JIAN-XIN XU and YING TAN

Linear and Nonlinear Iterative Learning Control

March 3, 2003

Springer
Berlin Heidelberg New York
Hong Kong London
Milan Paris Tokyo

To our parents and

Iris Hong Chen and Elizabeth Huifan Xu

– Jian-Xin Xu

Yang and my coming baby

– Ying Tan

Preface

Most existing control methods, whether conventional or advanced, robust or intelligent, linear or nonlinear, target at achieving the asymptotic convergence property in tracking a given trajectory. On the other hand, most practical control tasks, whether in process control or mechatronics, MEMS or space oriented, civil or military oriented, will have to be completed in a finite time interval. The scale of a finite interval can range from milliseconds to years. Tracking in finite horizon means the performance in transient process becomes more important. Often, perfect tracking performance is required from the very beginning. Obviously, asymptotic convergence along the time axis is inadequate, as it only guarantees the performance at the steady state when the time horizon goes to infinity. What is more, when the control task is repeated, the system will exhibit the same behavior. In practice there are many processes repeating the same task in a finite interval, ranging from a welding robot in a VLSI production line, to a batch reactor in pharmaceutical industry. Most existing control methods, devised in the time domain, are not able to fully capture and utilize the information available through the underlying nature of the system repeatability.

Iterative Learning Control (ILC) differs from most existing control methods in the sense that, it exploits every possibility to incorporate past control information: the past tracking error signals and in particular the past control input signals, into the construction of the present control action. This is realized through memory based learning. First the long term memory components are used to store past control information, then the stored control information is fused in a certain manner to form the feedforward part of the current control action. In certain sense, ILC complements the existing control methods.

Since the birth of iterative learning control in early 1980's, the history of ILC can be divided in to two phases. From early 1980s' to early 1990's was a linearly increasing period of ILC, in terms of reports and publications in theory and applications. From early 1990's, however, the research activities in ILC undergo a nonlinear (exponential) increase. One such evidence is, most premier control conferences have dedicated sessions related to iterative learning control, in addition to the increasing publications, special issues, and reports on the variety of applications. In order to update readers with the latest advances in this active area, this book provide a comprehensive coverage in most aspects of ILC, including linear and nonlinear ILC, lower order and higher order ILC, contraction mapping based and Lyapunov based ILC, output tracking ILC and state tracking ILC, model based and black-box based ILC design, robust optimal design of ILC, quantified ILC performance analysis, ILC for systems with global and local Lipschitz continuous nonlinearities, ILC for systems with parametric and non-parametric uncertainties, ILC with nonlinear optimality, etc.

The book can be used as a reference or textbook for a course at graduate level. It is also suitable for self-study, as most topics addressed in the book are self-contained in theoretical analysis, and accompanied by detailed examples to help readers, such as control engineers and graduate students, better capture the essence and the global picture of each ILC scheme. To further facilitate those who have interests but know little about ILC, two rudimentary sections are provided in Chapter 1 and Chapter 7 respectively. The first rudimentary section is written in such a way that it can be easily understood even by first year undergraduate students majoring in science and engineering. There are ten chapters in this monograph. Chapter 1 introduces the concept, rudiments and history of ILC. Chapters 2 - 6 reveal the intrinsic nature of contraction mapping based ILC. Chapters 7 - 9 extend the ILC to systems with more general nonlinearities. In Chapters 7 - 8 the energy function approaches, such as the Lyapunov technology, have been applied to repeated learning control problems. This serves as a bridge to connect the ILC field with the majority of nonlinear control fields, such as nonlinear optimality, adaptive control, robust control, etc. Also, in Chapter 9 the black-box approach using Wavelet network is integrated with ILC, which serves as another bridge to link the ILC field with the majority of intelligent control fields, such as neural network, fuzzy logics, etc. Finally, Chapter 10 concludes the book and points out several future research directions.

While preparing the book, the authors benefited greatly from stimulating discussions and judicious suggestions by ILC experts worldwide. Discussions with kevin Moore, Zeungnam Bien, Suguru Arimoto, Richard Longman, Zhihua Qu, David Owens, Yangquan Chen, Toshiharu Sugie, Danwei Wang, Tae-Yong Kuc, Chiang-Ju Chien, and many others, helped us clarify various aspects of the iterative learning control problems, which in turn motivated us to explore the underlying nature and properties of ILC, thereby lead to this book. The authors would like to express their special appreciation to the LNCIS series editor, Dr Thomas Ditzinger, for his strong support and professionalism.

Singapore,
Febuary, 2003

Jian-Xin Xu
Ying Tan

Contents

1

Introduction

According to Merrian-Webster's Collegiate Dictionary, the term *learning* is defined as

- the act or experience of one that learns
- knowledge or skill acquired by instruction or study
- modification of a behavioral tendency by experience (as exposure to conditioning)

In a word, learning generally implies a gaining or transfer of knowledge. In this book, the primary goal is centered on *iterative learning control*. The term "iterative" indicates a kind of action that requires the dynamic process be *repeatable*, i.e., the dynamic system is deterministic and the tracking control tasks are repeatable over a finite tracking interval. This kind of control problems is frequently encountered in many industrial processes, such as wafer manufacturing process, batch reactor process, IC welding process, and various assembly lines or production lines, etc. The motivation of iterative learning control comes from a deeper recognition, that knowledge can be learned from experience. In other words, when a control task is performed repeatedly, we gain extra information from a new source: past control input and tracking error profiles, which can be viewed as a kind of "experience". This kind of "experience" serves as a new source of knowledge related to the dynamic process model, and accordingly reduces the need for the process model knowledge. The new knowledge learned from the "experience" provides the possibility of improving the tracking control performance.

1.1 What is Iterative Learning Control

Let us start from a new class of control tasks: perfect tracking in a finite time interval under a repeatable control environment. The perfect tracking task implies that the target trajectory must be strictly followed from the very beginning of the execution. The repeatable control environment implies an identical target trajectory and the same initialization condition for all repeated control trials. Many existing control methods are not able to fulfill such a task, because they only warrant an *asymptotic* convergence, and being more essential, they are unable to *learn* from previous control trials, whether succeeded or failed. Without learning, a control system can only produce the same performance without improvement, even if the task repeats consecutively. ILC was proposed to best meet this kind of control tasks. The idea of ILC is straightforward: use the control information of the preceding trial to improve the control performance of the present trial. This is realized through memory based learning.

