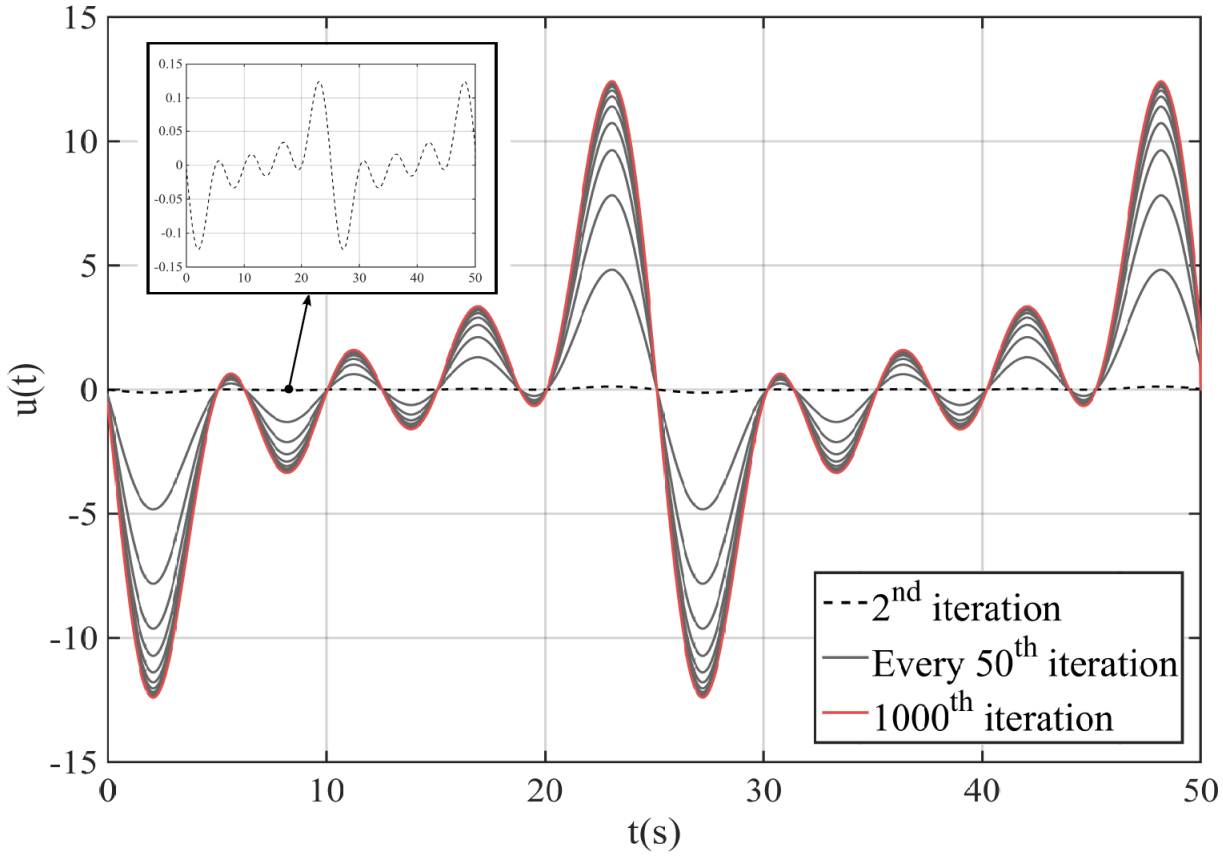

FIGURE 4.5: System input along iterations

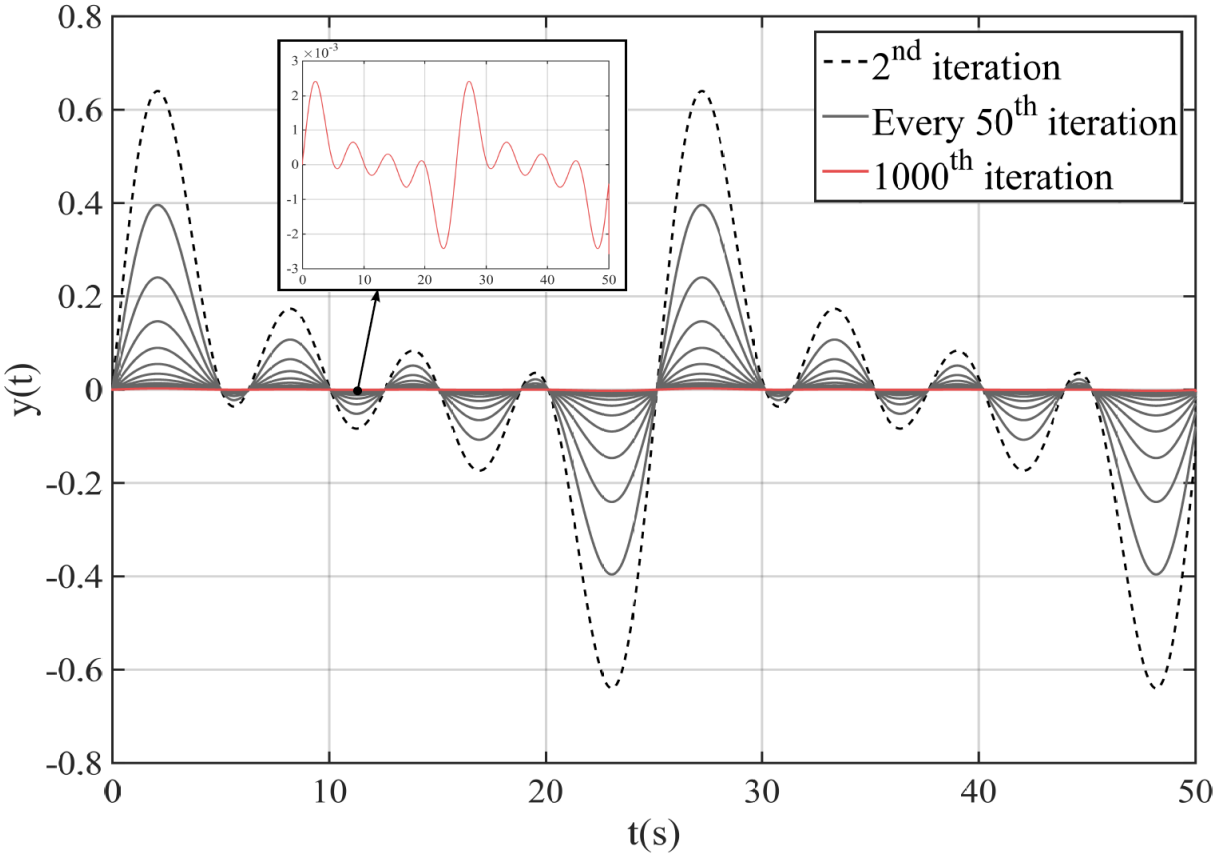

FIGURE 4.6: System output along iterations

The results until this point comprise the first 3 steps of the LBCT workflow and the remaining steps intend to identify the frequency spectrum of (4.7) and finding a linear model $H(s)$ for the ILC system via making a data fitting operation. Thus, the next step in the simulation procedure is to estimate the frequency components $\{\omega_j, \phi_j\}_{j=1}^{N_f}$ of the last ILC signal $u_{i=1000}$. For this purpose, ILC system is considered as a block having an impulse signal at the input and $u_{i=1000}$ at the output. When ILC is at this form, it is possible to utilise a method based on *Loewner framework* (detailed in [6] and [75]) to obtain an approximate fit on the frequency data with a rational function $H(s)$. The resulting fit and the frequency response data of the ILC system after this can be reviewed in Figure 4.7. Expectedly, the ILC system has peaks nearly at the frequencies of the disturbance (4.7) which proves its capability of singling out each repetitive component in the disturbance.

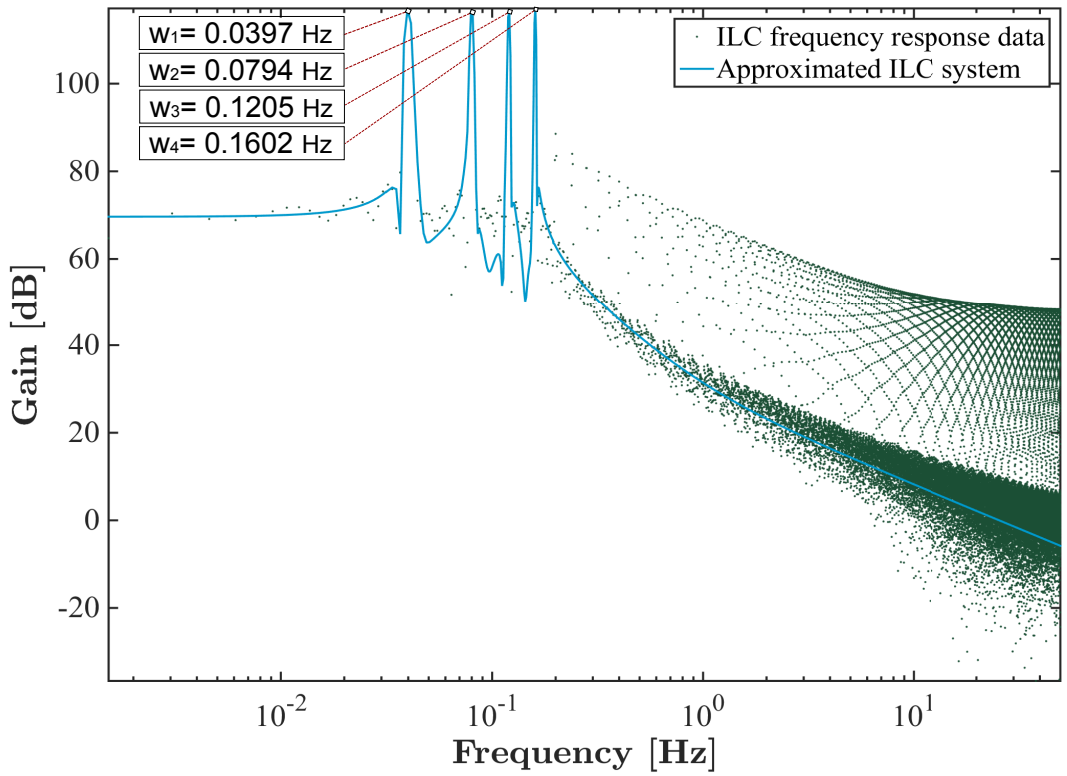

FIGURE 4.7: Frequency response of the ILC system and its approximated model, $H(s)$

Note that the presented model approximation process deduces the dominant eigenvalues of the ILC system.

4.3.2 Nonlinearly combined disturbance signals

Following the satisfactory rejection performance in the previous section, the analysis can now be extended to the case of unknown nonlinear periodic disturbance. For this purpose, $d(t)$ is this time selected in the form of three nonlinearly combined sinusoids with different frequencies :

$$\begin{aligned} &0.25((0.7(0.15 - 0.8\sin(w_1 t))^2 - 0.6\sin(w_2 t))^3 ... \\ &\quad ... - 0.35(\sin(w_3 t))^2) \end{aligned} \tag{4.8}$$

where $w_1 = 0.25rad/s \approx 0.04Hz$, $w_2 = 0.75rad/s \approx 0.12Hz$ and $w_3 = 1rad/s \approx 0.16Hz$. This disturbance is demonstrated in Figure 4.8. It can be seen that the nonlinear relations between w_1, w_2 and w_3 produces many other hidden frequencies which is the challenging part for approaches based on IMP or estimation. The same procedure of LBCT is applied and the only difference is the structure of the periodic disturbance. Moreover, the initialisation for ILC and the desired system reference remain the same as in previous section.

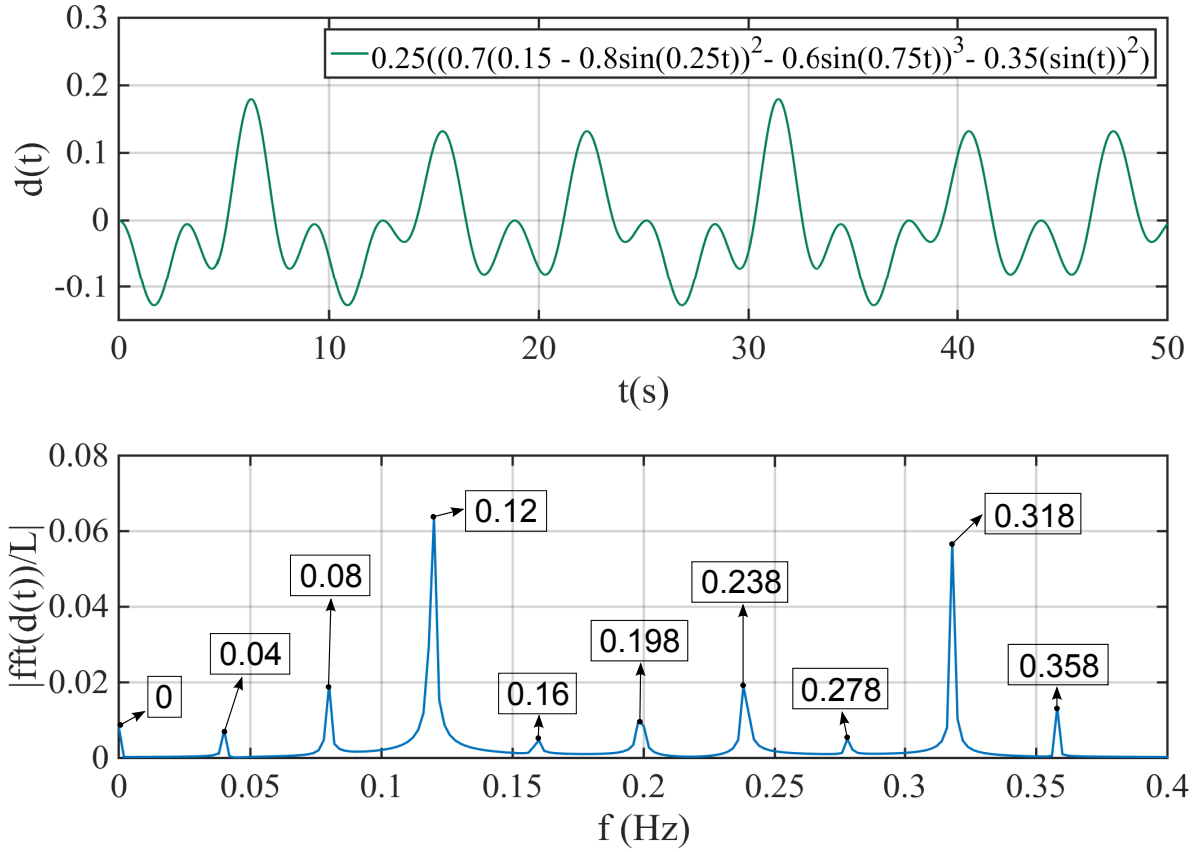

FIGURE 4.8: (Top) Non-linearly combined disturbance signal ; (Bottom) Fast Fourier transform of the disturbance signal showing the main frequencies (L : frequency data length).

The results from the ILC analysis are provided in Figure 4.9 and Figure 4.10 which show the evolutions of the system inputs and outputs, respectively. One can notice that the system output converges towards the desired reference as the ILC gradually calculates the proper inverse of the disturbed system. Next, analogically to the previous subsection, a frequency response of the converged ILC system is obtained. Then, an approximate fit on the corresponding data is applied following [6] and [75]. The result of this operation is given in Figure 4.11 where one can observe nearly the same frequencies as those found in the disturbance signal (5.22) (see Figure 4.8). Hence, it can clearly be concluded that the ILC algorithm successfully detects the frequencies to be attenuated inside the unknown nonlinear disturbance [2]. This also means that the linear approximate transfer function deduced from ILC's frequency data can provide accurate reference for tuning the parameters of a linear controller. Therefore, the LBCT workflow can be a powerful tool for automating the design of challenging disturbance rejection controllers. Furthermore, LBCT can be considered as a model-free alternative approach to unknown periodic disturbance rejection problem since the ILC filters in (2.98)), which can be recalled as

$$\mathbf{Q} = ((\lambda + \rho)I + \mathbf{P}^T\mathbf{P})^{-1}(\lambda\mathbf{I} + \mathbf{P}^T\mathbf{P})$$
$$\mathbf{L} = (\lambda\mathbf{I} + \mathbf{P}^T\mathbf{P})^{-1}\mathbf{P}^T,$$

use only the information of the existing closed-loop system $\mathbf{P}$. This type of data-based

2. In this thesis, the term nonlinear disturbance means a disturbance that is produced by a nonlinear system/oscillator

approach converts a difficult modelling problem to a simple filtering operation.

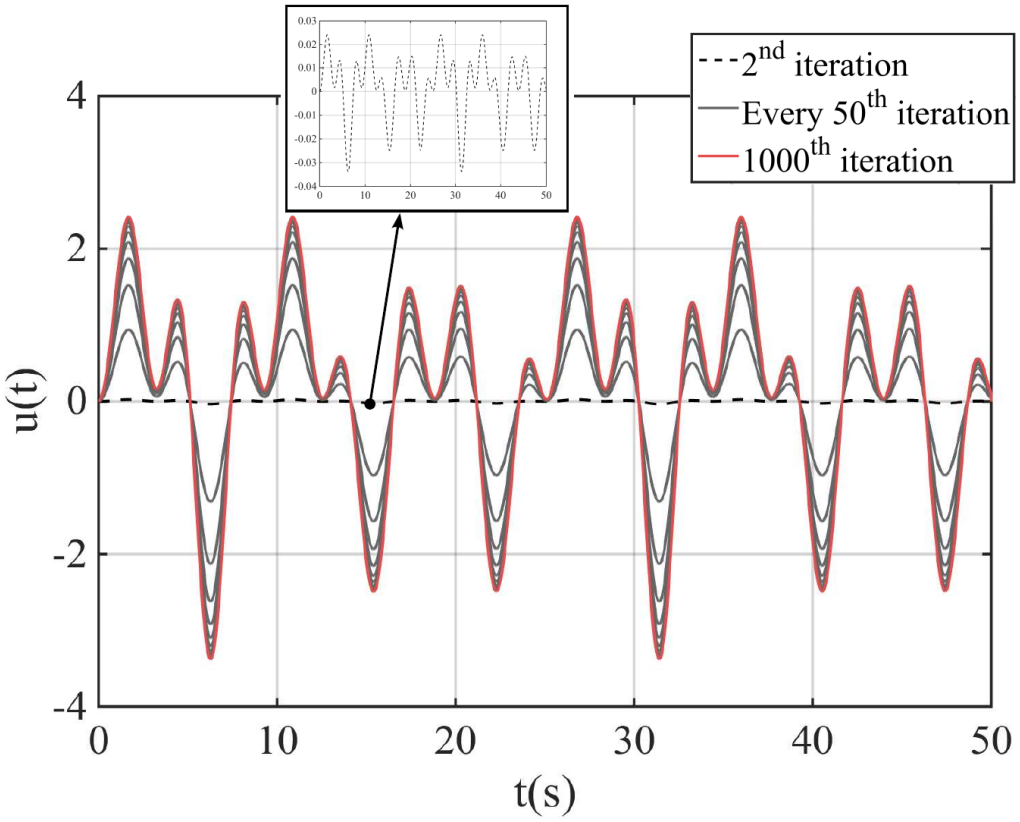

FIGURE 4.9: System input along iterations

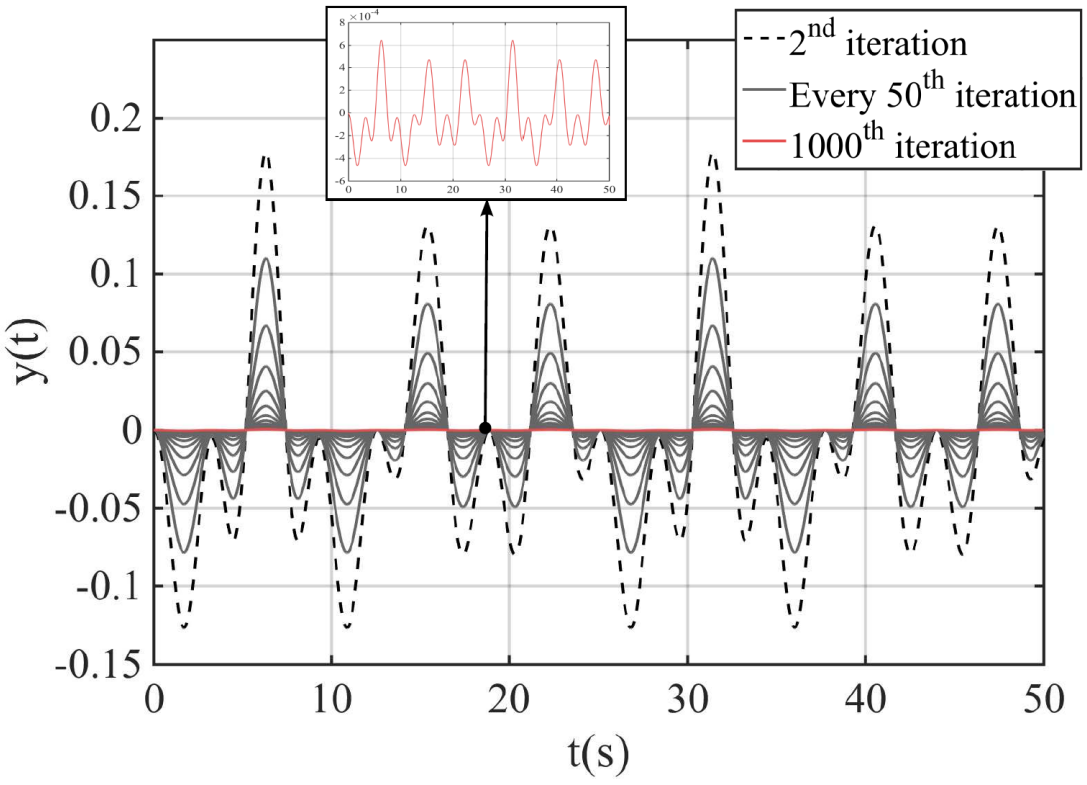

FIGURE 4.10: System output along iterations

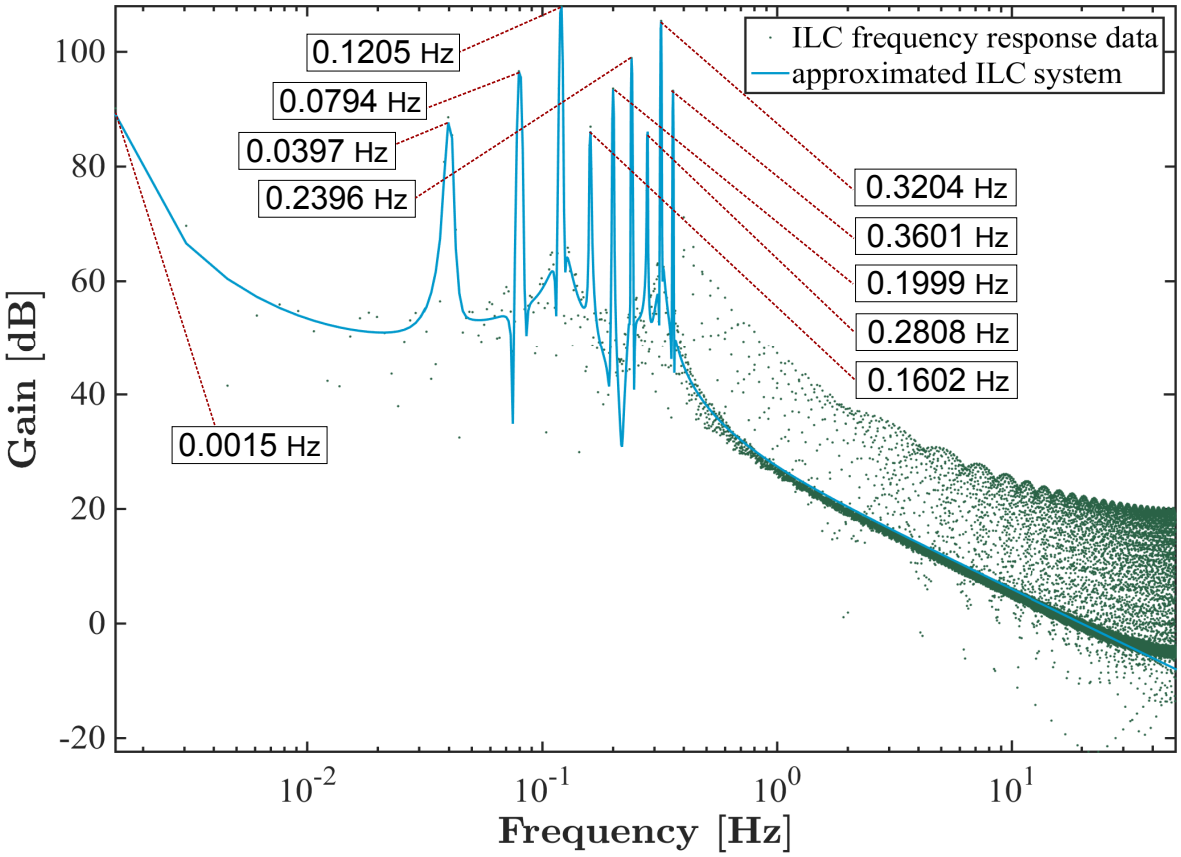

FIGURE 4.11: Frequency response of the ILC system and its approximated model, $H(s)$