Fig. 1.1 shows one such schematic diagram,

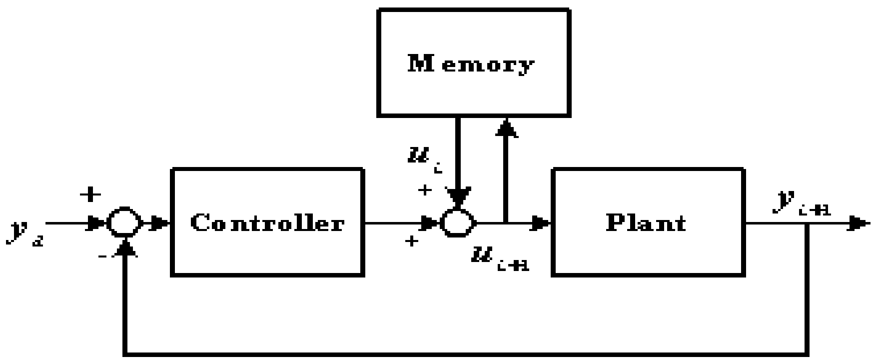

Fig. 1.1. Memory Based Learning

where the subscript i denotes the i-th control trial. Assume that the controller is memoryless. It can be seen, in addition to the standard feedback loop, a set of memory components are used to record the control signal of the preceding trial, $u_i(t)$, which is incorporated into the present control, $u_{i+1}(t)$, in a pointwise manner. The sole purpose is to embed an *internal model* into the feed-through loop. Let us see how this can be achieved. Assume that the target trajectory, $y_d(t)$, is repeated over a fixed time interval, and the plant is deterministic with exactly the same initialization condition. Suppose that the perfect output tracking is achieved at the i-th trial, i.e. $y_d(t)-y_i(t)=0$, where $y_i(t)$ is the system output at the i-th trial. The feedback loop is equivalently broken up. $u_i(t)$ who did the perfect job will be preserved in the memory for the next trial. In the sequel $u_{i+1}(t)=u_i(t)$, which warrants a perfect tracking with a pure feedforward.

A typical ILC, shown in Fig. 1.2, is somehow still different from Fig. 1.1.

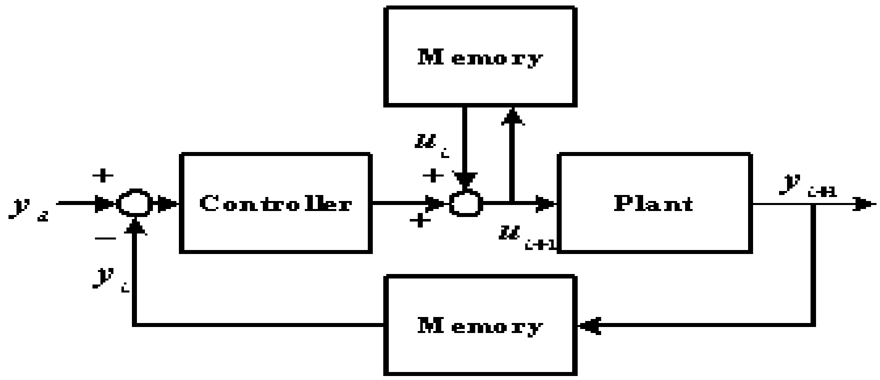

Fig. 1.2. A typical ILC

The interesting idea is to further remove the time domain feedback from the current control loop. Inappropriately closing the loop may lead to instability. To design an appropriate closed-loop controller, much of the process knowledge is required. Now suppose the process dynamics cannot escape to infinity during the tracking period that is always a finite interval in ILC tasks. We need not even be bothered to design a stabilizing controller in the time domain, as far as the control system converges gradually when the learning process repeats. In this way an ILC can be designed with the minimum system knowledge.

In the field of ILC, such repeated learning *trials* are frequently described by words like *cycles*, *runs*, *iterations*, *repetitions*, *passes*, etc. Since the majority of ILC related work done hitherto is under the framework of contraction mapping characterized by an iterative process, it is thus more appropriate to use the words *iteration(s)*, iteration axis, iteration domain, etc. to describe such an *iterative* learning process, as we shall follow in the rest of this book.

The concept of performance improvement under a repeated operation process have long been observed, analyzed and applied [93] [129]. The articles by [7, 38, 9, 12, 64, 65] have formed the initial framework of ILC, under which subsequent developments have taken place over the years. Since then, though undergoing rapid progress, the main framework of ILC has been determined and the majority of ILC schemes developed hitherto are still within this framework, which is characterized by two key features: the linear pointwise updating law and iterative convergence, and is subject to two fundamental conditions: the global Lipschitz continuity (GLC) condition and the identical initialization condition (*i.i.c*). The contraction mapping methodology, a method commonly used in function approximation and numerical analysis, has accordingly been brought up in iterative learning control design. In order to capture the concept of ILC from a more quantified point of view, in the rest of this section we shall conduct a rudimentary course briefing on ILC.

1.1.1 The Simplest ILC: an example

To illustrate the underlying concepts and properties of ILC, let us start with the simplest ILC problem: for a given process

$$y(t) = g(t)u(t)$$

where $g(t) \neq 0$ is defined over a period $[0,\ T]$, find the control input, $u(t)$, such that the target trajectory

$$y_d(t) \quad \forall t \in [0,\ T]$$

can be perfectly tracked. Without loss of generality we assume that $y_d(t)$ and $g(t)$ are bounded functions.

If $g(t)$ is known *a priori*, this problem becomes a little trivial, as we can simply calculate the desired control signal directly by inverting the process, which is exactly an open-loop approach

$$u_d(t) = \frac{y_d(t)}{g(t)} \quad \forall t \in [0,\ T].$$

However, we know that any open-loop control schemes are sensitive to the plant modeling inaccuracy. In our case, if the exact values of $g(t)$ are not available, the above simple open-loop scheme does not work. Let us assume that $g(t)$, though unknown, spans within $0 < \alpha_1 \leq g(t) \leq \alpha_2 < \infty$, where α_1 and α_2 are known lower and upper bounds. Can we find the desired control profile $u_d(t)$? One may think of a two-stage approach. From the input-output relationship and the availability of the measurement of $y(t)$ and $u(t)$, first identify the time-varying gain $g(t)$ point-wisely for $t \in [0,\ T]$. Then the desired control signal can be computed according to the inverse relationship, $u_d(t) = y_d(t)/g(t)\ \forall t \in [0,\ T]$, provided that the function $g(t)$ is captured perfectly over $[0,\ T]$.