4.4 Conclusion

In this chapter, it has been demostrated that ILC can be used as a tool for automatically creating (or tuning) linear feedback controllers for rejecting unknown periodic disturbances. The idea has been presented under the name of *LBCT workflow* and the main contribution of the work is twofold : (1) utilising ILC, the bridge between the requirement of having *a priori* knowledge of the periodic disturbance and building an internal model principle based feedback controller has been broken ; (2) utilising ILC, an automatisation is achieved for the tuning of feedback controller parameters.

The initial motivation of the presented work came after observing an equivalance of ILC to the augmented state feedback control in rejecting a simple sinusoidal repetitive disturbance. According to the results, the ILC showed a rather successful rejection performance that is superior to state feedback control. This is due to the fact that ILC was able to gradually remove the repeating errors and anticipate for the transients in the response without needing to know the disturbance. Hence, this brings out the following question : Why not make benefit of learning methods (like ILC) that are based on repetitive data manipulation when technology becomes less and less an issue in terms of computational speed, data storage etc. ? Trying to correctly model the repetitive disturbances can be quite a tedious process if the disturbances are unknown, partially known, nonlinear or lumped with other effects such as parametric uncertainties.

The missing part of this work is that the last step of the LBCT workflow, i.e. the transfer function $H(s)$, has not been demonstrated (see Workflow1). It could be interesting to create the $H(s)$ (which is an approximation of the ILC's performance in feedback) and compare its rejection performance directly to ILC such that the efficiency of the LBCT procedure can fully be evaluated. It could also be interesting to further test this remaining part with a practical application on a real system.

Another improvement can be done by developing the presented workflow for making it feasible to nonlinear systems under periodic disturbances. It is also always possible to test a different ILC method for obtaining the same results faster in terms of computational time. The selected tuning for the norm optimal (NO-ILC) may not be the best and other optimisation methods can be used (NO-ILC is based on Lagrange multipliers technique which is based on an analytical minimisation process).

Thankfully, the next chapter can provide some improvements for some of the above mentioned remaining issues. In the next chapter, a new control approach has been introduced which demonstrated how to fully transform ILC into a feedback controller. This approach utilises the LBCT workflow from this chapter and adds more to it by building a new framework which is the combination of ILC and robust output regulation based on internal model principle. Furthermore, the next chapter is important for the fact that it finally develops the combined approach for handling a *nonlinear system* under nonlinear periodic disturbances.

Chapter 5

Supervised Output Regulation via Iterative Learning Control

Contents

The previous section has demonstrated the effectiveness of using ILC as a tool for simplifying the design of state feedback controllers under complex disturbances. The main idea and the procedure to follow has been given by the *Learning Based Controller Tuning (LCBT)* workflow. The promising results obtained via LCBT has been a motivation for the further development of the presented idea towards a new and more generic framework which covers applications on nonlinear [1] repetitive systems.

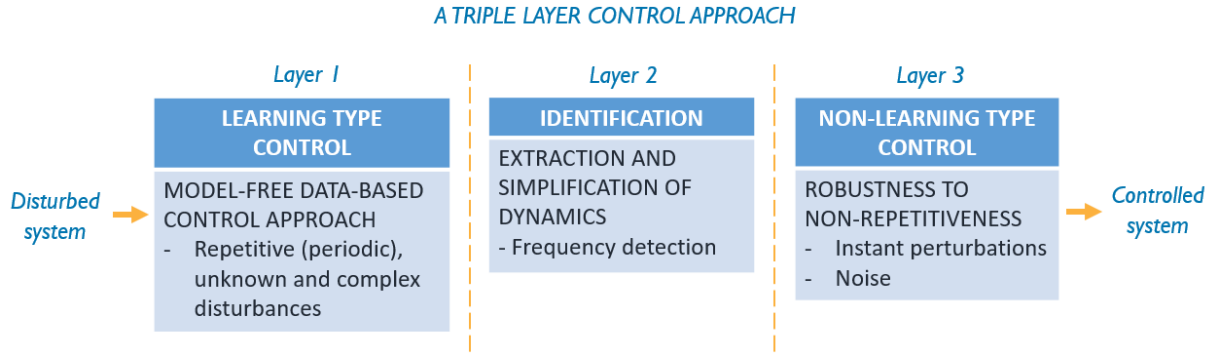

Figure 5.1: Diagram of the proposed triple-layer control approach

1. Note that here the term 'nonlinear' is used for the weak nonlinearities (i.e. excluding saturation, friction, nondifferentiablity etc.).

The new framework shown in Figure 5.1 is based on two design problems to be solved :

1. How to reject a complex unknown periodic disturbance without modelling it ?
2. How to improve controller robustness and rejection precision simultaneously ?

Following Figure 5.1, the solution to these problems is proposed in the form of a triple-layer control approach that benefits from the distinct features of three different methods : learning-type control, identification [2] and non-learning-type control. The disctinct characteristics of each layer can be seen as follows : (1) the learning-type control methods are generally designed to function in a model-free fashion and they utilise the repetitive patterns in the system data in order to accomodate unknown periodic disturbances ; (2) the identification methods can help extracting the most useful information from a signal (e.g. frequencies, amplitudes etc.) in an approximative way ; (3) the non-learning-type methods are capable of giving robustness against non-repetitive effects on a system (e.g instant perturbations and noise). It can of course be argued that both learning-type or non-learning-type methods can separately be used to solve the above mentioned design problem and this is already the case for existing approaches in both categories (see Section sec :introduction-1-1 and the references therein). However, instead of trying to force each method to achieve goals that are nonconforming to their nature, one can create a fusion of both approaches and get much better results in terms of performance and practicality. The author believes that this can be an important step in its broad application. A diagram for the proposed triple-layer approach can be seen in Figure. 5.1.

The application of this kind of scheme can be seen as a transformation from data-based logic to conventional logic in the control design. Assuming that the required control action for rejecting the periodic disturbance is denoted as u^*, the design steps of the triple-layer controller is as given in Figure 5.2. At this point, the question that rises in mind is : which methods to select for each layer ? A clever choice in this case would be to combine some methods that complement each other, i.e they fix each other's disadvantageous sides without sacrificing much (or if possible, at all) on their unique features. Considering this, it is proposed that a reasonable selection would be to go for *iterative learning control* as the learning-type method and *output regulation based on internal model principle (IMP)* as the non-learning-type method. As for the identification layer, a couple of different approaches will be shown during the design process (e.g. the Fast Fourier Transformation, the Loewner framework based identification from Chapter 2 and an identification method called *Hankel*). It should be noted that the identification is just an intermadiate tool required for combining the first and the third layer such that it does not have much effect on the method choices for these layers. The goal is not to identify the model but to find the information that is necessary to the internal model conroller. Thus, the feature analysis of the mentioned identification approaches will be revisited in the latter section. The reasons behind choosing ILC and output regulation has been demonstrated in Figure 5.3 which puts down the advantages and disadvantages of both sides. The conclusion of this visualisation is that both methods can really fix each other and their fusion can even produce something much better than they could do when applied alone.

2. Identification on some meta data.

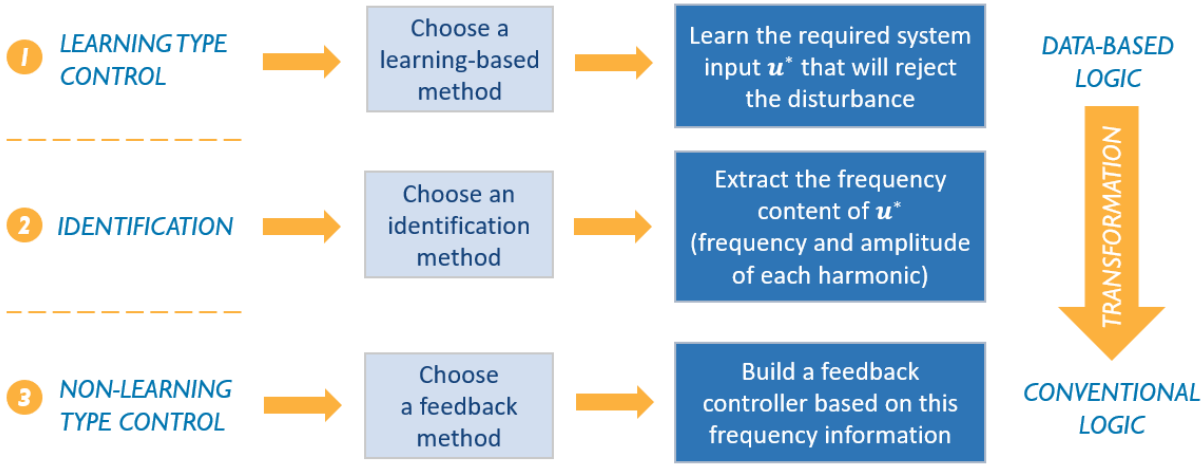

FIGURE 5.2: Design steps of the triple-layer controller

The resulting triple-layer controller which combines ILC and output regulation is given the name *Supervised Output Regulation via Iterative Learning Control (SOR-ILC)* in this chapter. This is simply because ILC's role in this fusion can be interpreted as being a *supervisor* that guides the actions of the output regulator towards rejection of an *a priori* unknown periodic disturbance. Below, the design procedure and analysis of SOR-ILC are given for two different cases :

1. A linear system under an unknown complex periodic disturbance,
2. A nonlinear system under an unknown complex periodic disturbance.

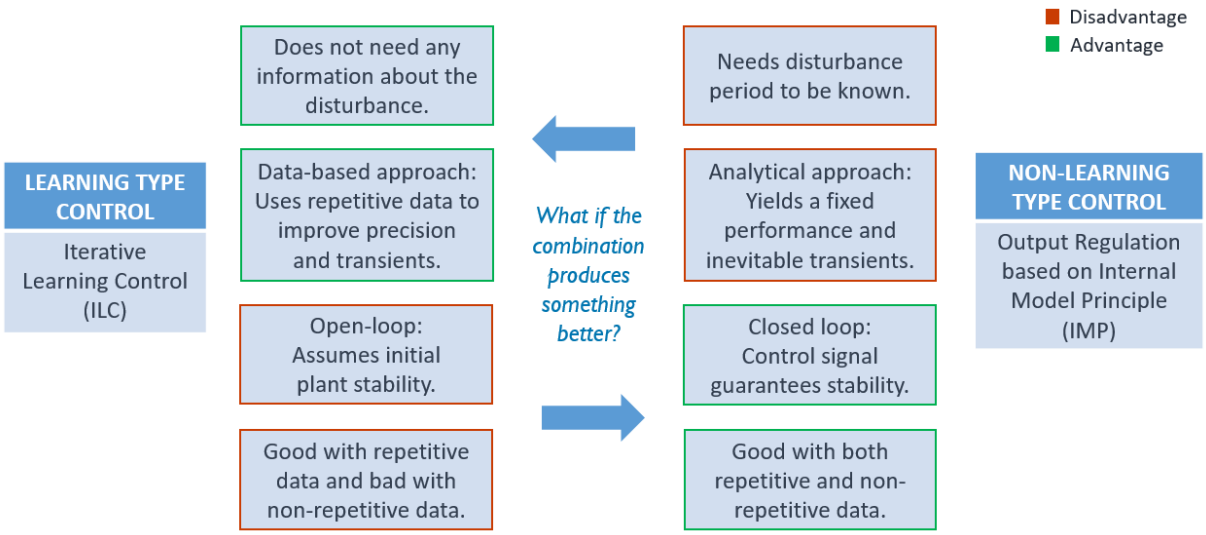

FIGURE 5.3: Selections for Layer 1 and Layer 3

The problem definitions and numerical results for both cases are thus provided separately in the following two sections. Section 5.1 is useful for understanding the insight of the process since it first includes a simple introductory example and then analyses the rejection of a more complicated disturbance with some additional effects. On the other hand, Section 5.2 provides a nice demonstration for a lot more challenging design problem which analyses the rejection of Van der Pol oscillations via SOR-ILC. Moreover, in both sections the

general design steps stay the same as in Figure 5.2. Let us now explain the design of SOR-ILC in the mentioned order.

5.1 SOR-ILC for linear systems under unknown complex periodic disturbances

5.1.1 Problem definition

Let us consider the following linear time invariant open-loop plant in continuous-time

$$\dot{x}(t) = \mathbf{A}x(t) + Bu(t) + \mathbf{W}_x w(t), \tag{5.1}$$
$$y(t) = Cx(t) + Du(t) + \mathbf{W}_y w(t). \tag{5.2}$$

where $x(t) \in \mathbb{R}^n$ is the state, $u(t) \in \mathbb{R}^1$ is the system input, $y(t) \in \mathbb{R}^1$ is the output, $w(t) \in \mathbb{R}^{n_w}$ is a signal representing unknown disturbances to be rejected and $\mathbf{W}_x \in \mathbb{R}^{n_w \times n_w}$ and $\mathbf{W}_y \in \mathbb{R}^{n_w \times n_w}$ are the weight matrices for the internal and external disturbances, respectively. For simplicity, it is supposed that the full state $x(t)$ is available for feedback design and that $D = 0$. Such assumptions are not restrictive since a straightforward extension to the multivariable and output feedback cases of all the forthcoming results can be formulated by following [10, 24] or [100]. The final goal of the rejection problem can be formulated as follows :

Global Objective : Find $u(t)$ such that $\lim_{t\to\infty} y(t) \to 0$

However, the calculation of the input $u(t)$ using SOR-ILC is not direct and, as mentioned before, it is found by applying three steps (i.e. three layers, see Figure 5.2). Therefore, let us divide the rejection problem to its subproblems :

Objective of Layer 1 (ILC) : Find $u^*(t) = u_{i=N}(t)$ such that $\lim_{i\to N} y(t) \to y^*(t)$ where $y^*(t) \approx 0$ (i.e $||e_{i=N}(t)|| \leq \epsilon_{ILC}$ where ϵ_{ILC} is an arbitrarily small ILC error).

Objective of Layer 2 (Identification) : Given $u^*(t)$ find its frequency data $\{\omega_j, A_j\}_{j=1}^{N_f}$ where N_f denotes the length of the data and the pair $\{\omega_j, A_j\}$ consists of the pulsation and amplitude, respectively.

Objective of Layer 3 (Output regulation via IMC) : Given the frequency data of $u^*(t)$ find a feedback control $u(t)$ such that $\lim_{t\to\infty} y(t) \to 0$ and $||e(t)|| \leq \epsilon_{OR} \leq \epsilon_{ILC}$ where ϵ_{OR} is an arbitrarily small output regulation error.

Having determined the objectives, let us now show the control design of SOR-ILC.

5.1.2 Control Design

Layer 1 : Iterative Learning Control (ILC)

The fundamental idea of ILC can be recalled by looking at the equation (2.7), i.e.

$$u_{i+1}(t) = \mathbf{Q}u_i(t) + \mathbf{QL}e_i(t)$$

In this simple form, ILC applies its $\mathbf{Q}$ and $\mathbf{L}$ filters on the previous system input $u_i(t)$ and the previous system error $e_i(t)$ and finds the next system input $u_{i+1}(t)$. With the proper choices of Q and L, the input solution $u^*(t) = u_{i=N}(t)$ that pushes the system error below an arbitrary value ϵ_{ILC} can theoretically be achieved within N iterations. Hence, the main challenge of ILC is to design these filters (Chapter 2 can be revisited to see the various possible design choices of an ILC technique).

In this chapter, the selected ILC method is again the norm-optimal ILC (NO-ILC) due to the satisfactory results and its feasibility obtained in the previous chapters. The details regarding the NO-ILC design procedure can be reviewed in Section 2.4.2. In brief, NO-ILC calculates the current system input u_{i+1} given above by analytically solving for the u_{i+1} that minimises the cost function

$$J(u_{i+1}(t)) = e_{i+1}^T(t)\mathbf{W}_e e_{i+1}(t) + u_{i+1}^T(t)\mathbf{W}_u u_{i+1}(t) + \lambda[(u_{i+1}(t) - u_i(t))^T(u_{i+1}(t) - u_i(t))]$$

where $\lambda \in \mathbb{R}^1$ is the Lagrange multiplier; $\mathbf{W}_e = \rho\mathbf{I} \in \mathbb{R}^{N\times N}$ and $\mathbf{W}_u = \mathbf{I} \in \mathbb{R}^{N\times N}$ are the weighting matrices for the error and the input, respectively, with $\mathbf{I}$ being the identity matrix. Thus, the optimal solution of $J(u_{i+1})$ gives the following required $\mathbf{Q} \in \mathbb{R}^{N\times N}$ and $\mathbf{L} \in \mathbb{R}^{N\times N}$ filters which constitute the core of NO-ILC.

$$\mathbf{Q} = ((\lambda + \rho)\mathbf{I} + \mathbf{P}^T\mathbf{P})^{-1}(\lambda\mathbf{I} + \mathbf{P}^T\mathbf{P})$$
$$\mathbf{L} = (\lambda\mathbf{I} + \mathbf{P}^T\mathbf{P})^{-1}\mathbf{P}^T$$

Here, the $\mathbf{P}$ matrix referes to the lifted-matrix (2.15) of the internal system [3]. Moreover, the convergence and robustness performance of NO-ILC depends on the heuristic selection of $\lambda > 0$ and $\rho > 0$ under the requirement of two criteria : $||\mathbf{Q}||_2 < 1$ and $||\mathbf{QL}||_2 \leq 0.5/\sqrt{\rho + \lambda}$.

Layer 2 : Identification via Loewner

For the linear system case, the chosen identification method is based on FFT and model approximation. It should be noted that this is just a choice and the given identification method is also applicable to the nonlinear system case that will be shown in the next section. The only difference is that the identification process in this section is based on directly using the frequency-domain data while the one given in the next section will be capable of providing similar results based on the time-domain data only. Thus, in this section the modes of the ILC system is found by utilising the frequency-domain information of the converged ILC signal from the last iteration (i.e. $u^*(t) = u_{i=N}(t)$). The overall procedure is defined by the following steps :

1. The frequency data of $u^*(t)$ is acquired by applying a Fast Fourier Transformation (FFT) on it. For the data of length N_f, this step yields the frequency data set $\{\omega_n, \phi_n\}_{n=1}^{N_f}$.
2. This data is divided into two parts : $\{\mu_j, v_j\}_{j=2n-1}^{q}$ and $\{\lambda_k, w_k\}_{k=2n}^{p}$ where $n = 1, ..., p, p+1$ with $q = p+1 = (N_f+1)/2$ and $p = (N_f)/2$; μ_j and λ_k are frequencies

3. Note that $\mathbf{P}$ only refers to the system (plant) and it does not contain any information regarding the disturbances acting on itself

at their corresponding j, k value ; and v_j and w_k are vectors containing both gains and phases at each μ_j and λ_k, respectively.

3. The identification problem is to find a rational function $\hat{H}$ that will closely approximate the dynamics of $u^*(t)$ which can be of a reduced order depending on the preference. This problem can be formulated as follows :

   Goal : Given $\{\mu_j, v_j\}_{j=2n-1}^{q}$ and $\{\lambda_k, w_k\}_{k=2n}^{p}$, seek $\hat{H}$ such that $\hat{H}(\mu_j) = v_j$ and $\hat{H}(\lambda_k) = w_k$ for $j = 1, ..., q$ and $k = j + 1, ..., p$.

4. Finally, the model approximation of $u^*(t)$ is obtained via a rational interpolation based on Loewner Framework [5] :

$$\hat{H}(s) = \mathbf{W}(-s\mathbb{L} + \mathbb{L}_\sigma)^{-1}\mathbf{V} \tag{5.3}$$

   where $\mathbb{L}$, $\mathbb{L}_\sigma$, W and V are given respectively as :

$$\mathbb{L} = \begin{bmatrix} \frac{v_1 - w_1}{\mu_1 - \lambda_1} & \cdots & \frac{v_1 - w_p}{\mu_1 - \lambda_p} \\ \vdots & \ddots & \vdots \\ \frac{v_q - w_1}{\mu_q - \lambda_1} & \cdots & \frac{v_q - w_p}{\mu_q - \lambda_p} \end{bmatrix} \tag{5.4}$$

$$\mathbb{L}_\sigma = \begin{bmatrix} \frac{\mu_1 v_1 - w_1 \lambda_1}{\mu_1 - \lambda_1} & \cdots & \frac{\mu_1 v_1 - w_p \lambda_p}{\mu_1 - \lambda_p} \\ \vdots & \ddots & \vdots \\ \frac{\mu_q v_q - w_1 \lambda_1}{\mu_q - \lambda_1} & \cdots & \frac{\mu_q v_q - w_p \lambda_p}{\mu_q - \lambda_p} \end{bmatrix} \tag{5.5}$$

$$\mathbf{W} = \begin{bmatrix} w_1 & \cdots & w_k \end{bmatrix} \tag{5.6}$$

$$\mathbf{V}^T = \begin{bmatrix} v_1 & \cdots & v_k \end{bmatrix} \tag{5.7}$$

   Note that the transfer function (5.3) is a data driven interpolation. If all the frequencies of $u^*(t)$ are used in the model approximation, the $\hat{H}(s)$ should provide an exact representation of the system.

Once a sufficiently [4] accurate model approximation is obtained for $u^*(t)$, the frequency data corresponding to this approximation can be used in the design of the output regulator's internal model (the data needed for the output regulation is $\{\omega_n, A_n\}_{n=1}^{N_f}$ and phase values ϕ_n are not necessary). Furthermore, the amplitude values A_n that are associated to each w_j are only used to put the identified frequencies ω_j in the order of descending amplitudes. This preference can be seen as way of determining the importance of each ω_n. Therefore, it is not obligatory and it may not indicate the importance of frequencies for other applications.

4. The accuracy of the model approximation is related to the amount of properly detected frequencies by the FFT and then to how many of these are used in the Loewner framework's rational interpolation (see (5.3)-(5.7)). One can simply include more frequencies and visually decide on the accuracy of the model fit to $u^*(t)$.

Layer 3 : Output Regulation

Let us now come back to the system (5.1)-(5.2). In output regulation theory (see [36, 43]), the signal $w(t)$ is usually denoted as *exosignal* and it is generated by an autonomous exosystem given as

$$\dot{w}(t) = \mathbf{S}w(t) \tag{5.8}$$

where $\mathbf{S}$ is a neutrally stable matrix of the form $\mathbf{S} = \text{blkdiag}(0\,^5, \omega_1\mathbf{J}, \ldots, \omega_\rho\mathbf{J})$, with

$$\mathbf{J} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{5.9}$$

Here, $\rho \leq N_f$ is the number of frequencies decided to be used for building the internal model (i.e. the matrix $\mathbf{S}$) and ω_n for $n = 1, \ldots, \rho$ are all different and characterize the frequencies of the signal $w(t)$. Thus, one is not obliged to use all the frequencies detected in the Layer 2. Figure 5.4 demonstrates the blockdiagram of a common output regulator scheme. The problem of regulating $e(t)$ to zero while maintaining the state $x(t)$ bounded for all positive times, denoted as output regulation, is solved under the following customary assumptions (see [24]).

Assumption 5.1 (Stabilisability) *The pair* $(\boldsymbol{A}, B)$ *is stabilisable.*

Assumption 5.2 (Non-resonance condition) *The matrix* $\begin{bmatrix} \boldsymbol{A} - \lambda \boldsymbol{I} & B \\ C & 0 \end{bmatrix}$ *has independent rows for each* $\lambda \in \mathrm{i}\mathbb{R}$ *where* i *denotes the imaginary number.*

The Assumption 5.2 [6] mainly states that the transfer function between $u(t)$ and $y(t)$ has no zeros with zero real part. Although this condition is more stringent than standard output regulation framework (where the rank condition needs to hold only for each λ eigenvalue of $\mathbf{S}$), it is necessary in the current scenario in which an adaptive solution is sought. Hence, the solution to the output regulation problem is obtained with the following two-step procedure.

S1) Extend the system (5.1)-(5.2) with an internal model unit (IMU) of the form

$$\dot{\eta}(t) = \hat{\mathbf{S}}\eta(t) + \Gamma e(t) \tag{5.10}$$

where $\eta = (\eta_0, \eta_1, \ldots, \eta_\varrho)^T \in \mathbb{R}^{1+2\varrho}$ is the state of the IMU. The matrix $\hat{\mathbf{S}}$ is built using the identified frequencies from Layer 2 (i.e. the $\{\omega_n\}_{n=1}^{\rho}$) such that $\mathbf{S} = \text{blkdiag}(0, \omega_1 J, \ldots, \omega_\varrho J)$, with J of the form (5.9). On the other hand, Γ has to be chosen in that $(\hat{\mathbf{S}}, \Gamma)$ is a controllable pair.

S2) Stabilize the extended system (5.1)-(5.2), (5.10) with a controller of the form

$$u(t) = K_1 x(t) + K_2 \eta(t) \tag{5.11}$$

5. This element has a purpose of including constant repetitive trajectory tracking or constant repetitive disturbance rejection cases.

6. The matrices $\mathbf{A}$ and B do not have to be known. They can be obtained simply using the model approximation procedure in Layer 2.

such that the unforced (i.e. when $w(t) = 0$) closed loop system is asymptotically stable.

Theorem 5.1 *Suppose $\varrho = \rho$, $\hat{\mathbf{S}} = \boldsymbol{S}$ in* (5.10) *with ρ, S given by* (5.8) *and that K_1, K_2 in* (5.11) *ensures asymptotic stability of the unforced closed loop system* (5.1), (5.2),(5.10), (5.11). *Then, the output regulation problem for the system* (5.1)- (5.2), *is solved, namely solutions of the closed-loop system* (5.1), (5.2), (5.10), (5.11), *forced by* (5.8), *are bounded for all $t \geq 0$ and satisfy $\lim_{t\to\infty} y(t) = 0$. Furthermore, the above properties are robust to any (small) parameter perturbations of the nominal matrices $(\boldsymbol{A}, B, C)$ that do not break the stability property of the (unforced) closed-loop system.*

Proof. See [24, 36, 43].

In addition to previous result, note that the design of the dynamical regulator (5.10), (5.11), is independent from how the exosignal $w(t)$ affects the plant (5.1)-(5.2), namely from the matrices $\mathbf{W}_x$ and $\mathbf{W}_y$.

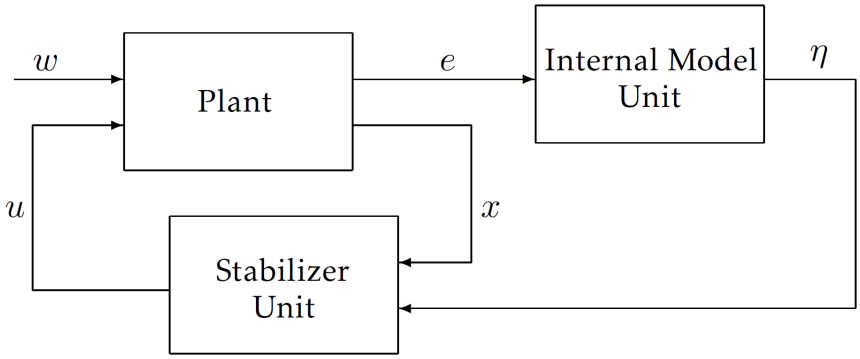

FIGURE 5.4: Output regulator scheme

As one can see from Theorem 5.1, the main challenge of the internal model based approach is that for the design of (5.10), one needs to perfectly know $\mathbf{S}$, that is the frequencies ω_n of the exosignals (5.8), and such assumption remains unrealistic in a practical scenario. Furthermore, each time the matrix $\hat{\mathbf{S}}$ in (5.10) is aimed to be adapted, the matrices K_1, K_2 of the controller (5.11) may need to be redesigned. Accordingly, a pole-placement strategy is not well suited in such context. A possible solution could be parametrizing the dynamic (5.10) by following the parametrisation of [82]. However, the extension to the nonlinear case is not trivial with this approach, see [11]. Therefore, in this work, we follow another route which is maintaining the structure of (5.10) and proposing a design of (5.11) based on forwarding techniques proposed in [10]. The advantage of such approach is the self-re-parametrisation of the stabilizer unit each time $\hat{\mathbf{S}}$ in (5.10) is modified (i.e. each time the frequencies $\hat{\omega}_i$ are modified). Thus, following [10], we design (5.11) as

$$u(t) = -\beta B^T \mathbf{R} x(t) + \sum_{i=1}^{\varrho} \mu_i B^T \mathbf{M}_i^T (\eta_i(t) - \mathbf{M}_i x(t)) \tag{5.12}$$

where the parameters $\beta \geq 0$ and $\mu_i > 0$ can be seen as free design parameters which can be utilized to put weight on specific frequencies as well as to increase rejection performance,

and the matrices $\mathbf{R}$, $\mathbf{T}$, and $\mathbf{M}$ are computed respectively as solution to

$$\mathbf{R}\mathbf{A} + \mathbf{A}^T\mathbf{R} = -\mathbf{I} \tag{5.13}$$

$$\mathbf{T}\hat{\mathbf{S}} + \hat{\mathbf{S}}^T\mathbf{T} = 0 \tag{5.14}$$

$$\mathbf{M}_i\mathbf{A} = \hat{\omega}_i\mathbf{J}\mathbf{M}_i + \Gamma_i C, \tag{5.15}$$

with J defined in (5.9) and $\Gamma = [\Gamma_0, \Gamma_i, \ldots, \Gamma_\rho]^T$ with $\Gamma_0 = 1$ and $\Gamma_i = [0, 1]^T$ for all $i = 1, \ldots, \varrho$. Note that the skew-symmetricity of the $\hat{\mathbf{S}}$ matrix allows $\mathbf{T}$ to be the identity matrix. Note also that with respect to (5.11), we selected

$$K_1 = -\beta B^T\mathbf{R} - \sum_{i=1}^{\rho} \mu_i B^T \tag{5.16}$$

$$K_2 = [\mu_1 B^T\mathbf{M}_1^T, \mu_2 B^T\mathbf{M}_2^T, ..., \mu_\rho B^T\mathbf{M}_\rho^T]. \tag{5.17}$$

Then, the following proposition can be stated.

Proposition 5.1 *Under Assumptions 5.1 and 5.2 for any $\hat{\mathbf{S}}$ designed as in S1, the unforced closed-loop system* (5.1)-(5.2)*,* (5.10)*,* (5.12) *with $\boldsymbol{R} = \boldsymbol{R}^T > 0$, $\boldsymbol{T} = \boldsymbol{T}^T$ and $\boldsymbol{M}$ designed as in* (5.13)-(5.15)*, respectively, is asymptotically stable.*

Proof. *The proof can be found in [10] and it is based on the analysis of the derivative of the Lyapunov function $V = x^T\boldsymbol{R}x + (\eta - \boldsymbol{M}x)^T\boldsymbol{T}(\eta - \boldsymbol{M}x)$.*

5.1.3 Numerical Analysis

After seeing the design steps of SOR-ILC, it is time to demonstrate its functioning. As said previously, the SOR-ILC is first demonstrated with a numerical simulation that takes into account a simple external periodic disturbance. Then, another simulation is done to demonstrate its performance under a complex disturbance case, i.e. a combination of complex effects such as internal and external periodic disturbances, parametric uncertainties and lack of frequency knowledge.

Analysis 1 : Simple periodic disturbance

Let us consider that (5.1)-(5.2) is a second-order open-loop system with

$$\mathbf{A} = \begin{bmatrix} -3.5014 & -3.0003 \\ 1 & 0 \end{bmatrix},\ B = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\quad \begin{matrix} \mathbf{C} = \mathbf{I}_{2\times 2}, \\ D = 0_{2\times 1}. \end{matrix}$$

It is assumed that both states are accesible and there is a sinusoidal disturbance acting on the second output only, i.e. $\mathbf{W}_x = \mathbf{0}_{2\times 2}$, $\mathbf{W}_y = [0, 0; 0, 1]$ and $w(t) = d(t) = a \cdot sin(\omega t)$ where $a = 0.2$ and $\omega = 1rad/s = 0.1591Hz$ are the amplitude and the pulsation of the signal, respectively. Since most systems have some already existing controllers in practice, it is also supposed that the system $(\mathbf{A}, B, \mathbf{C}, D)$ is in closed-loop with : $K_0 = [-3.4728, 15.5866]$. This step is a requirement for the proper functioning of ILC

in the first layer of SOR-ILC which yields a stable inner system ($\mathbf{P}$) that is defined by

$$\dot{x}(t) = \mathbf{A}_{in}x(t) + B_{in}u(t), \tag{5.18}$$
$$y(t) = C_{in}x(t) + W_y d(t) \tag{5.19}$$

where $\mathbf{A}_{in} = \mathbf{A} + BK$, $B_{in} = B$ and $C_{in} = [0, 1]$ and $W_y = [0, 1]$ since the disturbance is only considered on the second state. This system can be observed in Figure 5.5 (inner system block). Since the model of the disturbance is not included in the state feedback, the gains given above are already incapable of dealing with the given disturbance. Thus, this architecture portrays a scenario in which the existing closed-loop system is insufficient of dealing with an external periodic disturbance.

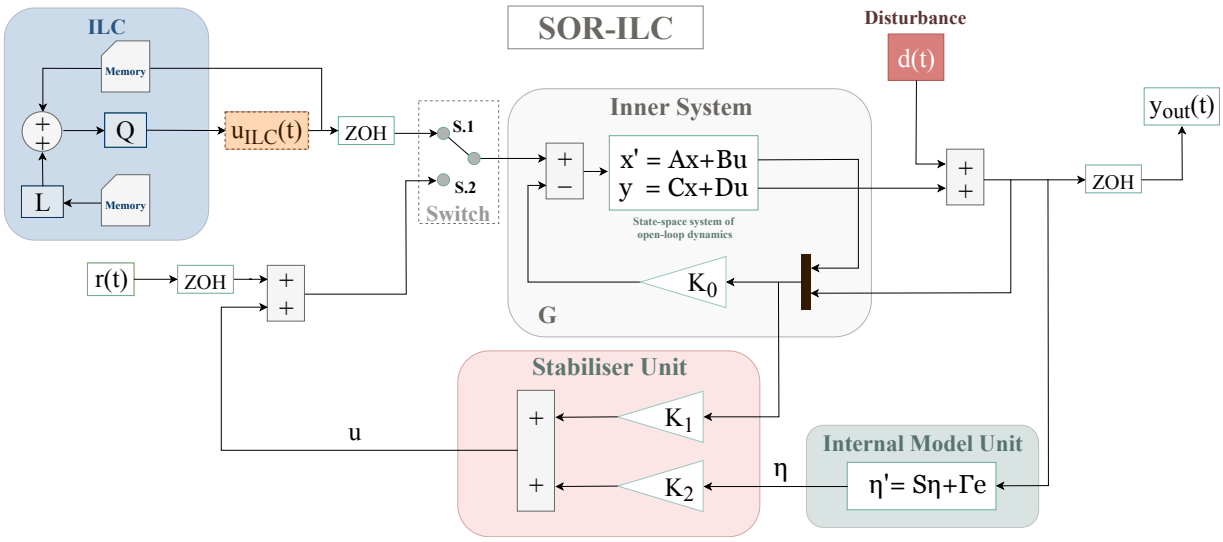

FIGURE 5.5: Simulation model

Once the inner system is determined, one can follow the procedure given in Section 5.1.2 where the first step is to properly initialise the ILC and iteratively run the system in Figure 5.5 until a satisfying rejection level is achieved. The amount of precision is a matter of tuning or simply is related to the power of the selected ILC approach. The focus of the analysis provided here is to rather analyse the functioning of SOR-ILC framework than finding the most precise tuning. Thus, if NO-ILC initialisation is carried out according to Table 5.1, the ILC inputs shown in Figure 5.6 iteratively improve the rejection performance as in Figure 5.7. The power of this optimisation-based approach can easily be understood by checking the output amplitude at the last iteration in Figure 5.7.

TABLE 5.1. NO-ILC initialisation

Sample time, T_s	0.01 sec.
Simulation time, T_{sim}	50 sec.
Initial states, x	$[0\ 0]^T \in \mathbb{R}^{2\times 1}$
Initial ILC input, u_{ILC}	0
Number of ILC iterations, M	1000
Weight on the error, $\mathbf{W}_e$	$\rho\mathbf{I} \in \mathbb{R}^{N\times N}$
Weight on the system input, $\mathbf{W}_u$	$\mathbf{I} \in \mathbb{R}^{N\times N}$
ρ	0.001
λ	0.1

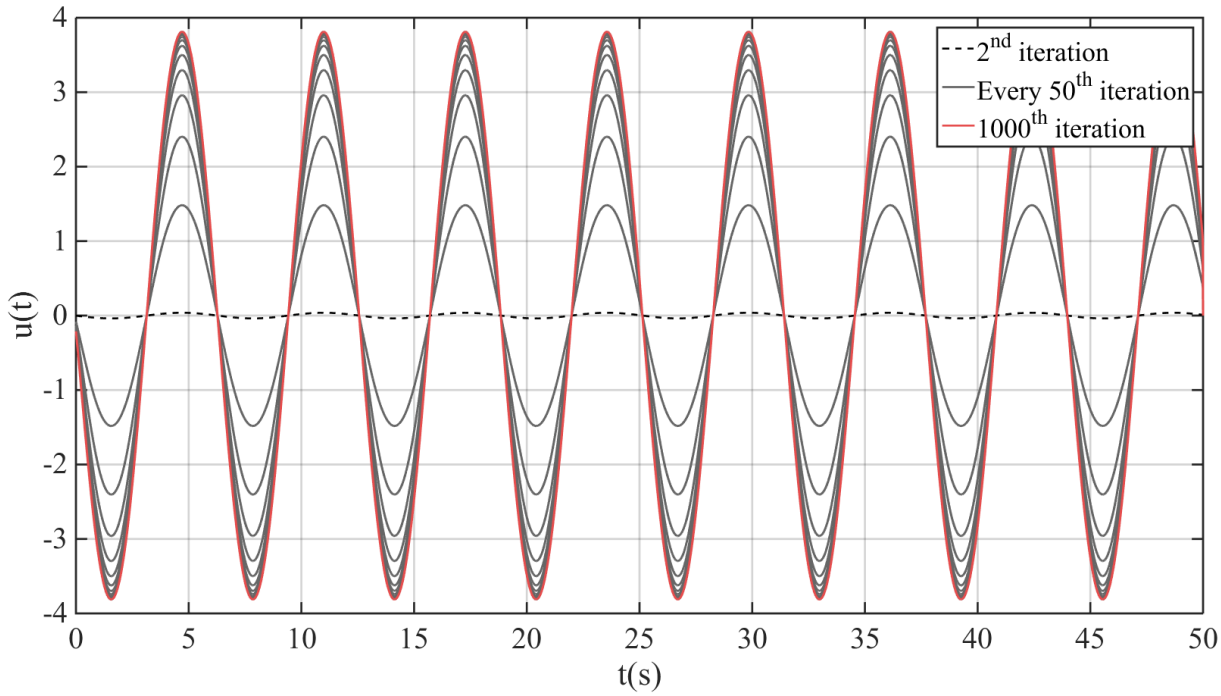

FIGURE 5.6: ILC inputs

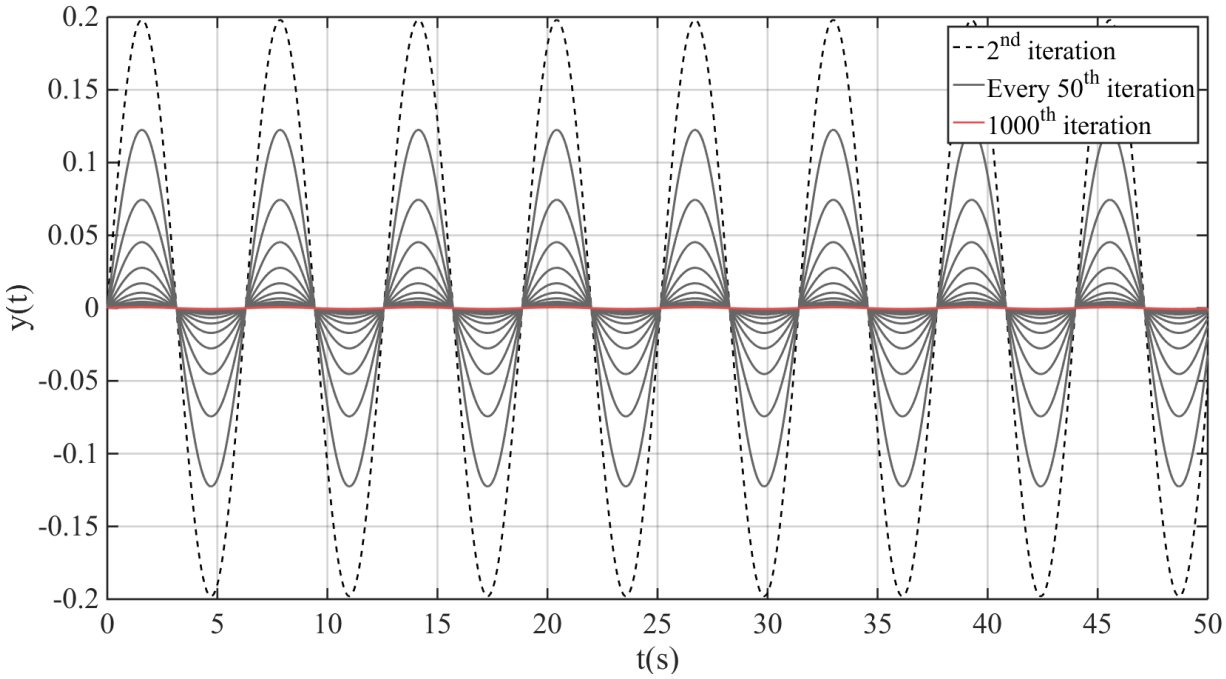

FIGURE 5.7: System outputs

The second step is to apply the identification layer of SOR-ILC. The frequency data of the converged ILC signal is the input at the last iteration in Figure 5.6 (i.e. $u^*(t)$). The result of the model approximation of $u^*(t)$ can be seen in Figure 5.8 where the peaks show approximated values of dominant frequencies $\overline{\omega}_i$. Since the disturbance was chosen as a sinusoidal signal, only the largest peak that is $\overline{\omega}_1 \approx 1 rad/s$ is sufficient for the design of the output regulator.