Can we merge the two stage control into one stage, that is, can we *directly* acquire the desired control signal without any parametric or function identification? This will make the control system more efficient, and avoid extra error incurred by any intermediate computation, e.g. a large numerical error may occur if $g(t)$ takes a very small value at some instant t and is inverted. If the control task runs once only and ends, we are not able to directly achieve the desired control signal. When the same control task is repeated many times, we can acquire the control signal *iteratively* by the following simplest iterative learning control scheme

$$u_{i+1}(t) = u_i(t) + q\Delta y_i(t) \quad \forall t \in [0,\ T] \tag{1.1}$$

where the subscript $i \in \mathcal{Z}_+$ is the iteration index, $\mathcal{Z}_+ = 0, 1, \cdots$ is the set of non-negative integers. $u_0(t)$ can be either generated by any control method or simply set to be zero. In fact all we need for $u_0(t)$ is to guarantee a bounded output $y_0(t)$. q is a constant learning gain, and $\Delta y_i(t) \triangleq y_d(t) - y_i(t)$ is the output tracking error sequence. Let us see how does this scheme work *iteratively*, and investigate conditions which ensure the system *learnability*.

First of all, we are going to execute the control action many times as i evolves, with the ultimate objective of finding out the desired control signal, $u_d(t)$, with respect to $y_d(t)$. Once $y_d(t)$ is given, $u_d(t)$ should be fixed. This implies that the system, in this particular case the function $g(t)$, defined over $[0,\ T]$, must be identical for any iterations. We know that a deterministic system will produce the same response when the same input repeats. We thus define a *repeatable control environment*: a deterministic system with the control task repeated over a fixed time interval. The repeatability is the very first necessary condition for any deterministic learning controller to effectively perform.

Under the repeatable control environment, can the simplest ILC warrants a convergence sequence of $y_i(t)$ to $y_d(t)$, or $u_i(t)$ to $u_d(t)$, as $i \to \infty$? There are two ways we can prove the convergence, either $\Delta y_i(t) \to 0$, or $\Delta u_i(t) \triangleq u_d(i) - u_i(t) \to 0$, when $i \to \infty$. For simplicity we will omit the time t for all variables from 0 to T if not otherwise mentioned. First we demonstrate the convergence of the output tracking sequence

$$\begin{aligned}\Delta y_{i+1} &= y_d - y_{i+1} = y_d - g u_{i+1} = y_d - g(u_i + q\Delta y_i)\\ &= (y_d - g u_i) - qg\Delta y_i = (1 - qg)\Delta y_i.\end{aligned}$$

Consequently

$$|\Delta y_{i+1}| \le |1 - qg| \cdot |\Delta y_i|.$$

On the other hand, we know $0 < \alpha_1 \le g(t) \le \alpha_2 < \infty$, hence a conservative selection of the learning gain is

$$q = \alpha_2^{-1}.$$

It is easy to verify

$$0 \le |1 - qg| \le \frac{\alpha_2 - \alpha_1}{\alpha_2} = \gamma < 1$$

and

$$\frac{|\Delta y_{i+1}|}{|\Delta y_i|} \le \gamma < 1, \quad \forall i \in \mathcal{Z}_+,$$

which shows

$$\lim_{i\to\infty} |\Delta y_i| \le \lim_{i\to\infty} \gamma^{i+1} |\Delta y_0| \to 0$$

because $y_0(t)$ and $y_d(t)$ are finite in $[0,\ T]$.

Now let us show, as an alternate way, that $\Delta u_i \to 0$. Note

$$\begin{aligned}\Delta u_{i+1} &= u_d - u_{i+1} = u_d - (u_i + q\Delta y_i)\\ &= (u_d - u_i) - q\Delta y_i = \Delta u_i - q\Delta y_i.\end{aligned}$$

On the other hand,

$$\Delta y_i = y_d - y_i = gu_d - gu_i = g\Delta u_i.$$

By substituting Δy_i we have

$$\Delta u_{i+1} = (1 - qg)\Delta u_i,$$

from which we can see that the convergence conditions are the same for the sequences Δy_i and Δu_i. In this particular problem, the convergence of u_i to u_d implies the convergence of y_i to y_d. For more complicated problems related to the dynamic process or MIMO cases, they may show some differences.

Following the above demonstration on the simplest ILC, a question may arise: does the simplest ILC still work if the process is nonlinear in control input u (non-affine-in-input)? In the following we will address this problem.

1.1.2 ILC for Non-affine Process

A non-affine-in-input process can be described by

$$y(t) = g(u(t), t) \quad t \in [0, T]$$

where $g(u,t)$ is nonlinear in u, e.g. $g = ue^u$. It is worth to point out that, even if g is known *a priori*, the closed form of g^{-1} may not exist for most nonlinear functions. Thus we are not able to find the desired control profile by inverting the process, consequently $u_d = g^{-1}(y_d(t), t)$ is not achievable. Moreover, in practice g could be only partially known. To capture the desired control signal, we need to look for a more powerful approach, which is again the simplest ILC (1.1) associated with certain condition imposed on the function g.

Let us first derive the convergence of the input sequence, $u_i \to u_d$. Assume that g is continuously differentiable to all the arguments, using the Mean Value Theorem

$$\begin{aligned}\Delta y_i &= y_d - y_i = g(u_d, t) - g(u_i, t) = g(u_d, t) - g(u_d - \Delta u_i, t)\\ &= g(u_d, t) - [g(u_d, t) - g_u(\xi_i, t)\Delta u_i] = g_u(\xi_i, t)\Delta u_i\end{aligned}$$

where $g_u \triangleq \frac{\partial f}{\partial u}$, and $\xi_i \in [u_d - |\Delta u_i|, u_d + |\Delta u_i|]$. Following the preceding derivation, and substituting the above relationship

$$\begin{aligned}\Delta u_{i+1} &= \Delta u_i - q\Delta y_i = \Delta u_i - qg_u(\xi_i, t)\Delta u_i\\ &= (1 - qg_u(\xi_i, t))\Delta u_i,\end{aligned}$$

hence

$$|\Delta u_{i+1}| \leq |1 - qg_u(\xi_i, t)||\Delta u_i|.$$

In order to let $|1 - qg_u(\xi_i, t)| = \gamma < 1$, the function g needs to meet the following condition:

$(C1)$ g_u must have known lower and upper bounds, both are of the same sign and strictly nonzero. Assume α_1 the lower bound and α_2 the upper bound, then either $0 < \alpha_1 \leq \alpha_2$ or $0 > \alpha_2 \geq \alpha_1$.

With this condition a learning gain q can be chosen to make γ strictly less than one. A conservative design is $q = \alpha_2^{-1}$ (for simplicity we only consider positive g_u). Note the similarity between the present non-affine and preceding linear cases. g_u is the equivalent process *gain*, like $g(t)$ in the linear case, thus it naturally leads to the same convergence condition within the same bounding condition. However, in the non-affine case the process gain g_u is depending on the control input u. Thus it is also necessary to limit u, especially when g_u turns out to be a radially unbounded function of u, i.e.

$$\lim_{|u| \to \infty} |g_u| \to \infty.$$

In such circumstance we have to limit u to a compact set $\mathcal{U}$. By virtue of the continuous differentiability of g, g_u is bounded on $\mathcal{U}$. For example consider $g = ue^u$ and $u \in [0, u_m]$, then $g_u = e^u + ue^u$, $\alpha_1 = 1$, and $\alpha_2 = e^{u_m} + u_m e^{u_m} < \infty$, as long as u_m is finite. Note that it is essential that, $g \neq 0$ for the linear cases, or $g_u \neq 0$ for the non-affine cases, because the singularity yields a zero process gain, hence the system is uncontrollable at the singular points.

Like the linear case, we can derive the same convergence property of the output sequence, $y_i \to y_d$, as below

$$\begin{aligned}\Delta y_{i+1} &= y_d - y_{i+1} = g(u_d, t) - g(u_{i+1}, t) = g(u_d, t) - g(u_i + q\Delta y_i, t) \\ &= g(u_d, t) - g(u_i, t) - g_u(\xi_i, t)q\Delta y_i = (1 - qg_u(\xi_i, t))\Delta y_i.\end{aligned}$$

The convergence condition is the same: $|1 - qg_u(\xi_i, t)| = \gamma < 1$.

1.1.3 ILC for Dynamic Process

Up to now the process discussed is static or memoryless. Now let us consider the following SISO linear dynamics

$$\begin{aligned}\dot{x} &= ax + bu \quad x(0) = x_0 \\ y &= cx + du\end{aligned} \tag{1.2}$$

where a, b, c and d are unknown system parameters, either time-invariant or time varying. Let us first focus on the time-invariant circumstance. If still using the simplest ILC, can we achieve either the output convergence, $\Delta y_i \to 0$, or input convergence $\Delta u_i \to 0$, as $i \to \infty$?

It is quite surprising, that we can still use the simplest ILC scheme (1.1). However, the learning convergence analysis becomes more complicated, and a very specific condition, identical initialization condition (*i.i.c.*), as stated below, is indispensable.

$(C2)$ $x_i(0) = x_0$

i.e., the dynamic process will always start from the same position regardless of iterations. This condition in general holds if the process is deterministic and strictly repeatable. We assume that the bound of the process gain, characterized by the direct transmission term d, is confined to a known interval $0 < \alpha_1 \le d \le \alpha_2 < \infty$.

Let us first derive the output tracking convergence

$$\begin{aligned}\Delta y_{i+1} &= y_d - y_{i+1} = y_d - cx_{i+1} - du_{i+1} \\ &= y_d - cx_{i+1} - d(u_i + q\Delta y_i) \\ &= y_d - (cx_i + du_i) - qd\Delta y_i - c(x_{i+1} - x_i) \\ &= (1 - qd)\Delta y_i - c(x_{i+1} - x_i).\end{aligned} \tag{1.3}$$

There is an extra term, $c(x_{i+1} - x_i)$, compared with the memoryless case. If $c = 0 \ \forall t \in [0,\ T]$, we have the same convergence property, because the process dynamics in this case is decoupled from the output. When $c \neq 0$, it is obvious that the system dynamics will have certain effect on the output tracking performance. To evaluate the dynamic effect, define $\Delta_i x(t) = x_{i+1}(t) - x_i(t)$,

$$\Delta_i x(t) = \Delta_i x_i(0) + \int_0^t e^{a(t-\tau)} b[u_{i+1}(\tau) - u_i(\tau)]d\tau.$$

Using the identical initialization condition (*i.i.c.*), and the learning law $u_{i+1} - u_i = q\Delta y_i$, we have

$$\Delta_i x(t) = \int_0^t e^{a(t-\tau)} qb\Delta y_i(\tau)d\tau.$$

In the memoryless case, a *monotonic* convergence is guaranteed at each instant t, because the relation $|\Delta y_{i+1}(t)| \le \gamma|\Delta y_i(t)|$ holds for every $t \in [0,\ T]$. In the presence of the system dynamics, we cannot expect such an ideal convergence property. Instead we look for *uniform* convergence over a subinterval in $[0,\ T]$, that is, the maximum tracking error of the $(i+1)$-th iteration is less than that of i-th. Mathematically define a norm for $\mathcal{C}[0,\ T]$ functions $f(t)$

$$|f|_s = \max_{t \in [0,\ T]} |f(t)|.$$

Then the maximum tracking error of the i-th iteration is $|\Delta y_i|_s$. Now the upper bound of $\Delta_i x$ can be evaluated

$$\begin{aligned}|\Delta_i x|_s &\le \int_0^t e^{a(t-\tau)}|qb|\ |\Delta y_i(\tau)|_s d\tau\\ &\le |qb|\ |\Delta y_i|_s \int_0^T e^{|a|(T-\tau)}d\tau\\ &\le w_i|\Delta y_i|_s \end{aligned}\tag{1.4}$$

where

$$w_i \triangleq \begin{cases} |qb|\dfrac{e^{|a|T}-1}{|a|} & a \neq 0\\ |qb|T & a = 0.\end{cases}$$

In the sequel, taking the norm $|\cdot|_s$ in (1.3) and using (1.4), yields

$$|\Delta y_{i+1}|_s \le |1-qd|_s|\Delta y_i|_s + |c|w_i|\Delta y_i|_s. \tag{1.5}$$

Since we can always choose q such that $|1-qd|_s \le \gamma < 1$, there exists a positive constant δ satisfying $\gamma+\delta<1$. It is easy to verify that if the tracking interval $[0,T_1] \subset [0,T]$, where T_1 is given as

$$T_1 \le \begin{cases} \dfrac{1}{|a|}\ln\left[1+\dfrac{\delta|a|}{|qcb|}\right] & a \neq 0\\ \dfrac{\delta}{|qcb|} & a = 0\end{cases}\tag{1.6}$$

then

$$|\Delta y_{i+1}|_s \le (\gamma+\delta)|\Delta y_i|_s.$$

As γ and δ are iteration independent, $\gamma+\delta<1$ ensures a geometric convergence of the output tracking error sequence.