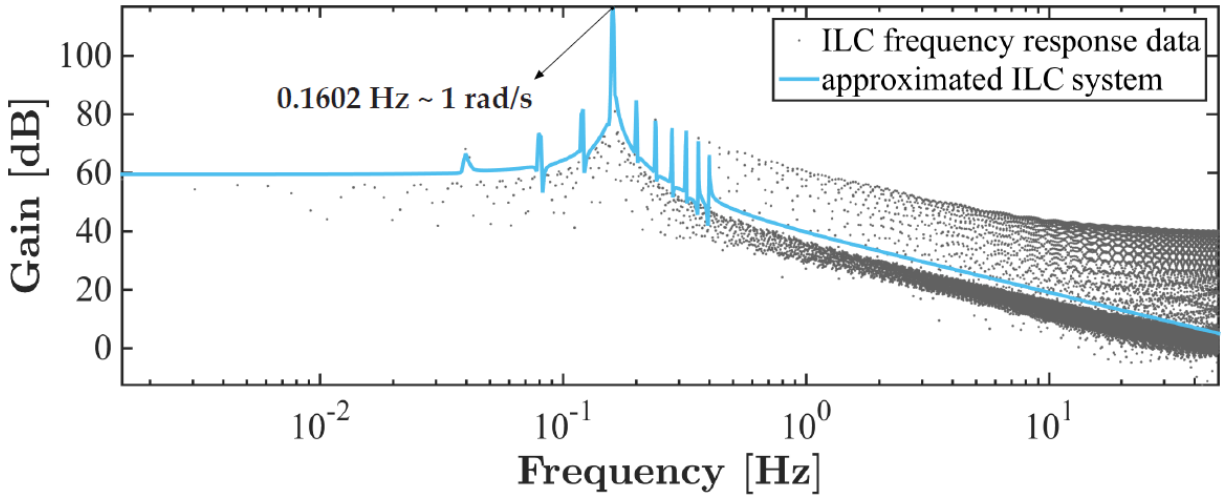

FIGURE 5.8: Model approximation of the ILC system

After detecting the approximate frequency content of the periodic disturbance, it is possible to build the output regulator part of the SOR-ILC as shown in the Layer 3 of Section 5.1.2. The first step is to build the internal model unit (5.10) using the identified frequencies $\overline{\omega}_i$. For the given disturbance $d(t)$, $\hat{\mathbf{S}} = \text{blkdiag}(\hat{S}_0, \hat{\mathbf{S}}_1)$ and $\Gamma = [\Gamma_0, \Gamma_1]^\top$ where $\hat{S}_0 = 0$, $\hat{\omega}_1 = \overline{\omega}$, $\Gamma_0 = 1$ and $\Gamma_1 = [0, 1]^T$. The second step is to build the stabiliser

unit given by (5.12). Since $d(t)$ is a single sine signal $\rho = 2$ such that $(\rho + 1)$ tuning parameters are needed for the output regulator. The values for these parameters can be set to following values : $\beta = 0.1$, $\mu_1 = 100$ and $\mu_2 = 100$ (note that these values are chosen by trial-and-error for a sample demonstration and other choices that lead to better rejection performance might be possible).

Additionally, for a better analysis, a phase shift of $\pi/4$ is added to the disturbance and the initial states of the inner model are set to some arbitrary values, i.e. $d(t) = 0.2\sin(t + \pi/4)$ and $x_0 = [0, 0.05]^\top$. Finally, the SOR-ILC performance can be tested against $d(t)$ by running the system in Figure 5.5 while the switch is at $S.2$. One can observe in Figure 5.9 that the disturbance amplitude appearing in the system output has been reduced by 99.6% after applying the output regulator of the SOR-ILC. The remaining small oscillations after $50sec$ are due to a small approximation error left from the ILC learning, $\overline{\omega}_1 \approx 1rad/s$.

Figure 5.10 demonstrate the control signal of SOR-ILC which is made up from two parts : u_{part1} and u_{part2} as can be seen in (5.11). The different shapes of u_{part1} and u_{part2} are results of the choices made for the tuning parameters β, μ_1 and μ_2. It can easily be observed that u_{part1} is more effective in the beginning whereas u_{part2} takes over for attenuation later on. u_{part1} uses the state information x from the inner model while u_{part2} depends on the internal model unit states η learned via ILC.

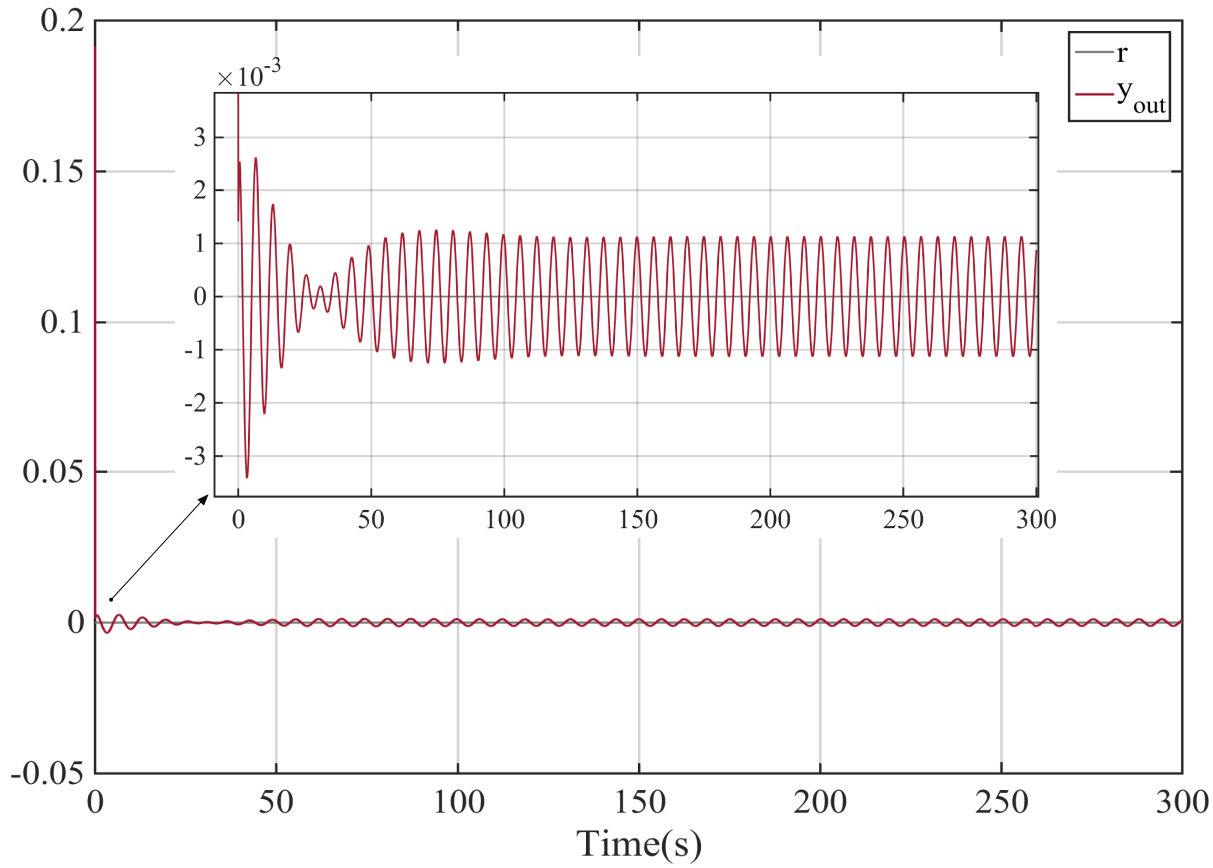

FIGURE 5.9: System output with SOR-ILC

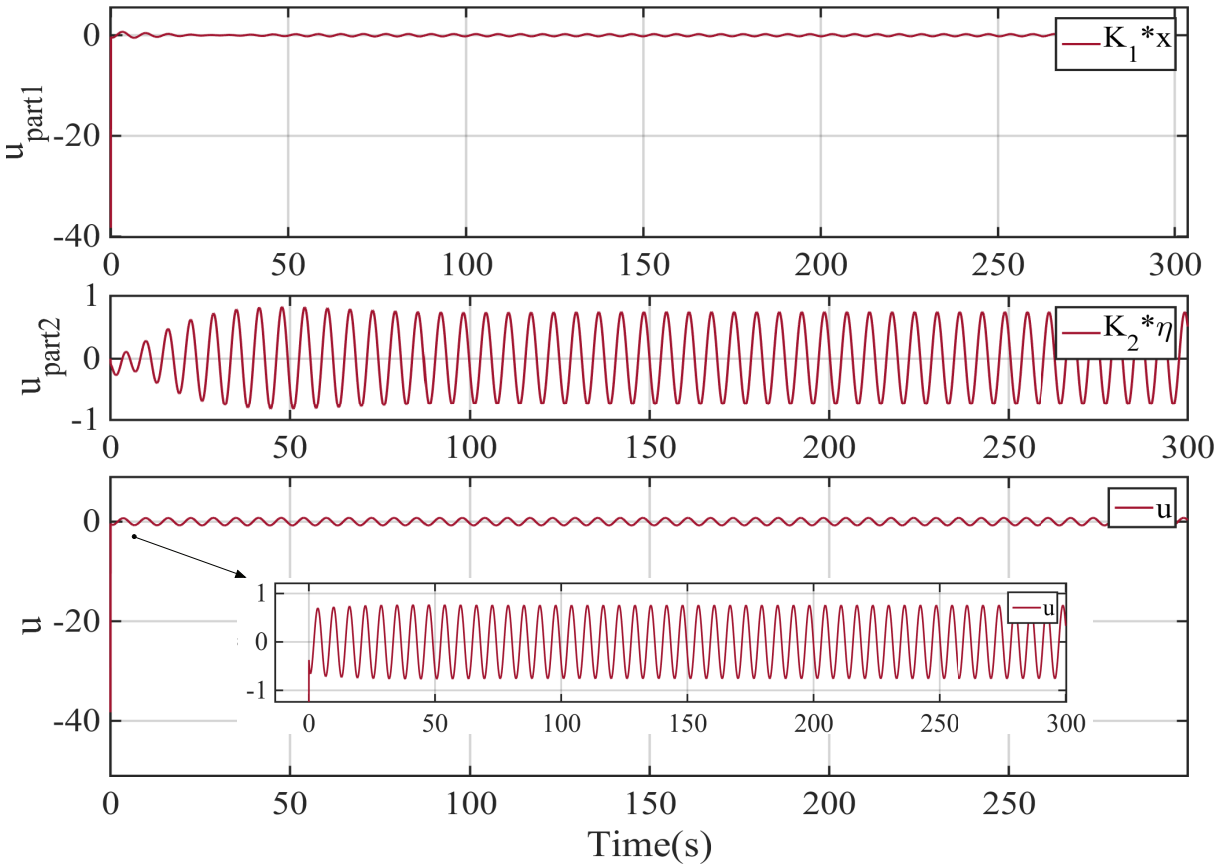

FIGURE 5.10: SOR-ILC control signals

Analysis 2 : Complex Disturbances and Uncertainty

The rejection performance of our SOR-ILC is now tested considering a lumped effect of complex disturbances and parameter uncertainty. For this numerical application, the inner system is updated with the following form :

$$\dot{x}(t) = \tilde{\mathbf{A}}(t) + \tilde{B}u(t) + [\sin x_1(t), 0]^\top, \tag{5.20}$$

$$y(t) = \tilde{C}x(t) + d(t). \tag{5.21}$$

Here, $\tilde{\mathbf{A}}$, $\tilde{B}$, $\tilde{C}$ are $\mathbf{A}$, B, C matrices in (5.1)-(5.2) with 20% parameter uncertainty, $x(t) = [x_1(t), x_2(t)]^\top$ and $x_0 = [0, 0]^\top$. The state disturbance is a sine function of the first system state and the output disturbance is in form of three non-linearly combined sine waves, i.e.

$$\begin{aligned} d(t) = 0.25\Big[\Big(&0.7(0.15 - 0.8\sin(\omega_1 t + \phi))^2 \\ &-0.6\sin(\omega_2 t + \phi)\Big)^3 - 0.35(\sin(w_3 t + \phi))^2\Big] \end{aligned} \tag{5.22}$$

where $\omega_1 = 0.27rad/s$, $\omega_2 = 0.76rad/s$ and $\omega_3 = 0.95rad/s$ are chosen to not have common divisors and $\phi = 0$ for the first analysis. SOR-ILC is created by following the procedure given in Section 5.1.2. First, ILC learns the periodic frequencies of the disturbances under the varying uncertainty between iterations (see Figure 5.11). It can be observed that

the maximum amplitude of $d(t)$ is already reduced by 86.8% via ILC only. Second, the identification process determines the number of frequencies to be used in output regulator as $N_f = 18$ and these frequencies are utilised for building the internal model unit of the output regulator as shown in Section 5.1.2. Then, the remaining step is to tune the stabiliser unit parameters considering the equations (5.10),(5.12), (5.16), (5.17). It can be seen in (5.12) that the number of needed tuning parameters become $\rho + 1 = 37$. For the simulation, β is set to 0.1 as before and all μ_i values are chosen to be 100.

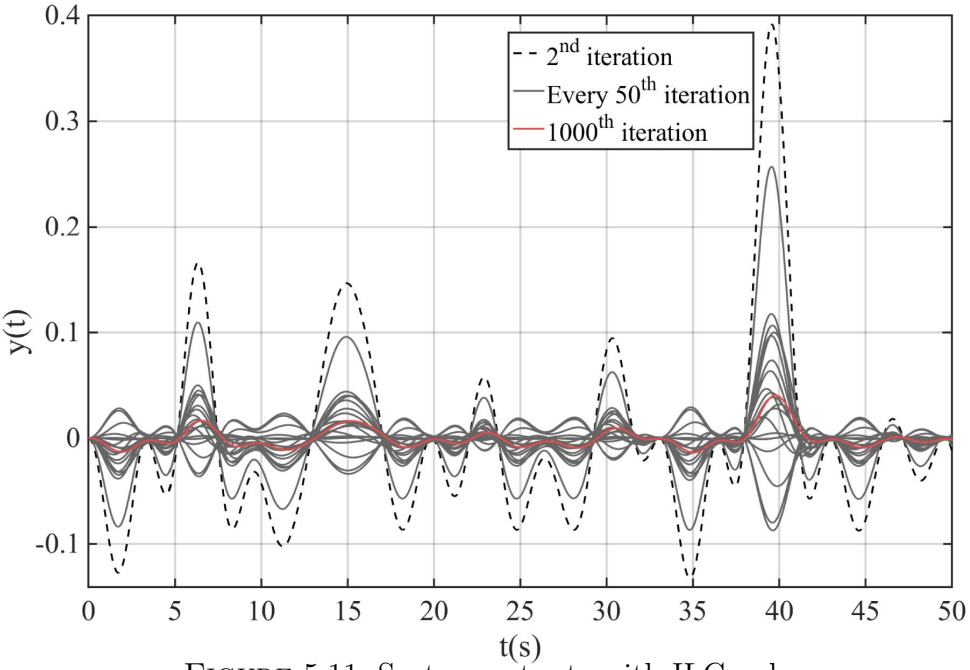

FIGURE 5.11: System outputs with ILC only

In addition, it is decided to analyse the effect of the number of disturbance frequencies used in creating the SOR-ILC. Therefore, the frequencies are put in the order of decreasing amplitude and 18 tests are carried out in total by adding a new frequency into SOR-ILC before each test. The results of these tests are shown in Figure 5.12. In the test 1, the amplitude of the disturbance is the highest since SOR-ILC uses only one singlefrequency. In the remaining tests, the amplitude of the disturbance approaches a smaller value as we include new frequencies in SOR-ILC. Furthermore, the final system output obtained in the test 18 reaches the same form of the signal calculated by ILC alone and it is less in amplitude which can be attributed to the feedback gains inside the output regulator. The maximum amplitude of the disturbance is observed to be 92.9% smaller than that obtained through ILC only. Another demonstration is done in Figure 5.13 by swithching the SOR-ILC on and off (this time $\phi = \pi/4$ in $d(t)$, $x_0 = [0, 0.2]^\top$ in (5.20) and SOR-ILC uses all the learned frequencies). One can observe that when SOR-ILC is switched on, the amount of disturbance attenuation highly increases which proves once again the efficiency of SOR-ILC.

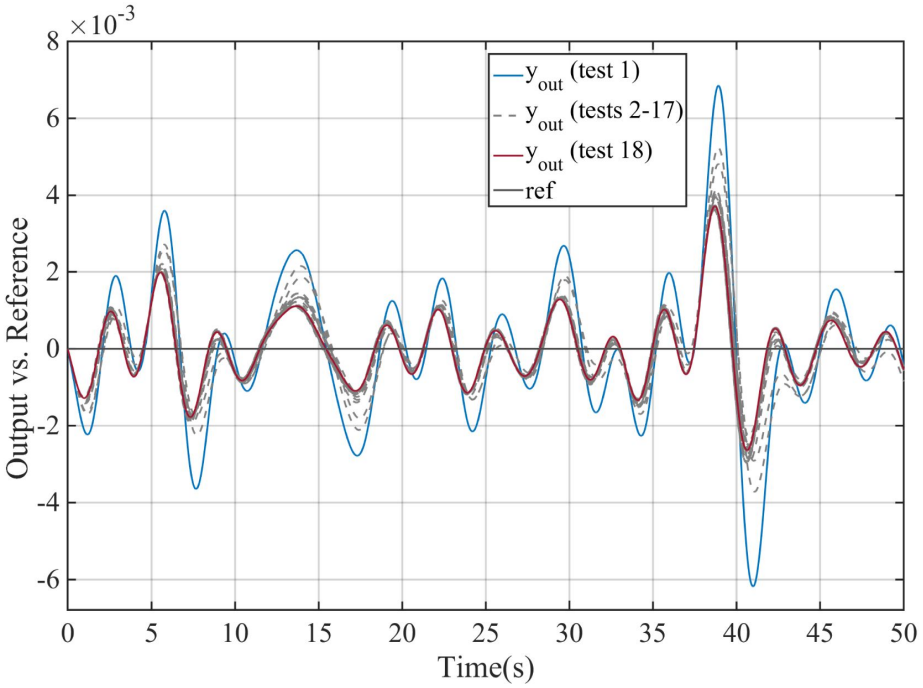

FIGURE 5.12: System outputs with SOR-ILC

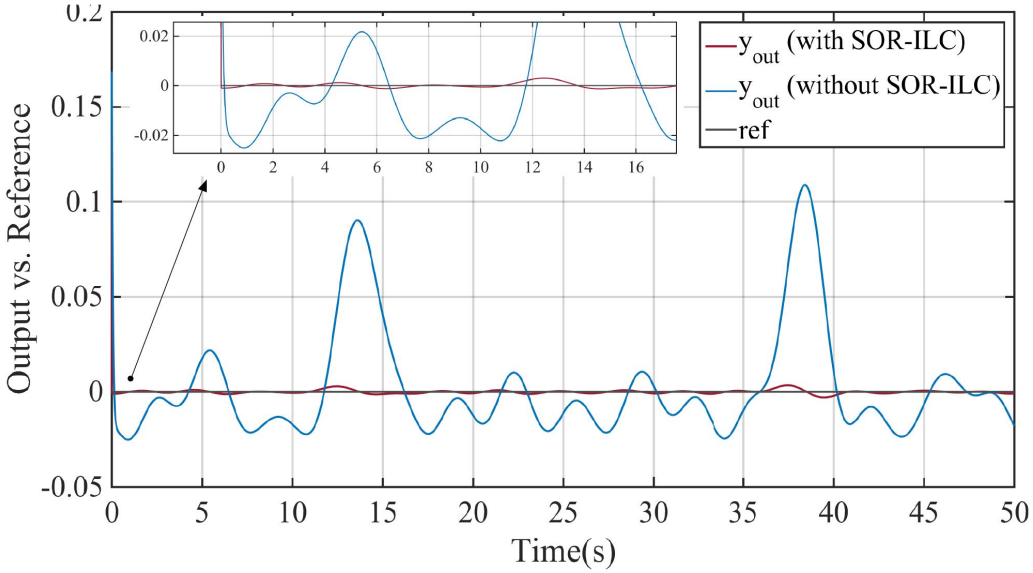

FIGURE 5.13: System outputs with SOR-ILC (on/off)

5.2 SOR-ILC for nonlinear systems under unknown complex periodic disturbances

In this section, the design and analysis of SOR-ILC are extended to a nonlinear system case. The disturbance rejection problem is also made practically more interesting by considering a Van der Pol (VDP) oscillator acting on one of the outputs. The VDP oscillator

creates a nonlinear disturbance [7] of which the properties of are assumed to be unknown.

5.2.1 Problem definition

Let us consider a nonlinear system in the form of a general chain of integrators which is written as

$$\begin{cases} \dot{x}_1(t) &= x_2(t) \\ &\vdots \\ \dot{x}_n(t) &= \varphi(t, x(t)) + u(t) \\ y(t) &= x_1(t) \end{cases} \tag{5.23}$$

where $x(t) = (x_1(t), \ldots, x_n(t))^\top \in \mathbb{R}^n$ is the state, $u(t) \in \mathbb{R}$ is the control input, and $y(t) \in \mathbb{R}$ is the output to be regulated and $\varphi(t, x(t))$ is assumed to be Lipschitz. For an easier analysis, it is useful to write the system (5.23) in the following compact form :

$$\dot{x} = \mathbf{F}x + B(\varphi(x, t) + u(t)), \tag{5.24}$$

$$y = Cx \tag{5.25}$$

where the matrices $\mathbf{F}, B, C$ are selected as

$$\mathbf{F} = \begin{pmatrix} 0_{n-1\times 1} & \mathbf{I}_{n-1\times n-1} \\ 0 & 0_{1\times n-1} \end{pmatrix} \; B = \begin{pmatrix} 0_{n-1\times 1} \\ 1 \end{pmatrix} \; C = \begin{pmatrix} 1 & 0_{1\times n-1} \end{pmatrix}$$

which is also called the *Bruwonsky canonical form*. For simplifying the explanations, let us consider (5.23) in second order and add a periodic disturbance in its output, i.e.