From the above derivation procedure, it can be seen that the learning convergence property in a dynamic process is quite different from a static process:

1. ILC in a static process achieves monotonic convergence at each time t, whereas in a dynamics process only ensures monotonic convergence in terms of the maximum error over a limited time interval. This is due to the dynamic influence from $\Delta_i x$ to Δy_i;
2. In the dynamic process, an extra *i.i.c.*, $\Delta_i x(0) = 0$, is required. This is again due to the dynamic influence from $\Delta_i x$ to Δy_i. The solution trajectory of a differential equation is determined by its initial value and exogenous input. If the initial value $x_i(0)$ varies at every iteration, it is no longer a strictly *repeatable* control environment;

3. The time interval of the target trajectory, $[0,\ T]$, is limited by (1.6). In the static learning process we do not have such a limit. Note that the dynamic influence, quantified by (1.4) in the worst case, enters the convergence condition (1.5) as a disturbance and increases as T increases. It can also be seen that the absolute value of the system parameter a is taken into consideration. Equivalently the dynamics is treated as a divergent one, even though the original a may take a negative value (stable). In order to prevent the assumed divergent term $\Delta_i x(t)$ from growing too larger and dominating the learning process, T_1 must be sufficiently small. This limitation on T can be overcome by introducing a time-weighted norm (to be addressed in next chapter).

Now let us derive the convergence conditions for the input sequence. It requires a different identical initialization condition

$(C2')$ $x_i(0) = x_d(0)$,

i.e., the dynamic process will have to start from the same initial position as the state trajectory $x_d(t)$ generated by $u_d(t)$. This condition is obviously more specific than $(C2)$. On the other hand, it provides certain convenience when deriving the higher order ILC schemes, as can be seen in subsequent chapters.

Assume there exists a control input u_d, which generates x_d and y_d, namely

$$\begin{aligned} \dot{x}_d &= a x_d + b u_d \quad x_d(0) = x_0 \\ y_d &= c x_d + d u_d. \end{aligned}$$

Our objective is to show that the simplest ILC scheme (1.1) can converge to the desired u_d. First we can derive the relation

$$\begin{aligned} \Delta u_{i+1} &= u_d - u_{i+1} = u_d - u_i - q\Delta y_i \\ &= \Delta u_i - qc\Delta x_i - qd\Delta u_i \\ &= (1 - qd)\Delta u_i - qc\Delta x_i \end{aligned} \tag{1.7}$$

where $\Delta x_i \overset{\triangle}{=} x_d - x_i$. The state error dynamics is

$$\begin{aligned} \Delta \dot{x}_i &= \dot{x}_d - \dot{x}_i = a(x_d - x_i) + b(u_d - u_i) \\ &= a\Delta x_i + b\Delta u_i. \end{aligned}$$

Integrating both sides of the state error dynamics and using the *i.i.c.* $(C2')$ yields

$$\begin{aligned} \Delta x_i(t) &= \Delta x_i(0) + \int_0^t e^{a(t-\tau)} b\Delta u_i(\tau) d\tau \\ &= \int_0^t e^{a(t-\tau)} b\Delta u_i(\tau) d\tau. \end{aligned}$$

The upper bound of the tracking error $\Delta x_i(t)$, analogous to (1.4), is

$$|\Delta x_i|_s \leq \tilde{w}_i |\Delta u_i|_s.$$

where

$$\tilde{w}_i \triangleq \begin{cases} |b|\dfrac{e^{|a|T}-1}{|a|} & a \neq 0 \\ |b|T & a = 0. \end{cases}$$

Consequently from (1.7)

$$|\Delta u_{i+1}|_s \leq |1 - qd|_s |\Delta u_i|_s + |c|\tilde{w}_i |\Delta u_i|_s. \tag{1.8}$$

Note the analogy between the above formula and (1.5), the differences lie in the replacement of Δy_i and w_i by Δu_i and $\tilde{w}_i$ respectively. Thus the same convergence property can be derived: if the interval T is less than the limit (1.6), $u_i(t)$ converges uniformly to $u_d(t)$.

Though only a linear time-invariant (LTI) system is considered, the above results can be easily extended to linear time-varying (LTV) systems with time-varying parameters $a(t)$, $b(t)$, $c(t)$ and $d(t)$. Assume $d(t) \in [\alpha_1, \alpha_2]$ as before, and $|a|_s$, $|b|_s$, $|c|_s$ are finite, the only difference is the convergence interval

$$T_1 \leq \begin{cases} \dfrac{1}{|a|_s} \ln \left[1 + \dfrac{\delta |a|_s}{|c|_s |b|_s |q|} \right] & a \neq 0 \\ \dfrac{\delta}{|c|_s |b|_s |q|} & a = 0. \end{cases} \tag{1.9}$$

1.1.4 D-Type ILC for Dynamic Process

Often the following LTI dynamics is encountered in practice

$$\begin{aligned} \dot{x} &= ax + bu \quad x(0) = x_0 \\ y &= cx \end{aligned}$$

where the direct transmission term du is absent. From the system theory, we known that $d \neq 0$ indicates a relative degree of zero, if $d = 0$ but $cb \neq 0$, the relative degree is one. Can we still apply the simplest ILC scheme (1.1) for systems without the direct transmission term d? From the preceding derivation (1.3), associated with the expression of $\Delta_i x(t)$, and setting $d = 0$, we have

$$\Delta y_{i+1}(t) = \Delta y_i(t) - \int_0^t e^{a(t-\tau)} qb \Delta y_i(\tau) d\tau.$$

The integral term may be either positive or negative, depending on the entire history of $\Delta y_i(\tau)$ $\forall \tau \in [0, t]$. As a consequence, there is no guarantee that the absolute value of the right hand side can be made less than that of the left hand side, no matter how to select the learning gain, q.

When du is missing, $y(t)$ cannot be manipulated directly by $u(t)$. The only channel is via the state $x(t)$, which is however the convolution of u over $[0,\ t]$. From the system point of view, a direct transmission from input to output, or a relative degree of zero, is needed for the simplest ILC to work. In the above dynamic process, what is proportional to u is not y but $\dot{y} = c\dot{x} = cax + cbu$, namely the relative degree is one instead of zero. The relative degree is a geometric property of a process, little can be changed by incorporating any static feedback structure. Since $\dot{y}$ is proportional to u, this fact motivated a new ILC scheme – the D-type ILC

$$u_{i+1} = u_i + q\Delta\dot{y}_i. \tag{1.10}$$

It is called D type because a derivative term is used. In contrast to this, the preceding simplest ILC scheme is called P-type ILC, that applies to systems with zero relative degree.

With the D-type ILC, we have the following relationship

$$\begin{aligned}
\Delta\dot{y}_{i+1} &= \dot{y}_d - c\dot{x}_{i+1} = \dot{y}_d - cax_{i+1} - cbu_{i+1}\\
&= \dot{y}_d - cax_{i+1} - cb(u_i + q\Delta\dot{y}_i)\\
&= \dot{y}_d - (cax_i + cbu_i) - qcb\Delta\dot{y}_i - ca(x_{i+1} - x_i)\\
&= (1 - qcb)\Delta\dot{y}_i - ca\Delta_i x.
\end{aligned}$$

The upper bound of $ca\Delta_i x$, according to the ILC law (1.10) and *i.i.c.* $(C2)$, is

$$\begin{aligned}
|\Delta_i x|_s &\le \int_0^t e^{a(t-\tau)}|qb|\ |\Delta\dot{y}_i(\tau)|_s d\tau\\
&\le \int_0^T e^{|a|(T-\tau)}d\tau|qb|\ |\Delta\dot{y}_i|_s\\
&\le w_i|\Delta\dot{y}_i|_s
\end{aligned}$$

where

$$w_i \triangleq \begin{cases} |qb|\dfrac{e^{|a|T} - 1}{|a|} & a \ne 0\\ |qb|T & a = 0. \end{cases}$$

The maximum absolute derivative errors between two consecutive iterations are correlated by

$$|\Delta\dot{y}_{i+1}|_s \le |1 - qcb|_s|\Delta\dot{y}_i|_s - |ca|w_i|\Delta\dot{y}_i|_s.$$

Clearly here cb plays the same role as the direct transmission term d, in the sequel it requires $0 < \alpha_1 \le |cb| \le \alpha_2 < \infty$. There also exists a T_1, depending on the system parameters a, b and c, such that in $[0,\ T_1]$ the sequence $|\Delta\dot{y}_i|_s$ converges monotonically as $i \to \infty$.

A concern towards the D-type ILC is the demand on the derivative signals. In control theory and control practice, the availability of derivative signals of the system states is out of the question. However, this problem does not arise in the D-type ILC, simply because the derivative signals used for the present iteration are from the previous iteration. Therefore, one may use various kinds of filters, such as non-causal or zero-phase filters, to generate the derivative signals from the past output measurement.

So far we only prove the convergence property of the error derivative. In order to get the error convergence, we need one more identical initialization condition.

$(C2'')$ $\Delta y_i(0) = y_d(0) - y_i(0) = 0$,

indeed, from $\Delta \dot{y}_i(t) = 0$ and $\Delta y_i(0) = 0$ as $i \to \infty$, we have $\Delta y_i(t) = \Delta y_i(0) = 0$ $\forall t \in [0,\ T]$.

Likewise we can derive the convergence property of Δu_i associated with the *i.i.c.* $(C2')$ and $(C2'')$. The results can also be extended to LTV systems.

1.1.5 Can We Relax the Identical Initialization Condition?

As shown in the previous analysis, the identical initialization conditions $(C2)$, $(C2')$ and $(C2'')$ play the key role in ILC. The identical initialization condition, together with the global Lipschitz continuity condition, are two fundamental assumptions of ILC. A frequently raised question is, can *i.i.c.* be relaxed or removed? The *i.i.c.* have been criticized by control experts from different disciplines. Indeed, there are real systems well equipped with almost perfect homing mechanism such as high precision XY-table, robotic manipulators on assembly line, etc. However, for many practical control problems, perfect initialization is hardly achievable.

This issue has been exploited by many researchers [71, 103, 104, 53, 72, 100, 132], etc. The results confirm that *i.i.c.* are really imperative, and any small discrepancy in the initial value may lead to a divergent sequence. The reason is simple: the ILC algorithms, such as (1.1) and (1.10), are integrators along the iteration axis, which will accumulate any biased signals. Let us look at the simplest ILC law (1.1), and assume a small initial discrepancy in one of the *i.i.c.*, $(C2'')$,

$$\Delta y_i(0) = y_d(0) - y_i(0) = \epsilon, \qquad \forall i \in \mathcal{Z}_+$$

where $0 < \epsilon \ll 1$ is a constant. Due to the integral nature of ILC in iteration domain, at $t = 0$

$$\begin{aligned} u_{i+1}(0) &= u_i(0) + q\Delta y_i(0) \\ &= u_{i-1}(0) + q\Delta y_{i-1}(0) + q\Delta y_i(0) \\ &= \cdots = u_0(0) + q\sum_{j=0}^{i} \Delta y_j(0) \end{aligned}$$

$$= u_0(0) + (i+1)q\epsilon \tag{1.11}$$

which goes to infinity as the iteration number $i \to \infty$. In this example, we assume the initial error sequence $|\Delta y_i(0)| \in l^\infty$. The problem can be mitigated to certain extent, if the initial error sequence is in l^1. Further, if $|\Delta y_i(0)|$ is a discrete-time zero mean white noise, then the mean value of $u_i(0)$ will converge to zero.

To avoid the worst case of divergence, a well adopted remedy is to add a forgetting factor to the past control signal, and the simplest ILC (1.1), for instance, becomes

$$u_{i+1}(t) = pu_i(t) + q\Delta y_i(t) \quad \forall t \in [0,\ T] \tag{1.12}$$

where $0 < p \leq 1$ is the forgetting factor. Obviously, $p = 1$ corresponds to the integral action, and $0 < p < 1$ corresponds to a low pass filter in the iteration domain. The trade-off is, the learned useful signal will also be discounted. Recently two works relevant to *i.i.c.* have been reported. In [26], an initial state learning algorithm has been proposed to address the first *i.i.c.*, $(C2)$, by making the initial state $x_i(0)$ a convergent sequence. The price is, one needs to assume that the system states, at least the initial states, are accessible and reachable. In [124], an initial rectifying action is proposed to address the *i.i.c.* $(C2'')$, by revising the target trajectory nearby the initial stage into a new one, which aligns its initial value with that of the actual system output.