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= \varphi(t, x(t)) + u(t) \\ y(t) &= x_1(t) + w(t, x_v(t)) \end{aligned} \tag{5.26}$$

where $\varphi(t, x(t)) = sin(x(t))$ and $w(t, x_v(t))$ is an unknown nonlinear smooth periodic signal generated by the following VDP oscillator :

$$w(t, x_v(t)) = -x_{v,1}(t) + \mu_v(1 - x_{v,1}^2)x_{v,2} \tag{5.27}$$

The goals of rejection problem stay the same as in Section 5.1.1. In order to make the ILC application feasible to this type of system, one must ensure that given some initial conditions the system will remain initially stable along the iterations. Therefore, a two-step pre-stabilisation procedure is needed before applying the layers of SOR-ILC :

1. Stabilise the system (5.23), i.e. select $K_s = (a_1, \ldots, a_n)$ such that $\mathbf{F} - BK_s$ is stable.

7. In this thesis, the term nonlinear disturbance means a disturbance that is produced by a nonlinear system/oscillator

For (5.26), in particular, select a_1, a_2 such that

$$\mathbf{F} + BK_s = \begin{pmatrix} 0 & 1 \\ -a_1 & -a_2 \end{pmatrix}$$

is stable. This model is obtained by linearising the chain of integrators in (5.26).

2. Build an inner controller to ensure that the ILC system will not diverge in the presence of $w(t, x_v(t))$. In other words, apply the following control law :

$$u(t) = -\kappa^2 a_1 x_1 - \kappa a_2 x_2 = K_s K_0 x(t) \tag{5.28}$$

where $K_0 = \text{diag}(\kappa^2, \kappa)$. With this control law, the matrix to be studied becomes

$$\mathbf{A}_{in} = \mathbf{F} + BK_s K_0 = \begin{pmatrix} 0 & 1 \\ -k^2 a_1 & -k a_2 \end{pmatrix} \tag{5.29}$$

The proper selection of κ can be done by analysing the Lipshitz constant L_φ of $\varphi(t, x_v(t))$. The property is that for any $\delta > 0$ there exists a $L_\varphi > 0$ such that

$$|\varphi(t, s) - \varphi(t, r)| \leq L_\varphi |s - r|$$

for all s, r such that $|s| \leq \delta$ and $|r| \leq \delta$. To be precise, one can choose $\kappa > 2L_\varphi \bar{\lambda}(\mathbf{P})$ where $\bar{\lambda}(\mathbf{P})$ is the maximum eigenvalue of the matrix P yielded by solving the Lypaunov equation :

$$\mathbf{P}(\mathbf{F} + BK) + (\mathbf{F} + BK)^\top \mathbf{P} = -\mathbf{I}$$

The Lipshitz constant L_φ for (5.27) can be calculated as

$$\begin{aligned}
L_\varphi &= \sup_{|x| \leq \delta} \left| \frac{\partial \varphi}{\partial x}(t, x) \right| \\
&= \sup_{|x| \leq \delta} \left| \Big(-1 - 2\mu x_1 x_2,\ \mu(1 - x_1^2) \Big) \right| \\
&\leq \sup_{|x| \leq \delta} \left| \sqrt{(1 + 2\mu x_1 x_2)^2 + \mu^2 (1 - x_1^2)^2} \right| \\
&\leq \sup_{|x| \leq \delta} \Big(|(1 + 2\mu x_1 x_2)| + \mu |(1 - x_1^2)| \Big) \\
&\leq 1 + 2\mu\delta^2 + \mu(1 + \delta^2) \\
&\leq 1 + \mu + 3\mu\delta^2
\end{aligned}$$

Therefore, from my experience, it can be said that the Lipshitz constant can be picked as $L_\varphi = 2\mu$. Then κ should be chosen bigger than this value according to above mentioned criterion (i.e. $\kappa > 2L_\varphi \bar{\lambda}(\mathbf{P})$).

5.2.2 Control Design

The control designs of Layer 1 (ILC) and Layer 3 (output regulator) of SOR-ILC for the nonlinear system case are exactly the same as in Section 5.1.2. Therefore, they are not shown here again. However, another identification method is utilised for the nonlinear case which can be seen below.

Layer 2 : Identification method via Hankel

The identification method called *Hankel* is selected due to the fact that it does not require a *Fast Fourier Transformation (FFT)* for yielding an accurate model approximation of the converged ILC signal $u^*(t)$. It directly uses time domain data and finds the approximate model of a system. What it basically does is to approximate the nonlinear system model by combining the outputs of many linear trasfer functions to the same impulse signal.

Let us now consider that the converged (and optimal) ILC control signal $u^*(t)$ has been obtained in Layer 1. Since the disturbance to be rejected is periodic, $u^*(t)$ is periodic as well. Therefore, following the same philosophy as in [7], one can seek for the dominant harmonics of this optimal open-loop signal and the use these harmonics in the IMC design of Layer 3.

Consider now $u^*(t) \equiv u^*(t_k)$, where $k = 1, \ldots, N = 2n$, as a sampled signal with constant period t_s. On this basis, one can construct the Hankel matrix $\mathcal{H} \in \mathbb{R}^{n\times(n+1)}$ as

$$\mathcal{H} = \begin{bmatrix} u^*(t_1) & u^*(t_2) & \ldots & u^*(t_{n+1}) \\ u^*(t_2) & u^*(t_3) & \ldots & u^*(t_{n+2}) \\ \vdots & \vdots & \vdots & \vdots \\ u^*(t_n) & u^*(t_{n+1}) & \ldots & u^*(t_{2n}) \end{bmatrix}. \tag{5.30}$$

Rooted on $\mathcal{H}$, let us define the following quadruple (E, A, B, C) [8]

$$\begin{array}{rcll} \mathbf{E} & = & \mathcal{H}_{1:n,1:n} & \in \mathbb{R}^{n\times n}\ , \\ \mathbf{A} & = & \mathcal{H}_{1:n,2:n+1} & \in \mathbb{R}^{n\times n}\ , \\ B & = & \mathcal{H}_{1:n,1} & \in \mathbb{R}^{n\times 1}\ , \\ C & = & \mathcal{H}_{1,1:n} & \in \mathbb{R}^{1\times n} \end{array} \tag{5.31}$$

that constitutes the raw model encoding the data generated by the optimal control signal $u^*(t_k)$. This model is linear time-invariant (LTI) and discrete-time, with the same sampling period t_s, and reads

$$\mathbf{E}x(t_k+1) = \mathbf{A}x(t_k) + Bu(t_k) \text{ and } u^*(t_k) = Cx(t_k) \tag{5.32}$$

with a non-zero initial condition $\mathbf{E}x(t_1) = B$ and where $x(t_k) \in \mathbb{R}^n$ and $u(t_k) \in \mathbb{R}$. As the

8. Here we denote the raw vector from element 1 to n of $\mathcal{H}$ as $\mathcal{H}(1, 1:n)$, the column vector from element 1 to n as $\mathcal{H}(1:n, 1)$.

above model is of dimension n, a reduced model and associated fundamental oscillations can be computed by a model reduction approach following the same approach as the *Loewner* one e.g. presented in [5, 60] or any *LTI* oriented one (see e.g. [8] book or [90] monograph for rather complete insight). The procedure involved here is the so-called *Pencil method*, which allows to obtain dominant sub-models by using a rank revealing factorisation such as the *Singular Value Decomposition (SVD)* one. Then, by first performing

$$\mathbf{U}_1\mathbf{S}_1\mathbf{V}_1^T = \text{SVD}([\mathbf{E}^T\ \mathbf{A}^T]^T) \tag{5.33}$$

$$\mathbf{U}_2\mathbf{S}_2\mathbf{V}_2^T = \text{SVD}([\mathbf{E}\ \mathbf{A}]) \tag{5.34}$$

and by selecting the $X \in \mathbb{R}^{n\times r}$ and $Y \in \mathbb{R}^{n\times r}$ projectors as

$$\mathbf{X} = \mathbf{U}_2(1:n, 1:r) \text{ and } \mathbf{Y} = \mathbf{V}_1(1:n, 1:r), \tag{5.35}$$

the quadruple $(\hat{\mathbf{E}}, \hat{\mathbf{A}}, \hat{B}, \hat{C}) = (\mathbf{X}^T\mathbf{E}\mathbf{Y}, \mathbf{X}^T\mathbf{A}\mathbf{Y}, \mathbf{X}^T B, C\mathbf{Y})$ leads to the discrete-time reduced dynamical model of dimension r,

$$\hat{\mathbf{E}}\hat{x}(t_k+1) = \hat{\mathbf{A}}\hat{x}(t_k) + \hat{B}\hat{u}(t_k) \text{ and } \hat{u}^*(t_k) = \hat{C}\hat{x}(t_k) \tag{5.36}$$

where $\hat{x}(t_k) \in \mathbb{R}^r$ and $u(t_k) \in \mathbb{R}$, and with initial conditions $\hat{x}(t_1) = X^T x(t_1)$. On the basis of this reduced quadruple, one may obtain the partial fraction expansion of the underlying associated transfer function $\hat{\mathbf{H}}(z) = \hat{C}(z\hat{\mathbf{E}} - \mathbf{A}r)^{-1}\hat{B}$, which impulse response reads

$$\hat{u}^*(t_k) = \sum_{i=1}^{r} \alpha_i e^{\sigma_i t_k} e^{\imath(\omega_i t_k + \theta_i)} \tag{5.37}$$

$$= \sum_{i=1}^{r} \hat{h}(\alpha_i, \sigma_i, \omega_i, \theta_i) \tag{5.38}$$

where $\alpha_i \in \mathbb{R}$ is the real amplitude and $\theta_i \in \mathbb{R}$ the phase angle. Then, the eigenvalues are expressed as $\lambda_i = e^{\sigma_i + \imath\omega_i} \in \mathbb{C}$ where $\sigma_i \in \mathbb{R}$ is the decay rate and $\omega_i \in \mathbb{R}$ the frequency (in *rad/s*) of the i-th oscillation (see [7] for additional comments). In general, increasing r leads to a perfect matching of the original signal $u^*(t_k)$.

In the presented case, as the signal $u^*(t_k)$ mainly contains oscillatory behaviour, one will practically observe $\sigma_i = 0$.

On this basis, it is now easy to sort the frequencies ω_i as a function of the amplitude of the oscillations α_i. This naturally leads to the set of frequencies that will be used in the IMC design in Layer 3.

Remark 5.1 (Selection of the approximation order r) : *Obviously in the above procedure, the order r of the approximated model $\hat{\mathbf{H}}$ has an importance. The higher r is, the more accurate the impulse response of the reduced model* (5.37) *is. However, a large r will embed many harmonics and thus lead to a more complex controller structure. One way to limit the number of harmonics (as proposed in [7]) is to monitor the angles between each harmonics. This latter may be computed as follows :*

$$\angle(u, v) = \mathbf{arccos}\frac{\langle u, v\rangle}{||u||||v||} \tag{5.39}$$

where u and v vectors are two consecutive harmonics obtained by evaluating

$$u = \hat{h}(\alpha_o, \sigma_o, \omega_o, \theta_o) \text{ and } v = \hat{h}(\alpha_p, \sigma_p, \omega_p, \theta_p)$$

along the time indices t_k for $\{o, p\} = \{1, ..., r$ and $||\omega_o|| \neq ||\omega_p||\}$. An angle close to $\pi/2$ leads to orthogonal signals u and v and thus an important information addition. Then, one may stop if $||\angle(u, v) - \pi/2|| > \epsilon$.

Now one may move on to the final step of the proposed procedure, namely the construction of the state feedback controller thought internal model control synthesis.

5.2.3 Numerical Analysis

Let us consider the nonlinear system (5.26) with $\varphi(t, x(t)) = sin(x(t))$ and the unknown nonlinear disturbance (5.27) where the Van der Pol (VDP) oscillator $w(t, x_v(t))$ with $\mu = 1$ produces the disturbance shown in Figure 5.14. The structure of the simulator in Figure 5.5 in this numerical analysis differs by the inner system in which the plant is replaced with (5.26) and the prestabilising feedback gain K_0 is calculated by the steps given in 5.2.1.

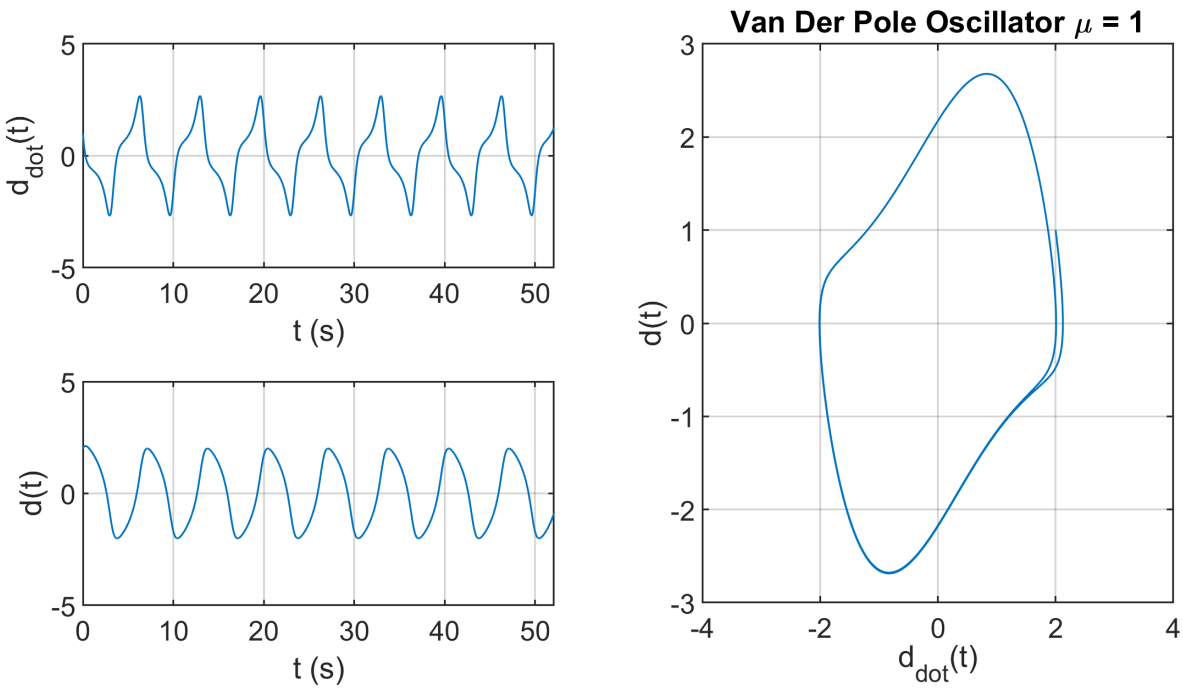

FIGURE 5.14: Disturbace generated by the Van der Pol Oscillator

The first step of SOR-ILC is to pre-stabilise the system (5.26) with the inner controller (5.28). To meet this purpose, the inner controller gain parameters in K_0 are selected as per $\kappa = 2\mu + 2$ following the precedure given in Section (5.2.1). This choice will guarantee the nondivergence of the system during the ILC application of SOR-ILC's first layer. Next, the initialisation parameters of ILC are set to the same values as shown on Table 5.2. The convergence related parameters ρ, λ and M are kept the same as in the previous numerical

application with norm optimal ILC (NO-ILC) since they provide sufficient performance for the demonstrating the functioning of this framework (running $M = 1000$ iterations in our case takes only few minutes). It should again be noted that achieving the best tuning for these ILC parameters is not the focus of the presented work here and one may probably find even better ILC methods to get faster or more precise results. To move on, the simulation model 5.2 is run with these parameters while the switch is at $S.1$ and the ILC inputs and outputs as a result of this process can be viewed in Figure 5.15 and Figure 5.16, respectively. It can be observed that ILC reduces the amplitude of the unknown disturbance to a very high precision and leaves oscillations roughly between $[-0.005, 0.005]$, thus making it more interesting for the identification process in layer 2.

TABLE 5.2. NO-ILC initialisation

Sample time, T_s	0.01 sec.
Simulation time, T_{sim}	52.07 sec.
Initial states (nonlinear Plant), x	$[0\ 0]^T \in \mathbb{R}^2$
Initial states (VDP Oscillator), x_v	$[2\ 1]^T \in \mathbb{R}^2$
Initial ILC input, u_{ILC}	0
Number of ILC iterations, M	1000
Weight on the error, $\mathbf{W}_e$	$\rho\mathbf{I} \in \mathbb{R}^{N\times N}$
Weight on the system input, $\mathbf{W}_u$	$\mathbf{I} \in \mathbb{R}^{N\times N}$
ρ	0.001
λ	0.1

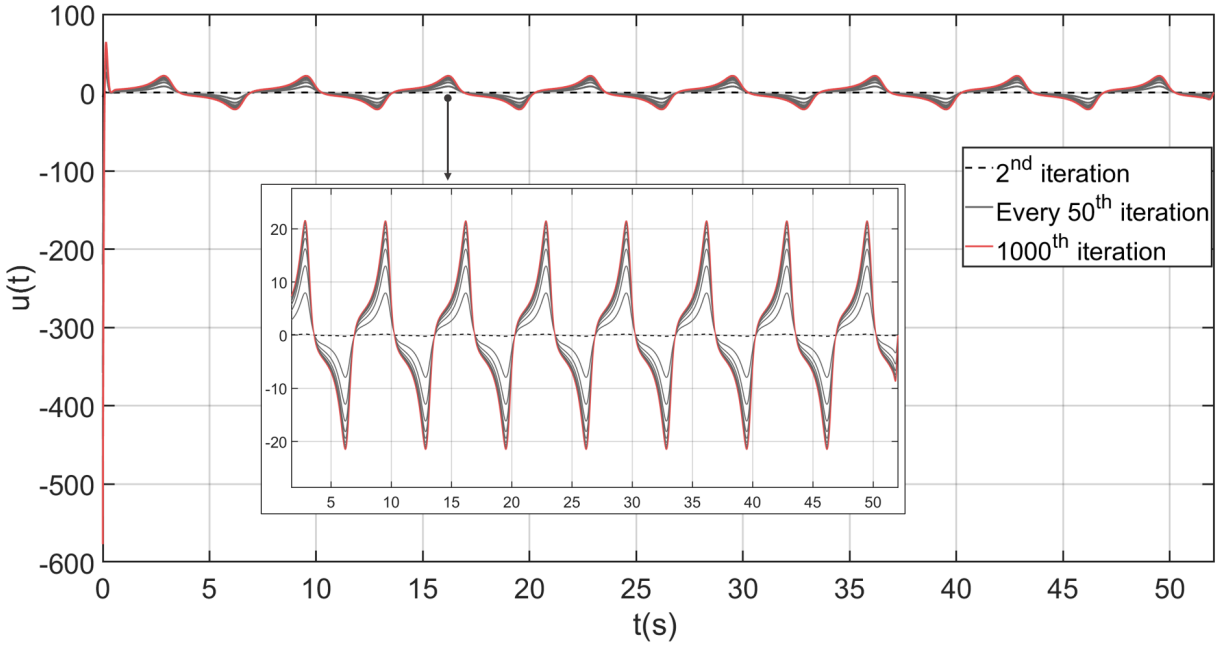

FIGURE 5.15: ILC inputs

The second step is to apply the Layer 2 of SOR-ILC. The model fitting on the converged ILC input data (i.e. $u^*(t) = u_{M=1000}(t)$) is carried out by the *Hankel* identification method

given in Section 5.2.2. The parameter r in this identification algorithm denotes the number of frequencies to be extracted from the data and it is chosen as $r = 18$ for a very close model fit to $u^*(t)$(see Figure 5.17 for the model fit and the extracted harmonics and Figure 5.18 the eigenvalues of the approximated model on the unit circle).

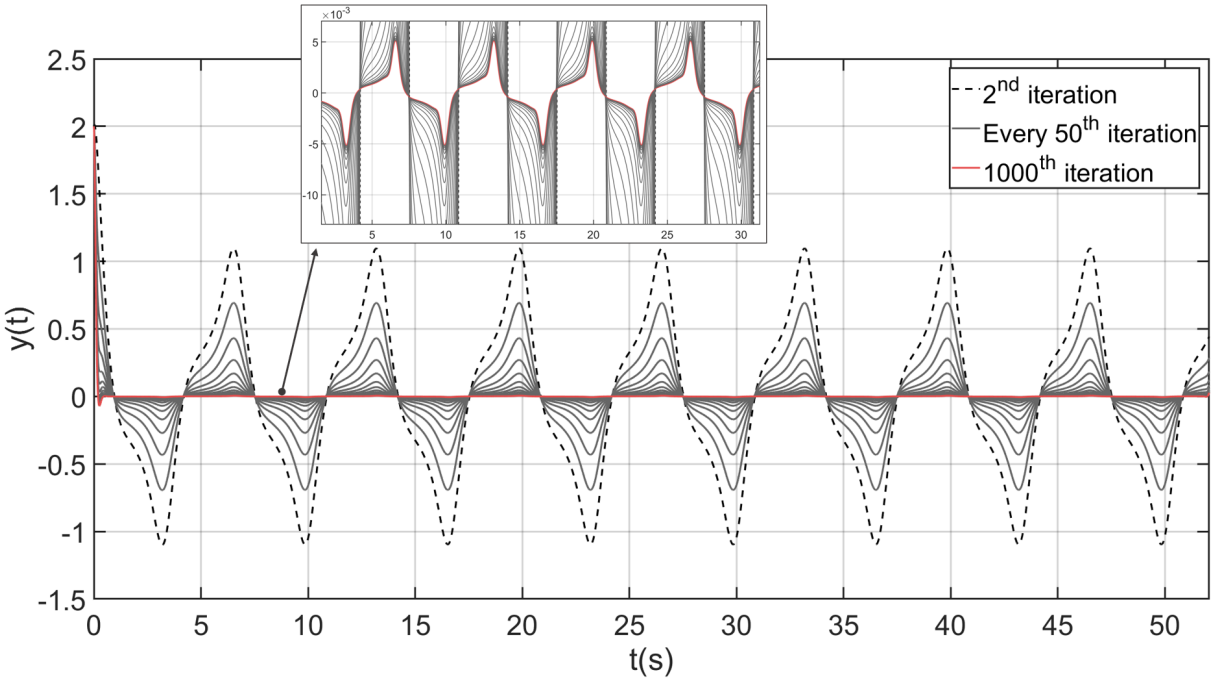

FIGURE 5.16: ILC outputs

As it was previously pointed out, what makes the given identification method interesting and motivating is that it does not need frequency domain data of a signal to extract its harmonics. The only requirement is to have the time domain data of the signal which means that it eliminates the step of Fast Fourier Transformation. Another advantage of the Hankel based identification is that it allows to monitor the angles between each harmonic and thus to reduce the the number of harmonics needed for the proper model approximation and controller design (see Remark 5.1). In other words, this feature can be used as a stop meachanism within the algoritm and eliminate the excessive information that does not contribute further to the model fitting precision.

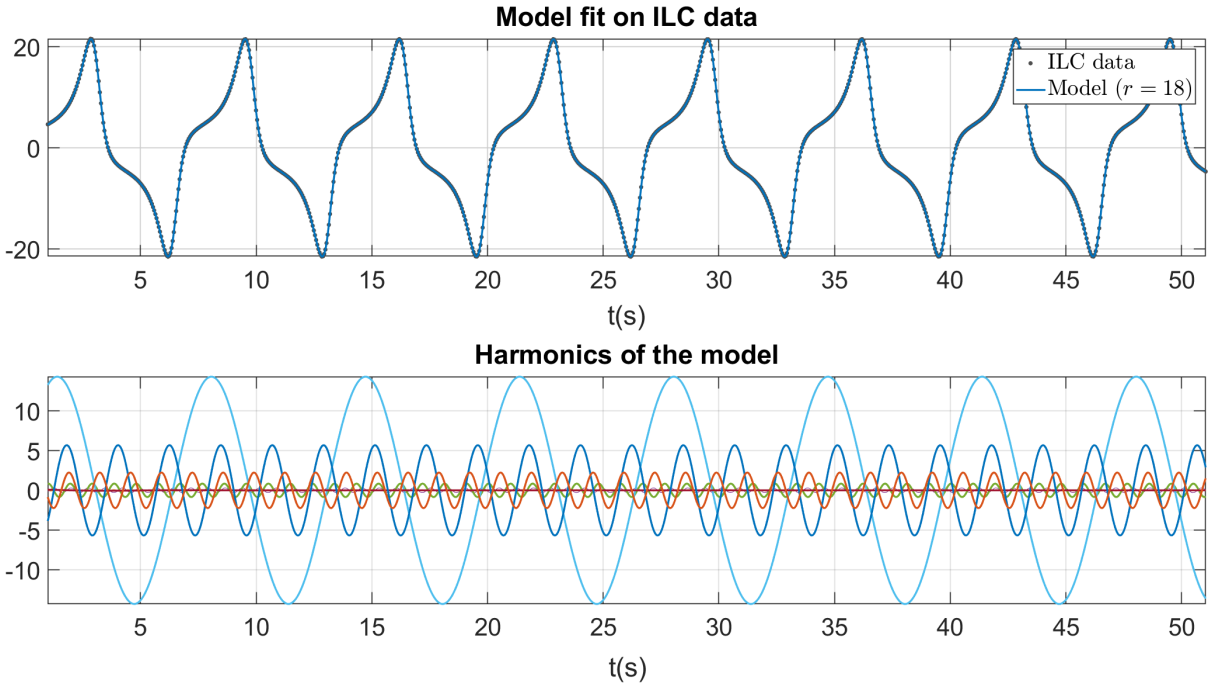

FIGURE 5.17: Identification of the ILC system via Hankel : Model fit and its harmonics

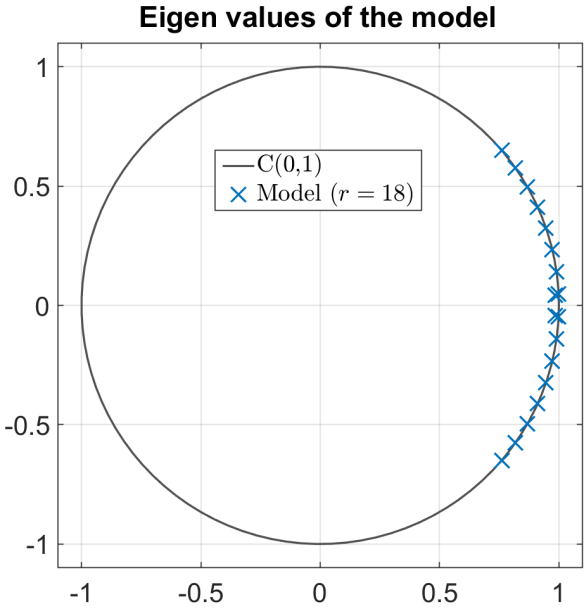

FIGURE 5.18: Eigenvalues of the model fit

Once the frequency content of $u^*(t)$ is obtained, it can be used in building an internal model unit for the output regulator described in Section 5.1.2. The procedure is exactly the same and the only difference comes from the stabiliser unit parameters where now $\beta = 0.00001$ and all μ_i are set to 10. It should be noted that these choises do not provide high performance; however, they are sufficient for demonstrating the functioning of the triple layer control logic (i.e. the transformation from data-based logic to conventional logic in control design) on a nonlinear system under a nonlinear disturbance. The tuning of β and μ_i can be considered as another research topic.

The simulation results[9] of the output regulator of SOR-ILC can be observed in Figure 5.19 and Figure 5.20 which represent the output and the input (with its subparts), respectively. It can be seen that the SOR-ILC process succesfully generates a feedback action utilising the internal model control created based on learning from data. However, compared to the results shown in Section 5.1.3 for the linear system case, the output regulator of SOR-ILC this time, with the given nonlinear system application, cannot outperform the results of ILC-only case in Layer 1. This can be related to the not fully accurate tuning of the output reglator. For improved results, it would probably needed to have a better insight on the interrelation of β and μ_i and the direct effects of both (and all μ) on each harmonic separately. There can also be issues related to the nonlinear system and the Van der Pol oscillator because when all μ_i values are chosen over 10 the system begins to diverge.

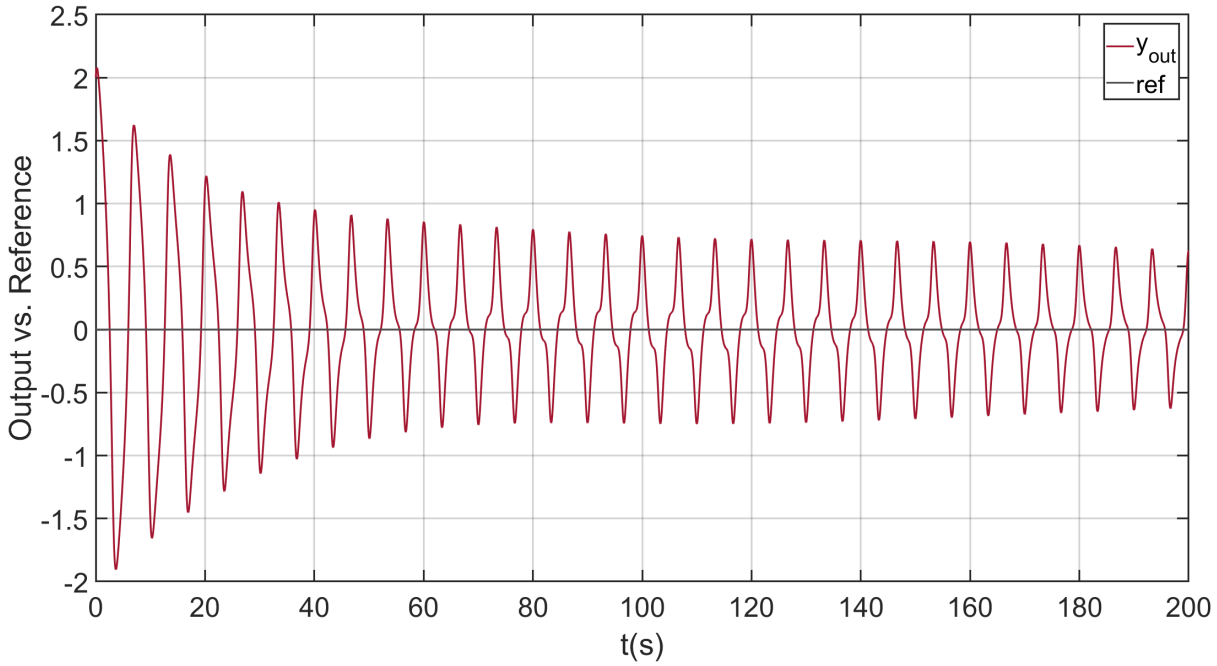

FIGURE 5.19: System output with SOR-ILC

9. Note that the simulation time for this step is set to 200s.

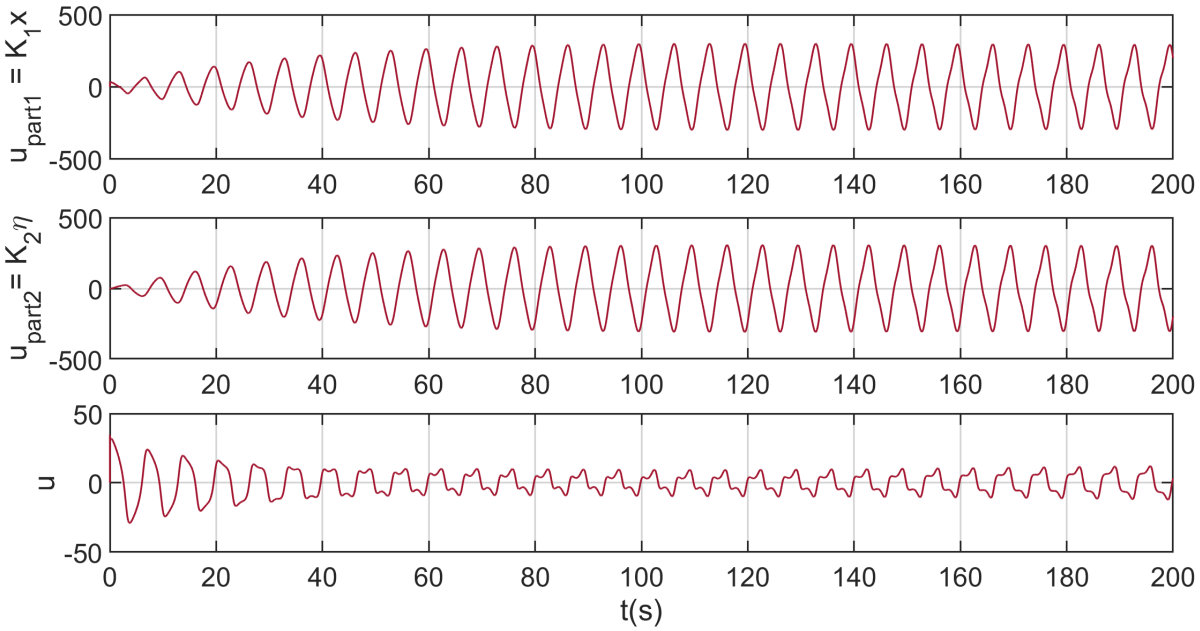

FIGURE 5.20: SOR-ILC control signals

5.3 Conclusion

This chapter can be considered as the main contribution of the thesis work. The main idea that has been presented is the *triple layer control approach* which allows a combination of learning-type control and non-learning-type control utilising an intermediate layer of data-based identification. The given concept is not limited to ILC and output regulation as choises for Layers 1 and 2, respectively. The main goal is to motivate the designers to consider such an approach that benefits from the best features of both control family and that fixes the disadvantages of both. For example, with the model-free nature of ILC, one can get rid of the necessity of knowing the frequencies periodic disturbance in order to build a controller. On the other hand, with the robust output regulation (feedback based on IMC), it is possible to cover the one of the major flaws of ILC which is the lack of robustness to nonrepetitiveness in the data due to instant disturbances, noises and parametric uncertainty (see Figure 5.1 and Figure 5.3 again for more details).

It is particularly important to realise that it was not so difficult to extend the SOR-ILC approach from linear systems to nonlinear systems. This has shown that the triple layer approach is not limited to simple linear systems and it can well be applied to nonlinear ones while keeping the overall control design relatively simple, compared to conventional non-learning-type methods even if the tuning still remains a complex process.

Moreover, in the last section of this chapter a time-domain identification method has been introduced which can be counted as another contibution of the thesis. This identification method is particularly interesting since it directly uses the time-domain data (period of the signal) to approximately find a linear model fit (i.e to extract dominant frequencies). A powerful and simple identification method (Layer 2) has crucial importance since it constitutes a mid-layer between learning-type control (Layer 1) and non-learning type

control (Layer 3). If the frequency extraction is not accurate, this has a direct effect on the Layer 3 since the internal model unit of the output regulator is built using the information acquired through the identification layer.

A future work for the presented approach can be done by focusing on the tuning of the output regulator parameters β and μs (in case SOR-ILC is chosen as the triple layer control). The lack of proper research on the tuning of these parameters can be related to the drop of performance occured while passing from Layer 1 results to Layer 3 results in Section 5.2 (nonlinear system case). Apparently, the tuning of the output regulator gets more complex when the plant is nonlinear (in linear system application in Section 5.1, a rough tuning improved even further the overall precision after ILC). This can be seen as an interesting issue to be looked at in more detail in a future work.

As stated in the previous chapters, it is always interesting to test different NO-ILC tunings or simply different ILC methods for Layer 1 of SOR-ILC. Since the goal of the thesis was to demonstrate new procedures (workflows) and frameworks, the tuning or theoretical improvement of the ILC methods was out of scope. However, having shown the proper functioning of the presented frameworks, one can carry out a reasearch on ILC and then test the same framework to achieve more satisfying (faster) results.

Finally, choosing different methods than ILC and the output regulation or even the identification in the proposed *triple layer approach* has not been tested yet and it might be of interest for future research. Thus, it remains rather interesting to see that this framework can make other combinations possible and produce results that are as good as the ones of SOR-ILC. One can refer to Table 1.1 for thinking of other combinations of learning-type and non-learning-type control methods.

General Conclusion

In this thesis, it has been shown that ILC can be used as a powerful tool in bettering the operation and design of existing conventional control approaches. The main objective has been to use ILC for building feedback controllers. In other words, the thesis work showed how an open-loop data-based method can be converted into a closed-loop feedback method in an approximative way. Moreover, it can be viewed that high precision reference tracking (Chapter 3) and disturbance rejection (Chapters 4 and 5) have both been analysed throughout the thesis. However, the major focus has been given to the rejection of unknown complex periodic disturbances.

The future perspectives of the thesis, the author's critiques on some challenging points and his suggestions for further improvements are provided in the sections below that cover the judgements on Chapter 3, Chapter 4 and Chapter 5 (i.e. the four contributions achieved throughourthe thesis).

Author's perspective and critiques

Chapter 3 : Contribution 1

In Chapter 3, the UAV experimentation with ILC has shown how a simple learning controller can make the UAV reach close to its highest agility allowed by its dynamics. The chapter has presented an application of norm optimal ILC (NO-ILC) on a real indoor UAV. The feasibility and efficiency of NO-ILC has been tested and good tracking results have been obtained during the experiments. A highly motivating point was that all the tracking results were obtained under nonlinearities and uncertainties. Basically, the NO-ILC algorithm allowed to achieve close to the maximum agility of the UAV and a large amount of precision has been achieved in only four flights. These results have been rather encouraging for using NO-ILC as a tool in the rest of the chapters and the chapter itself contributed to the development of the thesis as an initial study.

Apart from this, the main contribution of Chapter 3 to the literature can be seen as the proposed experimental procedure shown in Figure 5.21 recalled above. This new data flow has demonstrated that ILC experiments with UAVs can significantly benefit from a hybrid ILC update procedure that reduces the total time spent for the whole experiment. The core idea of this procedure is based on using a hybrid ILC update, i.e. integrating a large amount of predicted (simulated) flight data while performing the ILC updates, instead of directly using the real data as it can be seen on the left half of Figure 5.21 (refer to Section 3.3.1 for more details). By applying this experimental procedure on three different trajectories, it was possible to reach a large amount of tracking performance in only 4 flight experiments (note that this result is for the given specific flight experiments). During this application, the NO-ILC integration was done in open-loop to the UAV's already existing closed-loop system controlled by a feedback controller which has demostrated that ILC

can be used as tool for increasing the performance of a real nonlinear system such as a UAV. Note that the inner controller was a weak one in terms of tracking performance but a fairly good one in terms of robustness. This was an intentional choice that was made in order to test whether or not ILC can improve the performance of a bad feedback controller for tracking while at the same time benefiting from its robustness feature.

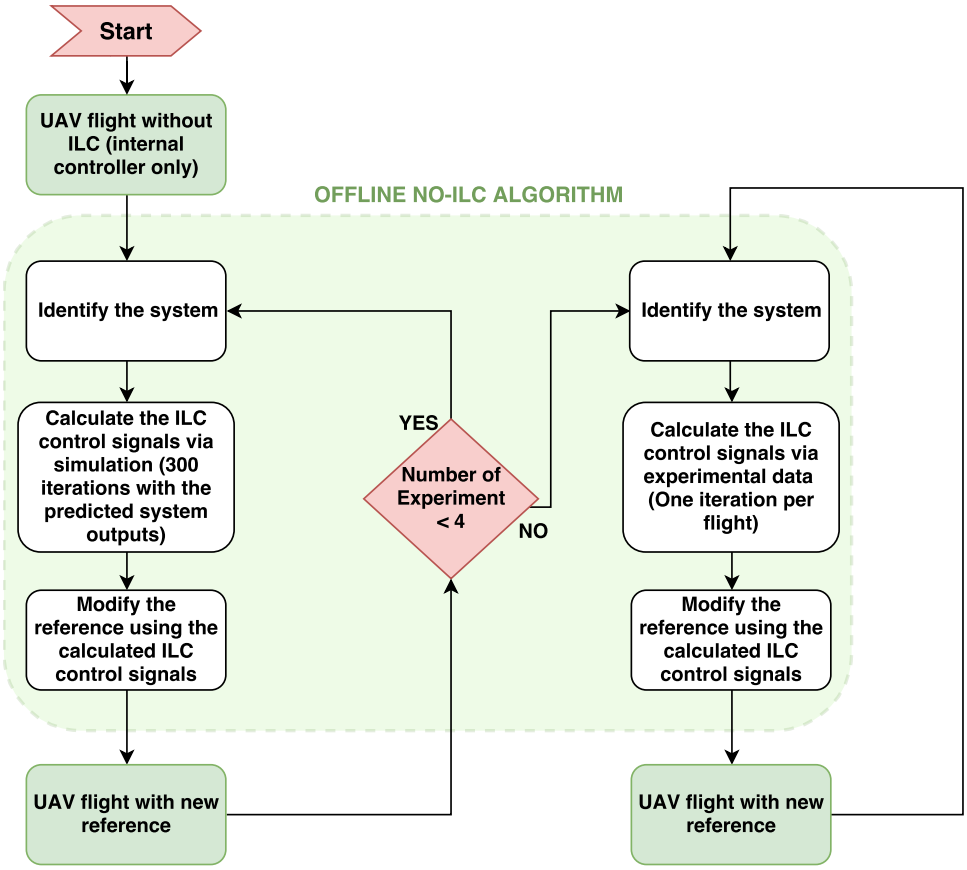

FIGURE 5.21: Proposed experimental procedure for NO-ILC experiment with UAVs (Recalled from Chapter 3)

One of the challenges observed during the experiments can be seen as the increased nonlinearity and parametric uncertainty in the UAV's dynamics. These two adverse effects had an impact on the learning of ILC as the UAV was reaching closer and closer to the given agile reference (ILC was basically pushing the UAV towards its dynamic limits in order to improve tracking precision at each fight experiment). This phenomenon can be observed in Figure 5.22 which depicts the model accuracy versus model update during reidentification using ILC.

An improvement for resolving this issue can be done first by trying to use higher-order identified transfer functions during the simulated ILC updates. Another improvement can come from changing the UAV and apply the same experiment. This is reasonable since the UAV used during the experiments had limited agility. Finally, the ILC method that was used can be changed or a different tuning can be applied. If the measurement noise and disturbances can be maintained well, some D-type or PD-type ILCs may also be tried as better candidates. However, it would be suggested to rather choose a more sophisticated

optimisation approach inside the NO-ILC (the one used in the application is based on the minimisation via Lagrange multiplier technique which is analytical).