Generally speaking, *i.i.c.* and GLC are necessary within the present ILC framework based on contraction mapping. In order to remove those conditions, we have to establish a new learning control framework.

1.1.6 Why ILC

From the above descriptions, though extremely simple, we can see the features of ILC:

1. ILC aims at output tracking control, without using any knowledge of the system state;
2. It has a very simple structure – an integration along the iteration axis;
3. It is a memory based learning process, as we need to record error signals $\Delta y_i(t)$ and control input signals $u_i(t)$ over the entire time interval $[0,\ T]$;
4. It is open-loop in the time domain, but $closed - loop$ in the iteration domain. The present control input consists of error and input signals of the previous iteration only;
5. It requires very little system knowledge, in fact only the bound of the direct tranmission term, for instance the parameter d, is needed to guarantee the learning convergence iteratively. Thus it is almost a model-free method. This is a very desirable feature in control implementation;
6. Due to the open-loop nature, there is no warranty of the stability of the dynamic system along the time axis. As far as the interval T is finite, and

the dynamics is either linear or GLC, finite escape time phenomenon does not occur, hence the system state is always bounded. We will discuss this in more details in Chapter 7;
7. The *i.i.c.* $(C2)$, $(C2')$ and $(C2'')$ play important role in the learning process;
8. The control task – the target trajectory $y_d(t)$ must be identical for all iterations.

We can see, that ILC *complements* existing control approaches in many aspects, and offers an alternative way to handle uncertain systems. One may argue, that for such a simple LTI or LTV system as (1.2), many other control methods are also applicable and able to achieve equally good performance. Let us consider a simple case, where $a(t)$ in (1.2) is an unknown and rapid time varying parameter. Since $a(t)$ is unknown to us, we have to assume the worst case that it may take any positive value (open-loop unstable). Consequently we need to close the loop with a sufficiently high gain, which leads to a very conservative control design, even though $a(t)$ may actually be negative $\forall t \in [0,\ T]$, i.e. the system is open-loop stable. On the contrary, ILC offers a *low gain control*, which is able to learn and improve the control performance iteratively. We can verify this property from the preceding ILC design, where the learning gain q is always chosen to be sufficient low.

Let us consider another example

$$\dot{x} = f(x, u, t) \quad x(0) = x_0$$
$$y = g(x, u, t)$$

where f and g are global Lipschitz continuous functions of the arguments x and u. As we will demonstrate later in Chapters 2 to 6, the only prior information required in ILC design, is

$$\frac{\partial g}{\partial u} \in [\alpha_1, \alpha_2] \quad \forall x, u, t.$$

This prior information, however, is inadequate for us to complete the design by means of many existing control methods.

It is natural for control experts to raise another question, that it would be practically ill-logical to totally ignore the dynamics $f(x, u, t)$ if we do have partial knowledge, for instance, $f(x, u, t)$ may consist of a nominal part, and a parametric or non-parametric uncertain component. Indeed, the best control method should be the one fully using all available information. From Chapters 7 to 9, we shall show a new ILC framework, which allows us to incorporate all the system information, and further integrate both nonlinear feedback control and memory based nonlinear learning control.

1.2 History of ILC

Various ILC schemes have been proposed in the past two decades. The beginning work [8], focused on the output tracking, first formulated the P-type and D-type ILC schemes with convergence analysis. Since then, ILC has been extensively studied and achieved significant progress in both theory and applications, and now becomes one of the most active fields in intelligent control and system control. In the history of ILC development, theory, design and applications have been widely exploited. We summerize briefly the development of ILC in the following aspects.

Contraction Mapping Method

Contraction mapping method forms the basis of ILC theory. All preceding ILC design examples can be regarded as the simple form of contraction mapping, though without rigorous mathematical derivations. The key requirements of such kind of method are the global Lipschitz continuity (GLC) of system dynamics, as well as the identical initialization condition. The perfect output tracking can be achieved in any finite time interval, by means of the time-weighted norm. Along the direction of [8, 10], P-type ILC [14, 113, 111, 33, 27], D-type ILC [8, 51, 53], PD-type ILC [74], high-order ILC [17, 24, 28, 27, 23], etc. have been reported.

Note that the ILC mechanism, such as (1.1) and (1.10), links the system dynamics between two consecutive iterations, hence generates a discrete dynamics along iteration axis. The learning performance analysis has been conducted in the iteration domain, by [77, 6, 118, 69, 16, 142, 148], etc., and in particular the robustness of ILC has also been investigated by [13, 14, 24, 113, 33, 72, 118, 144, 74, 27, 148], etc.

ILC has been extended to miscellaneous systems, including time-delay systems [55, 109, 100, 59, 139], sampled-data systems [138, 32], discrete-time systems [127, 49, 68, 128, 3, 6, 112, 31, 28, 131, 60] and distributed parameter systems [79, 106], etc.

Design and analysis of ILC in the frequency domain has attracted equal attention as in the time domain, see [54, 58, 43, 81, 78, 42, 59, 140], etc. The frequency analysis plays a crucial role in ILC applications, because the convergence condition can be relaxed from the infinite frequency bandwidth to a finite frequency bandwidth. ILC with H_∞ design [96, 5, 40], ILC with optimal design [4, 46, 70], etc. have also been exploited.

2-D Theory Based Method

2-D system theory [19, 108, 112, 110] considers the time domain and the iteration domain simultaneously. Taking advantage of the advanced linear multivariable theory, the convergence of the tracking error in iteration domain and stability in time-domain can be analyzed systematically. ILC designs based on Linear 2-D theory 2-D theory have been explored [49, 68, 2, 35, 44], etc. Nonlinear 2-D theory has also been exploited [67].

Energy Function Method

Lyapunov's direct method has been widely employed in the control design and analysis of nonlinear dynamic systems, and now regarded as one of the most important tools in dealing with nonlinear uncertain systems. Being inspired by Lyapunov's direct method, the concept of energy function in both the time domain and the iteration domain has been developed, which opens a new avenue for the learning control design and convergence analysis in the iteration domain.

ILC based on an energy function defined in the iteration domain have been developed [50, 97, 144]. With the help of this approach, robust learning control [63, 143, 135] and adaptive learning control [82, 66, 94, 62, 122, 134] have been developed to handle nonlinear systems with parametric or non-parametric uncertainties.