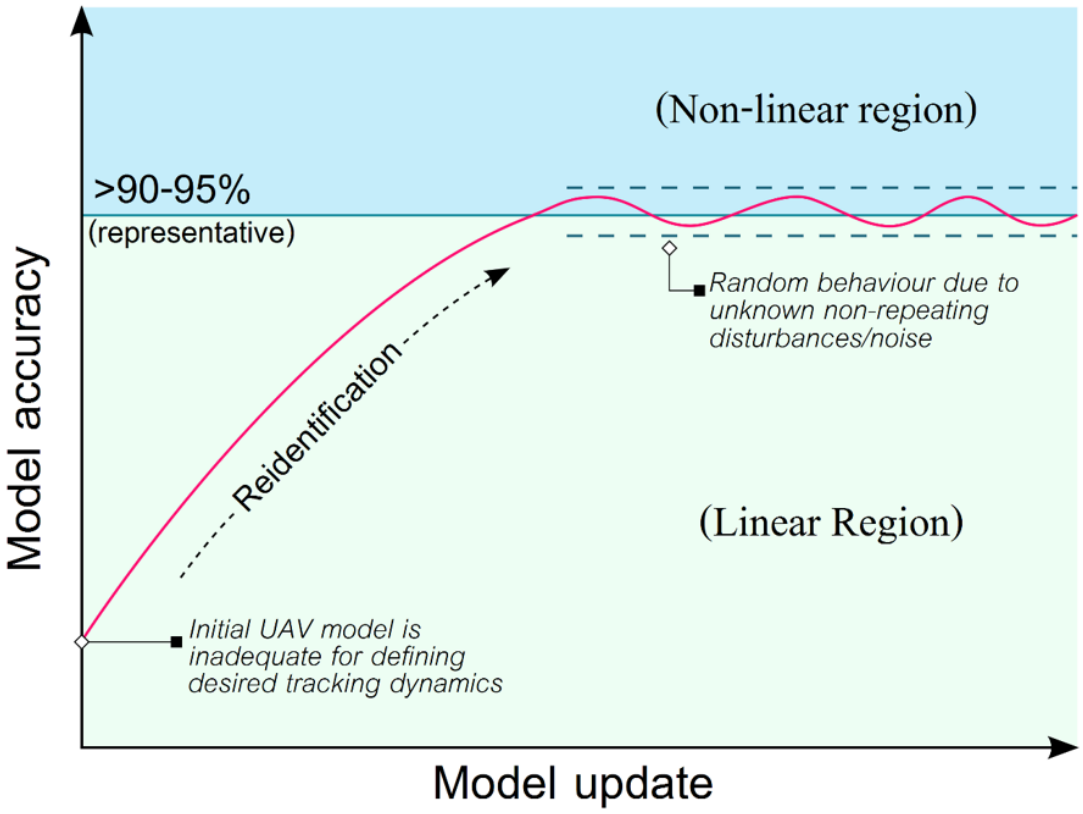

FIGURE 5.22: Proposed experimental procedure for NO-ILC experiment with UAVs (Recalled from Chapter 3)

Furthermore, it is also possible to address the issue of reduced learning due to nonlinearities after several flights by trying the switch to a nonlinear ILC method from the linear ILC (i.e. the one used during experiments). It may improve convergence speed ; yet, it is generally required that there is some well-known information about these nonlinearities (or uncertainties, disturbances etc.).

Another point that is worth to mention is related to the offline nature of the ILC. The term offline signifies here that after each real flight trajectory tracking, the UAV has to stop flying (or wait in hovering position) and wait for ILC to compute the next system input. This is quite normal since ILC is a method developped for the need of improving batch processes, i.e. discontinuous processes (e.g. reference tracking during a pick-and-place of robotic arm). However, it can be quite useful for a UAV to be able to carry out an online computation during its flight. In other words, the ILC signal for the next flight can tried to be calculated while the UAV is performing the current flight. Of course, some limitations for this type of application can be the computational speed, the communication speed and the length of the simulation (flight trajectory).

Chapter 4 : Contribution 2

In Chapter 4, it has been demostrated that ILC can be used as a tool for automatically creating (or tuning) linear feedback controllers for rejecting unknown periodic disturbances. This idea has been given under the name of *learning based controller tuning (LBCT)* workflow (see Workflow 2) which has shown that ILC and feedback can be united for achieving a satisfying performance with much less design effort. The main contribution of this chapter is twofold : (1) utilising ILC, the bridge between the requirement of having *a priori* knowledge of the periodic disturbance and building an internal model principle based feedback controller has been broken ; (2) utilising ILC, an automatisation is achieved for the tuning of linear feedback controller parameters.

To begin, for periodic systems a large amount of repetitive data is generally available and thus ILC can be used as a beneficial tool for reaching superior rejection performances that would not be possible to obtain by means of feedback laws only. The usage of ILC becomes even more motivating from the point of control design when the periodic system is under *a priori* unknown periodic nonlinear disturbances. This is simply because ILC allows the designer to find the right control input that will reject the periodic disturbance without needing to model the disturbance. Furthermore, the feedback methods are not able to improve the control performance by detecting repetitive patterns in the system which means that there is no anticipation and the transients cannot be removed. On the other hand, this is not the case for ILC which anticipates for transients using repetitive data. Therefore, the LBCT workflow has been proposed to show that one can achieve much better rejection performances by including ILC in the design procedure of a feedback controller and such process actually corresponds to the automatic tuning of feedback controller parameters. To mention again, the LBCT approach becomes particularly interesting when the periodic disturbance is completely unknown. Thanks to ILC, in such case, the feedback controller design can still be automatically obtained without the need of any modelling.

Workflow 2: Learning Based Controller Tuning (LCBT) (Recalled, Chapter 4)

Data: An internal closed-loop system $P \in \mathcal{H}_\infty$ (the open-loop plant plus a feedback controller) that is stable (see Figure 4.1) and subject to repetitive disturbance $d(t)$; a desired reference input $r(t)$; values for $\{\rho,\, \lambda\} \in \mathbb{R}^+$ and $M \in \mathbb{N}^+$ on Table 4.1.

Result: A linear controller rejecting a non-modelled repeating disturbance, $d(t)$.

1 ρ, λ and M can be chosen as suggested in [83];
2 Consider the switch is at '$S.2$' position in Figure 4.1;
3 Run ILC to find the system input that will attenuate the unwanted repeating frequencies;
4 Obtain the frequency data $\{\omega_j, \phi_j\}_{j=1}^{N_f}$ of the converged input signal, $u^*_{i=M}(t)$, from the last iteration;
5 Approximate a stable linear model $H(s) \in \mathcal{H}_\infty$ making a fit to this frequency data utilising [6] and [75];
6 **Design a controller based on $H(s)$ properties (internal model control)**.

The initial motivation of the work in Chapter 4 came after observing an equivalance

of ILC to the augmented state feedback control in rejecting a simple sinusoidal repetitive disturbance. According to the results, the ILC demonstrated a rather successful rejection performance that is superior to state feedback control. This is due to the fact that ILC was able to gradually remove the repeating errors and anticipate for the transients in the response without needing to know the disturbance. Hence, this brings out the following question : Why not make benefit of learning methods (like ILC) that are based on repetitive data manipulation when technology becomes less and less an issue in terms of computational speed, data storage etc. ? Trying to correctly model the repetitive disturbances can be quite a tedious process if the disturbances are unknown, partially known, nonlinear or lumped with other effects such as parametric uncertainties.

The missing part of this work is that the step 6 of the LBCT workflow above, i.e the transfer function $H(s)$, has not been demonstrated in the paper contribution. Therefore, it could be interesting to create $H(s)$ (which is an approximation of the ILC's performance in feedback form) and compare its rejection performance directly to ILC. This would allow to fully assess the efficiency of the LBCT procedure. Moreover, it could be interesting to further test this remaining part with a practical application on a real system.

Another improvement can be done by developing further the LBCT workflow in order to make it feasible to nonlinear systems under periodic disturbances. In fact, this issue has already been adressed in Chapter 5 and there is still an open work remaining from the tuning aspect (see Section 5.2). Apart from this it is always possible to check the performance with a different ILC method which may provide similar but faster results faster in terms of computational time. For example, the selected tuning for the norm optimal ILC (NO-ILC) in Chapter 4 may not be the best one and it is possible that other optimisation methods can give better results (NO-ILC is based on Lagrange multipliers technique which is based on an analytical minimisation process). In general, the focus of Chapter 4 was to demonstrate the functioning of a new workflow and the tuning was not the particular focus. Further research on this point can enhance the practicality of LBCT worklow.

Chapter 5 : Contribution 3 and 4 [10]

The Chapter 5 can be considered as the major contribution of the thesis study. The ultimate goal of the thesis has been achieved in this chapter and the previous contributions provided in Chapter 3 and Chapter 4 simply built the basis for the work presented here. There is especially a direct relation between Chapter 4 and 5 since the results obtained in Chapter 4 have been improved and completed further in Chapter 5 which led to a new framework called the *triple layer control approach* (see Figure 5.23). The significance of this approach comes from the fact that it unifies the learning-type control logic (layer 1) to the non-learning-type control logic (layer 3)(in other words the data-based logic to the conventional logic) via utilising an identification process (layer 2) in between its layers.

An interesting feature of the triple layer control is that it also motivates the trial of different method combinations which is now open for further research. In Chapter 5, the

10. Contribution 4 is an ongoing work.

Layers 1 and 3 of the triple layer controller are selected as ILC and output regulation via internal model control (IMC), respectively. This version of the triple layer controller was named *supervised output regulation via ILC (SOR-ILC)*. The rejection performance of SOR-ILC has been shown to function well for both the *linear* and *nonlinear* systems [11] under *nonlinear disturbances* [12]. This point is basically the most crucial outcome of the whole thesis work since finally the approach of designing a feedback law using iterative learning control has been fully developed.

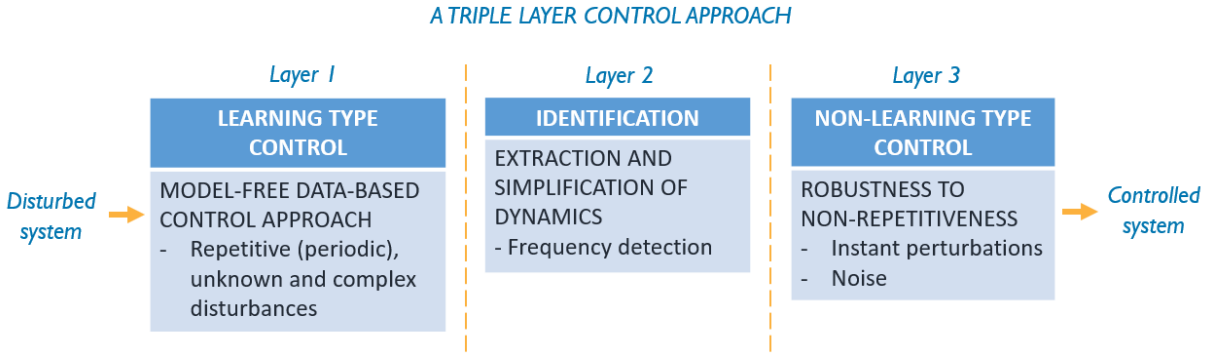

FIGURE 5.23: Triple-layer control approach (Recalled from Chapter 5)

As it was mentioned above, the given concept in Chapter 5 is not limited to ILC and output regulation as design choises on Layers 1 and 2, respectively. These choises have been done rather to motivate the designers to utilise the triple layer approach in order to benefit from the best features of both control family and decrease the disadvantages of both. For instance, the model-free nature of ILC helps to remove the necessity of having *a priori* knowledge of periodic disturbance frequencies while build a controller. On the other side, one can compensate for the major drawbacks of ILC (i.e. the lack of robustness to nonrepetitiveness in the data due to instant disturbances, noises and parametric uncertainty) by utilising a robust output regulator based on IMC (Figure 5.1 and Figure 5.3 can be revisited for more details). Here again, other methods can be considered by adding some optimisation criteria which would create an openning to the dynamic output feedback design instead of the full state feedback one.

It is worth to put emphasis on the fact that SOR-ILC approach is not only limited to simple linear systems and it can well be extended to nonlinear ones (as it was done in Section 5.2) without losing the relative simplicty of the overall control design compared to conventional non-learning-type methods. However, the precise tuning of SOR-ILC is still a process to be researched further.

In case where SOR-ILC is selected as the triple layer controller, the tuning of the output regulator parameters β and μs can be considered as a future study (these parameters can be used to put different weights on the contribution fo each disturbance frequency in the feedback). Such analysis would most probably improve the reduced performance occured while passing from Layer 1 results to Layer 3 results in Section 5.2 for the nonlinear system

11. Weak nonlinearities, see
12. In this thesis, the term nonlinear disturbance means a disturbance that is produced by a nonlinear system/oscillator

case (linear case has no issues regarding performance). Clearly, the output regulator tuning gets more complex if the plant is nonlinear (with a linear system in Section 5.1, a rough tuning of SOR-ILC raised even further the overall precision after ILC-only-performance). A particular interest can be given to this point as well in the future work in order to understand the main factors determining the SOR-ILC rejection in linear case and their relation to the increased complexity when switching from linear case to nonlinear case.

Moreover, it is useful to give some comments on the identification method used in the Layer 2 in Section 5.2.2 which is called *Hankel*. This method is particularly interesting since it directly uses the time-domain data (period of the signal) to approximately find a linear model fit (i.e to extract dominant frequencies). A powerful and simple identification method (Layer 2) has crucial importance since it constitutes a mid-layer between learning-type control (Layer 1) and non-learning type control (Layer 3). If the frequency extraction is not accurate, this has a direct effect on the Layer 3 since the internal model unit of the output regulator is built using the information acquired through the identification layer. Although the presented identification and model approximation results show that Hankel method works very well, there can still be some improvements to be considered as future work. Firstly, one can use faster and more memory efficient *singular value decomposition (SVD)* methods than the *Pencil method* in order to reduce the Hankel matrix $\mathcal{H}$ in (5.30) recalled below :

$$\mathcal{H} = \begin{bmatrix} u^*(t_1) & u^*(t_2) & \dots & u^*(t_{n+1}) \\ u^*(t_2) & u^*(t_3) & \dots & u^*(t_{n+2}) \\ \vdots & \vdots & \vdots & \vdots \\ u^*(t_n) & u^*(t_{n+1}) & \dots & u^*(t_{2n}) \end{bmatrix}$$

In other words, this means to use some other SVD method such as Zoom-SVD, Randomised-SVD or TallSkinny-SVD (see e.g. [55]) in the equations (5.33)-(5.34) recalled below :

$$\begin{aligned} \mathbf{U}_1\mathbf{S}_1\mathbf{V}_1^T &= \text{SVD}([\mathbf{E}^T\ \mathbf{A}^T]^T) \\ \mathbf{U}_2\mathbf{S}_2\mathbf{V}_2^T &= \text{SVD}([\mathbf{E}\ \mathbf{A}]) \end{aligned}$$

Secondly, one can make a study on how to optimise the number of harmonics to be used in Hankel (i.e. the dimension of the parameter r in (5.37) recalled below) such that the output regulator of SOR-ILC yields good performance with less identified frequency data.

$$\begin{aligned} \hat{u}^*(t_k) &= \sum_{i=1}^{r} \alpha_i e^{\sigma_i t_k} e^{\imath(\omega_i t_k + \theta_i)} \\ &= \sum_{i=1}^{r} \hat{h}(\alpha_i, \sigma_i, \omega_i, \theta_i) \end{aligned}$$

A criterion for determining the lowest number of harmonics in the model approximation is based on monitoring the orthogonality of the harmonics which is explained in Remark 5.1 by the following equation :

$$\angle(u, v) = \mathbf{arccos}\frac{\langle u, v\rangle}{||u||||v||} \tag{5.40}$$

where one can select the harmonics satisfying $||\angle(u, v) - \pi/2|| > \epsilon$ with ϵ being a small

computational error defined by the user. Clearly, this equation is a very useful practical tool ; however, it may also be possible to reduce the number of orthogonal harmonics even further by analysing which frequencies are the most useful ones for the output regulator performance. Hence, this point can be seen as a topic of future research for SOR-ILC or any other new triple layer approach.

Finally, it is always interesting to test different NO-ILC tunings or simply different ILC methods for Layer 1 of SOR-ILC. Since the goal of the thesis was to demonstrate new procedures (workflows) and frameworks, the tuning or theoretical improvement of the ILC methods was out of scope. However, having shown the proper functioning of the presented frameworks, one can carry out a specified reasearch on new ILC methods for SOR-ILC in order to achieve more satisfying (faster, less computational-heavy etc.) results with the triple layer framework. In addition to this, it remains still rather interesting to see whether this framework can make other method combinations possible and produce results that are as good as the ones of SOR-ILC. A good reference for further thinking on this topic can be done by reviewing Table 1.1 which provides a list of learning-type and non-learning-type control methods.

Annexe A

Extra Details

Basic ILC design procedure in frequency domain

Generally speaking, the design procedure of an ILC system depends on the ILC method used for the calculation of the filters. Yet, it can be said that a basic guiding procedure exists in essence. The procedures presented in [69] and [68] seem to be good references for understanding the basic ILC design steps in frequency-domain. The authors of [69] and [68] apply the same steps but their focuses are to design PD-type and fractional order ILCs, respectively. These steps can be given as follows :

1. Specify a desired cut-off frequency (w_c) for the ILC system such that its bandwidth becomes,
$$w \in [0, w_c].$$
2. Choose a low-pass filter $Q(e^{jw})$ such that
$$\begin{aligned} |Q(e^{jw})| &= 1, \quad \forall w \in [0, w_c], \\ |Q(e^{jw})| &= 0, \quad \forall w \in (w_c, \infty) \end{aligned} \tag{A.1}$$
A low-pass filter that will remove the unwanted high-frequencies as given in (A.1) can be designed through various different methods such as FIR, Butterworth, Chebyshev, Gaussian or any other filter that serves for the same purpose.
3. Specify a lower convergence margin (γ_{safe}) if a safer operation is desired. If not, let the margin be at least in $\gamma \in [0, 1)$. For the safer case scenario with a Q-filter that assures (A.1), the monotonic convergence condition in is written as,
$$0 \leq |1 - L(e^{jw})G(e^{jw})| \leq \gamma_{safe}, \quad \forall w \in [0, w_c]. \tag{A.2}$$
The purpose of introducing a γ_{safe} is to reduce the radius of the unit circle which is the area defined by MC condition. The aim here is to prevent the ILC system from exceeding the convergence rate limit of $\gamma = 1$ so that the convergence can be guaranteed.
4. Find a learning filter $L(e^{jw})$ such that (A.2) is satisfied at the boundaries. In different words, guarantee the following conditions :
$$\begin{aligned} |1 - L(e^{jw})G(e^{jw})| &= \gamma_{safe}, \ for\ w = w_c, \\ |1 - L(e^{jw})G(e^{jw})| &= 0, \qquad for\ w = 0. \end{aligned} \tag{A.3}$$
The learning filter $L(e^{jw})$ can be of various types depending on the ILC update method. For the systems using P-type, D-type, PD-type, PID-type ILCs or FO-ILC,

the design process of $L(e^{jw})$ is based on solving for the gains that will satisfy (A.3). Note that for the optimization-based ILC algorithms, the calculation of $L(e^{jw})$ and $Q(e^{jw})$ depends on the chosen cost function and the optimization method. Therefore, the steps are different than the ones described in this section. Yet, the filters can still be designed in a way that the conditions (A.1)-(A.3) are achieved.

References

[1] Hyo-Sung Ahn, Kevin L Moore et YangQuan Chen. *Iterative learning control : robustness and monotonic convergence for interval systems*. Springer Science & Business Media, 2007 (cf. p. 9, 10, 33).

[2] Notker Amann. « Optimal algorithms for iterative learning control. » Thèse de doct. University of Exeter, 1996 (cf. p. 36).

[3] Notker Amann, David H Owens et Eric Rogers. « 2D systems theory applied to learning control systems ». In : *Proceedings of 1994 33rd IEEE Conference on Decision and Control*. T. 2. IEEE. 1994, p. 985-986 (cf. p. 24).

[4] Notker Amann, David H Owens et Eric Rogers. « Iterative learning control for discrete-time systems with exponential rate of convergence ». In : *IEE Proceedings-Control Theory and Applications* 143.2 (1996), p. 217-224 (cf. p. 36).

[5] A Antoulas, S Lefteriu et A Ionita. « Model reduction and approximation theory and algorithms, chapter A tutorial introduction to the Loewner framework for model reduction ». In : *SIAM, Philadelphia. P. Benner, A. Cohen, M. Ohlberger and K. Willcox Eds* (2016) (cf. p. 96, 110).

[6] A.C. Antoulas, S. Lefteriu et A.C. Ionita. « Model reduction and approximation theory and algorithms ». In : SIAM, Philadelphia. P. Benner, A. Cohen, M. Ohlberger et K. Willcox Eds, 2016. Chap. A tutorial introduction to the Loewner framework for model reduction (cf. p. 81, 84, 86, 122).

[7] Athanasios C Antoulas et al. « A novel mathematical method for disclosing oscillations in gene transcription : A comparative study ». In : *PloS one* 13.9 (2018), e0198503 (cf. p. 109, 110).

[8] Athanasios Constantinos Antoulas, Christopher Andrew Beattie et Serkan Gǔğercin. *Interpolatory methods for model reduction*. SIAM, 2020 (cf. p. 110).

[9] Suguru Arimoto, Sadao Kawamura et Fumio Miyazaki. « Bettering operation of robots by learning ». In : *Journal of Robotic systems* 1.2 (1984), p. 123-140 (cf. p. 9).

[10] Daniele Astolfi. « Observers and robust output regulation for nonlinear systems ». Thèse de doct. 2016 (cf. p. 11, 94, 98, 99).

[11] Daniele Astolfi, Laurent Praly et Lorenzo Marconi. « Francis-Wonham nonlinear viewpoint in output regulation of minimum phase systems ». In : *11th IFAC Symposium on Nonlinear Control Systems*. IFAC. 2019 (cf. p. 98).

[12] Karl Johan Åström. « Theory and applications of adaptive control—a survey ». In : *Automatica* 19.5 (1983), p. 471-486 (cf. p. 4).

[13] Karl Johan Åström et Richard M Murray. *Feedback systems : an introduction for scientists and engineers*. Princeton university press, 2010 (cf. p. 75).

[14] Zeungnam Bien et Kyung M Huh. « Higher-order iterative learning control algorithm ». In : *IEE Proceedings D (Control Theory and Applications)*. T. 136. 3. IET. 1989, p. 105-112 (cf. p. 29).

[15] Zeungnam Bien et Jian-Xin Xu. *Iterative learning control : analysis, design, integration and applications*. Springer Science & Business Media, 2012 (cf. p. 9).

[16] Franco Blanchini et Mario Sznaier. « Persistent disturbance rejection via static-state feedback ». In : *IEEE Transactions on Automatic Control* 40.6 (1995), p. 1127-1131 (cf. p. 75).

[17] Lennart Blanken et Tom Oomen. « Multivariable iterative learning control design procedures : From decentralized to centralized, illustrated on an industrial printer ». In : *IEEE Transactions on Control Systems Technology* (2019) (cf. p. 13).

[18] Marc Bodson. « Rejection of periodic disturbances of unknown and time-varying frequency ». In : *International Journal of Adaptive Control and Signal Processing* 19.2-3 (2005), p. 67-88 (cf. p. 5).

[19] Marc Bodson et Scott C Douglas. « Adaptive algorithms for the rejection of sinusoidal disturbances with unknown frequency ». In : *Automatica* 33.12 (1997), p. 2213-2221 (cf. p. 5).

[20] Marc Bodson, Alexei Sacks et Pradeep Khosla. « Harmonic generation in adaptive feedforward cancellation schemes ». In : *IEEE Transactions on Automatic control* 39.9 (1994), p. 1939-1944 (cf. p. 5).

[21] D.A. Bristow, M. Tharayil et A.G. Alleyne. « A survey of iterative learning control : A learning-based method for high-performance tracking control ». In : *IEEE Control Systems Magazine* 26.3 (mar. 2006), p. 96-114 (cf. p. 28, 29).

[22] Douglas A Bristow, Marina Tharayil et Andrew G Alleyne. « A survey of iterative learning control ». In : *IEEE control systems magazine* 26.3 (2006), p. 96-114 (cf. p. 16, 18, 26).

[23] Jeffrey A Butterworth, Lucy Y Pao et Daniel Y Abramovitch. « The effect of nonminimum-phase zero locations on the performance of feedforward model-inverse control techniques in discrete-time systems ». In : *2008 American control conference*. IEEE. 2008, p. 2696-2702 (cf. p. 35).

[24] Christopher I Byrnes, Francesco Delli Priscoli et Alberto Isidori. « Output regulation of nonlinear systems ». In : *Output Regulation of Uncertain Nonlinear Systems*. Springer, 1997, p. 27-56 (cf. p. 94, 97, 98).

[25] W Jarrett Campbell et al. « A comparison of run-to-run control algorithms ». In : *Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)*. T. 3. IEEE. 2002, p. 2150-2155 (cf. p. 8).

[26] Giuseppe Casalino. « A learning procedure for the control of movements of robotic manipulators ». In : *IASTED Sympo. on Robotics and Automation, 1984* (1984) (cf. p. 9).

[27] Jeang-Lin Chang. « Robust output feedback disturbance rejection control by simultaneously estimating state and disturbance ». In : *Journal of Control Science and Engineering* 2011 (2011) (cf. p. 4).

[28] YangQuan Chen et Kevin L Moore. « An optimal design of PD-type iterative learning control with monotonic convergence ». In : *Proceedings of the IEEE Internatinal Symposium on Intelligent Control*. IEEE. 2002, p. 55-60 (cf. p. 36).

[29] Yangquan Chen et Changyun Wen. *Iterative learning control : convergence, robustness and applications*. Springer, 1999 (cf. p. 9, 33).

[30] Yangquan Chen et al. « Extracting projectile's aerodynamic drag coefficient curve via high-order iterative learning identification ». In : *Proceedings of 35th IEEE Conference on Decision and Control*. T. 3. IEEE. 1996, p. 3070-3071 (cf. p. 29).

[31] Tommy WS Chow et Yong Fang. « An iterative learning control method for continuous-time systems based on 2-D system theory ». In : *IEEE transactions on circuits and systems I : Fundamental theory and applications* 45.6 (1998), p. 683-689 (cf. p. 31).

[32] Roger C Conant et W Ross Ashby. « Every good regulator of a system must be a model of that system ». In : *International journal of systems science* 1.2 (1970), p. 89-97 (cf. p. 2).

[33] Ramon Costa Castelló et al. « An educational approach to the internal model principle for periodic signals ». In : *International journal of innovative computing information and control* 8.8 (2012), p. 5591-5606 (cf. p. 1, 5).

[34] John J Craig. « Adaptive control of manipulators through repeated trials ». In : *American Control Conference*. 21. 1984, p. 1566-1573 (cf. p. 9).

[35] Li Cuiyan, Zhang Dongchun et Zhuang Xianyi. « A survey of repetitive control ». In : *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*. T. 2. IEEE. 2004, p. 1160-1166 (cf. p. 6).

[36] Edward Davison. « The robust control of a servomechanism problem for linear time-invariant multivariable systems ». In : *IEEE transactions on Automatic Control* 21.1 (1976), p. 25-34 (cf. p. 97, 98).

[37] Enrique Del Castillo et Arnon M Hurwitz. « Run-to-run process control : Literature review and extensions ». In : *Journal of Quality Technology* 29.2 (1997), p. 184-196 (cf. p. 8).

[38] S. Dooren. « Iterative Learning Control for Internal Combustion Engines ». Mém. de mast. Institute for Dynamic Systems et Control - Swiss Federal Institute of Technology (ETH) Zurich, avr. 2015 (cf. p. 13).

[39] Guy A Dumont et Mihai Huzmezan. « Concepts, methods and techniques in adaptive control ». In : *Proceedings of the 2002 American control conference (IEEE Cat. No. CH37301)*. T. 2. IEEE. 2002, p. 1137-1150 (cf. p. 4).

[40] Marouane El Azzaoui et al. « Comparative study of the sliding mode and backstepping control in power control of a doubly fed induction generator ». In : *2016 International Symposium on Fundamentals of Electrical Engineering (ISFEE)*. IEEE. 2016, p. 1-5 (cf. p. 5).

[41] SV Emelyanov. « Variable structure control systems ». In : *Moscow, Nouka* (1967) (cf. p. 4).

[42] Benjamin T Fine, Sandipan Mishra et Masayoshi Tomizuka. « Model inverse based iterative learning control using finite impulse response approximations ». In : *2009 American Control Conference*. IEEE. 2009, p. 931-936 (cf. p. 35).

[43] Bruce A Francis et Walter Murray Wonham. « The internal model principle of control theory ». In : *Automatica* 12.5 (1976), p. 457-465 (cf. p. 2, 97, 98).

[44] Murray Garden. *Learning control of actuators in control systems*. US Patent 3,555,252. 1971 (cf. p. 9).

[45] Fei Gong et al. « Active Disturbance Rejection and Adaptive Backstepping Control for Induction Motor with Smooth Switching of Rotor Flux ». In : *Chinese Intelligent Automation Conference*. Springer. 2019, p. 296-304 (cf. p. 5).

[46] Svante Gunnarsson et Mikael Norrlöf. « On the design of ILC algorithms using optimization ». In : *Automatica* 37.12 (2001), p. 2011-2016 (cf. p. 36).

[47] Svante Gunnarsson et Mikael Norrlof. « Some aspects of an optimization approach to iterative learning control ». In : *Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No. 99CH36304)*. T. 2. IEEE. 1999, p. 1581-1586 (cf. p. 36).

[48] Jari Hätönen. « Issues of algebra and optimality in iterative learning control ». In : (2004) (cf. p. 31, 36, 37).

[49] Xiang He, Dejun Guo et Kam K Leang. « Repetitive control design and implementation for periodic motion tracking in aerial robots ». In : *2017 American Control Conference (ACC)*. IEEE. 2017, p. 5101-5108 (cf. p. 7).

[50] Peter Hippe et Joachim Deutscher. *Design of observer-based compensators : From the time to the frequency domain*. Springer Science & Business Media, 2009 (cf. p. 3, 5).

[51] Roberto Horowitz et James McCormick. « A self-tuning control scheme for disk file servos ». In : *IEEE transactions on magnetics* 27.6 (1991), p. 4490-4495 (cf. p. 4).

[52] Deqing Huang et al. « Adaptive Iterative Learning Control for High-Speed Train : A Multi-Agent Approach ». In : *IEEE Transactions on Systems, Man, and Cybernetics : Systems* (2019) (cf. p. 13).

[53] John Y Hung, Weibing Gao et James C Hung. « Variable structure control : A survey ». In : *IEEE transactions on industrial electronics* 40.1 (1993), p. 2-22 (cf. p. 4).

[54] TKSMT Inoue et al. « High accuracy control of a proton synchrotron magnet power supply ». In : *IFAC Proceedings Volumes* 14.2 (1981), p. 3137-3142 (cf. p. 6).

[55] Jun-Gi Jang et al. « Zoom-svd : Fast and memory efficient method for extracting key patterns in an arbitrary time range ». In : *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 2018, p. 1083-1092 (cf. p. 125).

[56] Honghai Ji, Zhongsheng Hou et Ruikun Zhang. « Adaptive iterative learning control for high-speed trains with unknown speed delays and input saturations ». In : *IEEE Transactions on Automation Science and Engineering* 13.1 (2015), p. 260-273 (cf. p. 13).

[57] Wang Jie et Meng Xiuyun. « Active disturbance rejection backstepping control for trajectory tracking of the unmanned airship ». In : *2017 IEEE International Conference on Unmanned Systems (ICUS)*. IEEE. 2017, p. 407-412 (cf. p. 5).

[58] Suparoek Junsupasen, Witthawas Pongyart et Nantakrit Yodpijit. « Adaptive Feedforward Cancellation (AFC) in Sound Fields : Computer-Based Approach for Active Noise Control (ANC) ». In : (cf. p. 5).

[59] Thomas Kailath. *Linear systems*. T. 156. Prentice-Hall Englewood Cliffs, NJ, 1980 (cf. p. 24).

[60] Dimitrios Karachalios, Ion Victor Gosea et Athanasios C Antoulas. « The loewner framework for system identification and reduction ». In : *Model Reduction Handbook : Volume I : System-and Data-Driven Methods and Algorithms*. De Gruyter, 2020 (cf. p. 110).

[61] Sadao Kawamura. « Iterative learning control for robotic systems ». In : *Proc. of IECON'84, Tokyo* (1984), p. 393-398 (cf. p. 9).

[62] Muhammad Kazim et al. « Robust backstepping control with disturbance rejection for a class of underactuated systems ». In : *IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society*. IEEE. 2017, p. 3104-3109 (cf. p. 5).

[63] Dehri Khadija, Ltaief Majda et Nouri Ahmed Said. « Rejection of sinusoidal disturbances with time varying frequency for discrete multivariable systems : Adaptive control with Q-parametrization ». In : *2013 9th Asian Control Conference (ASCC)*. IEEE. 2013, p. 1-6 (cf. p. 5).

[64] Kyong-Soo Kim et Qingze Zou. « Model-less inversion-based iterative control for output tracking : piezo actuator example ». In : *2008 American Control Conference*. IEEE. 2008, p. 2710-2715 (cf. p. 35).

[65] Burak Kürkçü, Coşku Kasnakoğlu et Mehmet Önder Efe. « Disturbance/uncertainty estimator based robust control of nonminimum phase systems ». In : *IEEE/ASME Transactions on Mechatronics* 23.4 (2018), p. 1941-1951 (cf. p. 1-4).

[66] Hak-Sung Lee et Zeungnam Bien. « Study on robustness of iterative learning control with non-zero initial error ». In : *International Journal of Control* 64.3 (1996), p. 345-359 (cf. p. 32).

[67] Guojun Li. « High-order iterative learning control for nonlinear systems ». In : *2017 6th Data Driven Control and Learning Systems (DDCLS)*. IEEE. 2017, p. 191-196 (cf. p. 29).

[68] Hongsheng Li et al. « A frequency-domain approach to PD-type iterative learning control ». In : *The 2010 IEEE International Conference on Information and Automation*. IEEE. 2010, p. 1652-1656 (cf. p. 33, 41, 43, 127).

[69] Hongsheng Li et al. « Design of fractional order iterative learning control on frequency domain ». In : *2011 IEEE International Conference on Mechatronics and Automation*. IEEE. 2011, p. 2056-2060 (cf. p. 34, 127).

[70] Lebao Li, Lingling Sun et Jie Jin. « Survey of advances in control algorithms of quadrotor unmanned aerial vehicle ». In : *2015 IEEE 16th International Conference on Communication Technology (ICCT)*. IEEE. 2015, p. 107-111 (cf. p. 13).

[71] Zhenxuan Li, Zhongsheng Hou et Chenkun Yin. « Iterative learning control for train trajectory tracking under speed constrains with iteration-varying parameter ». In : *Transactions of the Institute of Measurement and Control* 37.4 (2015), p. 485-493 (cf. p. 13).

[72] Jinchun Liu et al. « Adaptive feedforward cancellation strategy for unknown and time-varying disturbance with changing plant ». In : *2018 37th Chinese Control Conference (CCC)*. IEEE. 2018, p. 3437-3442 (cf. p. 5).

[73] Kai Liu et al. « A survey of run-to-run control for batch processes ». In : *ISA transactions* 83 (2018), p. 107-125 (cf. p. 8).

[74] Sergei Lupashin et al. « A simple learning strategy for high-speed quadrocopter multi-flips ». In : *2010 IEEE international conference on robotics and automation*. IEEE. 2010, p. 1642-1648 (cf. p. 58).

[75] A J. Mayo et A C. Antoulas. « A framework for the solution of the generalized realization problem ». In : *Linear Algebra and its Applications* 425.2 (2007), p. 634-662 (cf. p. 81, 84, 86, 122).

[76] Roel Merry, René van de Molengraft et Maarten Steinbuch. « Removing non-repetitive disturbances in iterative learning control by wavelet filtering ». In : *2006 American Control Conference*. IEEE. 2006, 6-pp (cf. p. 32).

[77] Q. Miao et al. « Iterative learning control method for improving the effectiveness of upper limb rehabilitation ». In : *23rd International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*. 16618808. Nanjing, China : IEEE, jan. 2016 (cf. p. 13).

[78] Kevin L Moore. *Iterative learning control for deterministic systems*. Springer Science & Business Media, 2012 (cf. p. 9, 33, 34).

[79] Kevin L Moore. « Multi-loop control approach to designing iterative learning controllers ». In : *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No. 98CH36171)*. T. 1. IEEE. 1998, p. 666-671 (cf. p. 24).

[80] Kevin L Moore, Mohammed Dahleh et SP Bhattacharyya. « Iterative learning control : A survey and new results ». In : *Journal of Robotic Systems* 9.5 (1992), p. 563-594 (cf. p. 9).

[81] T. Nagata et M. Tomizuka. « Robust Engine Torque Control by Iterative Learning Control ». In : *American Control Conference*. Hyatt Regency Riverfront, St. Louis, MO, USA, juin 2009, p. 2064-2069 (cf. p. 13).

[82] VO Nikiforov. « Adaptive non-linear tracking with complete compensation of unknown disturbances ». In : *European journal of control* 4.2 (1998), p. 132-139 (cf. p. 98).

[83] M. Norrlof. « Iterative Learning Control : Analysis, Design, and Experiments ». Thèse de doct. University of Linkoping, 2000 (cf. p. 81, 122).

[84] Mikael Norrlöf. « Iterative Learning Control–Analysis, Design, and Experiments ». In : (2000) (cf. p. 10, 36, 37, 40).

[85] Mikael Norrlöf et Svante Gunnarsson. « Time and frequency domain convergence properties in iterative learning control ». In : *International Journal of Control* 75.14 (2002), p. 1114-1126 (cf. p. 23, 24, 26).

[86] David H Owens. *Iterative learning control : an optimization paradigm*. Springer, 2015 (cf. p. 9).

[87] Jinwen Pan et Yong Wang. « Internal model based active disturbance rejection control ». In : *2016 American Control Conference (ACC)*. IEEE. 2016, p. 6989-6994 (cf. p. 1, 2, 5).

[88] Kwang-Hyun Park et Zeungnam Bien. « A study on the effect of variable initial state error in average operator-based iterative learning control ». In : *IFAC Proceedings Volumes* 41.2 (2008), p. 12448-12453 (cf. p. 32).

[89] R Parvathy et Asha Elizabeth Daniel. « A survey on active disturbance rejection control ». In : *2013 International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s)*. IEEE. 2013, p. 330-335 (cf. p. 5).

[90] Charles Poussot-Vassal. « Large-scale dynamical model approximation and its applications ». Thèse de doct. 2019 (cf. p. 110).

[91] E Rogers et DH Owens. « 2D systems theory and applications-a maturing area ». In : *IEE CONFERENCE PUBLICATION*. IET. 1994, p. 63-63 (cf. p. 24).

[92] Eric Rogers et David H Owens. *Stability analysis for linear repetitive processes*. T. 175. Springer, 1992 (cf. p. 24).

[93] Wilson J Rugh et Wilson J Rugh. *Linear system theory*. T. 2. prentice hall Upper Saddle River, NJ, 1996 (cf. p. 24).

[94] Emanuel Sachs et al. « Process control system for VLSI fabrication ». In : *IEEE Transactions on Semiconductor manufacturing* 4.2 (1991), p. 134-144 (cf. p. 7, 8).

[95] Angela P Schoellig, Fabian L Mueller et Raffaello D'Andrea. « Optimization-based iterative learning for precise quadrocopter trajectory tracking ». In : *Autonomous Robots* 33.1-2 (2012), p. 103-127 (cf. p. 13).

[96] Jin-Hua She, Xin Xin et Yasuhiro Ohyama. « Estimation of equivalent input disturbance improves vehicular steering control ». In : *IEEE Transactions on Vehicular Technology* 56.6 (2007), p. 3722-3731 (cf. p. 5).

[97] Jin-Hua She et al. « Improving disturbance-rejection performance based on an equivalent-input-disturbance approach ». In : *IEEE Transactions on Industrial Electronics* 55.1 (2008), p. 380-389 (cf. p. 5).

[98] Dong Shen et Xuefang Li. *Iterative Learning Control for Systems with Iteration-Varying Trial Lengths*. Springer, 2019 (cf. p. 32).

[99] Mark W Spong, Frank L Lewis et Chaouki T Abdallah. *Robot control : dynamics, motion planning, and analysis*. IEEE press, 1992 (cf. p. 23).

[100] Maarten Steinbuch et MJG Van De Molengraft. « Iterative learning control of industrial motion systems ». In : *1st IFAC Conference on Mechatronic Systems, Darmstadt, Germany*. 2000, p. 967-972 (cf. p. 94).

[101] Kuo-Tai Teng et Tsu-Chin Tsao. « A comparison of inversion based iterative learning control algorithms ». In : *2015 American Control Conference (ACC)*. IEEE. 2015, p. 3564-3569 (cf. p. 34, 35).

[102] Marina Tharayil et Andrew Alleyne. « A time-varying iterative learning control scheme ». In : *Proceedings of the 2004 American Control Conference*. T. 4. IEEE. 2004, p. 3782-3787 (cf. p. 32).

[103] Masaki Togai et Osamu Yamano. « Analysis and design of an optimal learning control scheme for industrial robots : A discrete system approach ». In : *1985 24th IEEE Conference on Decision and Control*. IEEE. 1985, p. 1399-1404 (cf. p. 36).

[104] Masayoshi Tomizuka. « Dealing with periodic disturbances in controls of mechanical systems ». In : *Annual Reviews in Control* 32.2 (2008), p. 193-199 (cf. p. 2).

[105] Masaru Uchiyama. « Formation of high-speed motion pattern of a mechanical arm by trial ». In : *Transactions of the Society of Instrument and Control Engineers* 14.6 (1978), p. 706-712 (cf. p. 9).

[106] Vadim Utkin. « Variable structure systems with sliding modes ». In : *IEEE Transactions on Automatic control* 22.2 (1977), p. 212-222 (cf. p. 4).

[107] Danwei Wang. « On D-type and P-type ILC designs and anticipatory approach ». In : *International Journal of Control* 73.10 (2000), p. 890-901 (cf. p. 33).

[108] Hongbin Wang, Jian Dong et Yueling Wang. « High-order feedback iterative learning control algorithm with forgetting factor ». In : *Mathematical Problems in Engineering* 2015 (2015) (cf. p. 29).

[109] Xuan Wang, Bing Chu et Eric Rogers. « Higher-order iterative learning control law design using linear repetitive process theory : Convergence and robustness ». In : *IFAC-PapersOnLine* 50.1 (2017), p. 3123-3128 (cf. p. 29).

[110] Youqing Wang, Furong Gao et Francis J Doyle III. « Survey on iterative learning control, repetitive control, and run-to-run control ». In : *Journal of Process Control* 19.10 (2009), p. 1589-1600 (cf. p. 6-8).

[111] Jan Willems. « Almost invariant subspaces : An approach to high gain feedback design–Part I : Almost controlled invariant subspaces ». In : *IEEE Transactions on Automatic Control* 26.1 (1981), p. 235-252 (cf. p. 4).

[112] Jan Willems. « Almost invariant subspaces : An approach to high gain feedback design–Part II : Almost conditionally invariant subspaces ». In : *IEEE Transactions on Automatic Control* 27.5 (1982), p. 1071-1085 (cf. p. 4).

[113] Shao Xingling et Wang Honglun. « Back-stepping active disturbance rejection control design for integrated missile guidance and control system via reduced-order ESO ». In : *ISA transactions* 57 (2015), p. 10-22 (cf. p. 5).

[114] Jian-Xin Xu. « A survey on iterative learning control for nonlinear systems ». In : *International Journal of Control* 84.7 (2011), p. 1275-1294 (cf. p. 32).

[115] Jian-Xin Xu, Sanjib K Panda et Tong Heng Lee. *Real-time iterative learning control : design and applications*. Springer Science & Business Media, 2008 (cf. p. 9).

[116] Jian-Xin Xu et Ying Tan. *Linear and nonlinear iterative learning control*. T. 291. Springer, 2003 (cf. p. 9).

[117] Jian-Xin Xu et Ying Tan. « On the P-type and Newton-type ILC schemes for dynamic systems with non-affine-in-input factors ». In : *Automatica* 38.7 (2002), p. 1237-1242 (cf. p. 18, 32).

[118] Manabu Yamada et al. « Adaptive repetitive control system for asymptotic rejection of periodic disturbances with unknown multiple periods ». In : *IFAC Proceedings Volumes* 37.12 (2004), p. 469-474 (cf. p. 6, 7).

[119] Han Woong Yoo et al. « Iterative Learning Control for Laser Scanning based Micro 3D Printing ». In : *IFAC-PapersOnLine* 52.15 (2019), p. 169-174 (cf. p. 13).

[120] K David Young, Vadim I Utkin et Umit Ozguner. « A control engineer's guide to sliding mode control ». In : *IEEE transactions on control systems technology* 7.3 (1999), p. 328-342 (cf. p. 4).

[121] Shu-Wen Yu. « Enhanced iterative learning control with applications to a wafer scanner system ». Thèse de doct. UC Berkeley, 2011 (cf. p. 75).

[122] Alessandro RL Zachi et al. « Robust disturbance rejection controller for systems with uncertain parameters ». In : *IET Control Theory & Applications* 13.13 (2019), p. 1995-2007 (cf. p. 1, 5).

[123] Ma Zhaowei et al. « An iterative learning controller for quadrotor UAV path following at a constant altitude ». In : *2015 34th Chinese Control Conference (CCC)*. IEEE. 2015, p. 4406-4411 (cf. p. 13).