Recently the composite energy function (CEF), which reflects the system energy in both the time domain and the iteration domain, is further developed and applied to ILC [145, 147], etc. By means of energy function, the new and powerful control design methods, such as the backstepping design and nonlinear optimality, can be used as systematic design tools for the ILC construction [47, 146, 107].

Applications

Besides the theoretical development, ILC schemes have been applied to many practical control engineering problems. One of the main application areas is robotic control, for instance, the rigid robotic manipulators [11, 150, 15, 91], robotic manipulators with flexible joints [130], and flexible links [79], etc. ILC has also been applied to a variety of real-time control problems. Here we only select a subset from the application list: batch process [73, 139], wafer temperature control [34], welding process [86], disk drives [83, 30], hydraulic servo [99], piezoelectronic motor [126], PMS motor [141], linear motor [125], switched reluctance motor [114], injection molding machine [52], functional neuromuscular stimulation [41], aero-related problems [29], etc. A number of practical ILC design guidelines have been summarized, e.g. [75, 76], etc.

Supplementary Literature Survey

Among the numerous ILC publications, there are a number of publications offering wide survey on the state-of-the-art progress in the ILC field, including [88, 85, 136, 132, 133], etc. Besides, two monographs [84, 25], one edited volume [18] and two special issues [36, 37] summarized the latest ILC developments and achievements respectively, by the time of publication.

1.3 Book Overview

The contents of the book can be roughly categorized into two parts. The first part of this book, from Chapters 2 to 6, focuses on ILC design and analysis under the framework of contraction mapping. The dynamic systems under consideration are essentially linear or global Lipschitz continuous. The ILC is to perform the output tracking tasks. As far as the learning algorithms are concerned, the first part can be further divided into the linear ILC (Chapters

2-4) and nonlinear ILC (Chapters 5-6). The second part of this book, from Chapters 7 to 9, concentrates on ILC based on the composite energy function. The dynamic systems under consideration can be local Lipschitz continuous and associated with either time-varying parametric or non-parametric uncertainties. The ILC is to perform tracking control in the state space.

Distinct from most control methods, ILC permits an extra degree of freedom – learning along the iteration axis. This new property, however, also gives rise to a new challenge – how to quantify and evaluate the performance of ILC in the iteration domain? Second, with the only available system knowledge regarding the interval bound of the system gain, can we still design an ILC in some optimal sense? Chapter 2 addresses those two issues. First, three performance indices – convergence speed, the global uniform bound of the learning error in the iteration domain, and the monotonic convergence interval – have been introduced to quantify and evaluate the learning performance in the iteration domain. Then a new robust optimal design, based on the interval bound of the system gain, is developed for linear ILC schemes to achieve the fastest learning convergence speed.

Most ILC schemes use the control information acquired in the previous iteration. One question is: why not use the control information of more than one iteration? Intuitively, the more information can an ILC use, the better would be the learning performance. This is a rather controversial issue in ILC area. In Chapter 3 we show that this intuition may not be true, as far as the convergence speed under the interval uncertainty is concerned. The simplest first order linear ILC, such as (1.1), could be the best.

The robust optimal design of linear-type ILC, presented in Chapter 2, is extended to MIMO systems in Chapter 4. The main difficulty of such extension lies in the process of solving a min-max problem, where the system gain matrix under consideration could be highly nonlinear, time-varying, uncertain, and asymmetric. An interesting method is developed in Chapter 4, which provides a new mathematical tool solving the min-max problem in general, and the robust optimal ILC design in particular.

In practice, the learning speed along the iteration axis is the most important factor. A question immediately following the Chapter 3 is, can we come up with an ILC scheme with a faster convergence speed than the simplest first order linear ILC? Chapter 5 provides the answer – a *conditional* yes, if we are allowed to use the system state information, and the output relation, i.e. $y = g(x, u, t)$, is completely known to us. Two types of *nonlinear* ILC schemes – the Newton-type and Secant-type, are developed. It is shown by rigorous proof, that the Newton-type ILC achieves the fastest convergence speed, and the Secant-type ILC wins the second place.

In Chapter 6, the nonlinear ILC schemes, the Newton-type and Secant-type, are extended to MIMO systems.

Upto Chapter 6, all ILC schemes focus on GLC systems with output tracking, in an almost model free manner, at most only use the output relation. Can the iterative learning be extended to more general classes of nonlinear systems? Can an iterative learning controller incorporate more of system state dynamics knowledge if it is available? Chapter 7 attempts to establish such a framework based on the composite energy function (CEF). The CEF consists of two components: one is a positive scalar function capturing the tracking error information in the time domain, and the other is a functional over the entire learning period reflecting the (parametric) learning error in the iteration domain. When the system unknowns are limited to time-varying parameters, it is easy to construct an *nonlinear* learning mechanism, consisting of a nonlinear stabilizing feedback part and a pointwise learning part. The bottleneck in contraction mapping based ILC, the GLC assumption, can thus be overcome.

One important contribution of CEF is, it bridges the gap between ILC and other advanced nonlinear control methods, such as nonlinear optimal control, nonlinear internal model control, adaptive control, nonlinear H_∞ control, nonlinear robust control, etc. Chapter 8 demonstrates one such integration – quasi-optimal ILC, based on a modified Sontag's formula for the known nominal part of the process. Meanwhile, learning is carried out iteratively to handle time-varying uncertainties in the nonlinear dynamic system. Along with the pointwise convergence in learning, the suboptimal properties, such as the balance between the control effort and the tracking error in the iteration domain, are preserved.

Finally we explore another important direction: combination of ILC and other predominant intelligent control methods characterized by black-box based approximations. By virtue of the CEF based design, Chapter 9 exploits a learning wavelet control method to achieve the pre-defined tracking performance in finite iterations and ensure the boundedness of all signals. Different from other intelligent control methods such as neural control, the novelty of the learning wavelet control lies in that both the network parameters and structure can be tuned easily, owing to the orthonormality property of the wavelet network. Further, by virtue of the multi-resolution property of the wavelet network, the finiteness of the network structure is assured. The learning wavelet control mechanism can successfully handle the lumped (nonparametric) system uncertainties, for instance the $\mathcal{L}^2(\mathcal{R})$ class in the state space.

Conclusions are drawn and some recommendations for future research are given in Chapter 10. [1, 2]